

# MFRC530

## Standard MIFARE reader solution

Rev. 3.4 — 12 February 2014  
057434

Product data sheet  
COMPANY PUBLIC

## 1. Introduction

---

This data sheet describes the functionality of the MFRC530 Integrated Circuit (IC). It includes the functional and electrical specifications and from a system and hardware viewpoint gives detailed information on how to design-in the device.

**Remark:** The MFRC530 supports all variants of the MIFARE Classic, MIFARE 1K and MIFARE 4K RF identification protocols. To aid readability throughout this data sheet, the MIFARE Classic, MIFARE 1K and MIFARE 4K products and protocols have the generic name MIFARE.

## 2. General description

---

The MFRC530 is highly integrated reader IC for contactless communication at 13.56 MHz. The MFRC530 reader IC provides:

- outstanding modulation and demodulation for passive contactless communication
- a wide range of methods and protocols
- pin compatibility with the CLRC632, MFRC500, MFRC531 and SLRC400

All protocol layers of the ISO/IEC 14443 A are supported

The receiver module provides a robust and efficient demodulation/decoding circuitry implementation for compatible transponder signals (see [Section 9.10 on page 31](#)). The digital module, manages the complete ISO/IEC 14443 A standard framing and error detection (parity and CRC). In addition, it supports the fast Crypto1 security algorithm for authenticating the MIFARE products (see [Section 9.13 on page 37](#)).

The internal transmitter module ([Section 9.9 on page 28](#)) can directly drive an antenna designed for a proximity operating distance up to 100 mm without any additional active circuitry.

A parallel interface can be directly connected to any 8-bit microprocessor to ensure reader/terminal design flexibility. In addition, Serial Peripheral Interface (SPI) compatibility is supported (see [Section 9.1.4 on page 9](#)).



## 3. Features and benefits

---

### 3.1 General

- Highly integrated analog circuitry for demodulating and decoding card response
- Buffered output drivers enable antenna connection using the minimum of external components
- Proximity operating distance up to 100 mm
- Supports the ISO/IEC 14443 A standard, parts 1 to 4
- Supports MIFARE dual-interface card ICs and MIFARE Classic protocol
- Crypto1 and secure non-volatile internal key memory
- Pin-compatible with the CLRC632, MFRC500, MFRC531 and the SLRC400
- Parallel microprocessor interface with internal address latch and IRQ line
- Flexible interrupt handling
- Automatic detection of parallel microprocessor interface type
- 64-byte send and receive FIFO buffer
- Hard reset with low power function
- Software triggered Power-down mode
- Programmable timer
- Unique serial number
- User programmable start-up configuration
- Bit-oriented and byte oriented framing
- Independent power supply pins for analog, digital and transmitter modules
- Internal oscillator buffer optimized for low phase jitter enables 13.56 MHz quartz connection
- Clock frequency filtering
- 3.3 V to 5 V operation for transmitter in short range and proximity applications
- 3.3 V or 5 V operation for the digital module

## 4. Applications

---

- Electronic payment systems
- Identification systems
- Access control systems
- Subscriber services
- Banking systems
- Digital content systems

## 5. Quick reference data

Table 1. Quick reference data

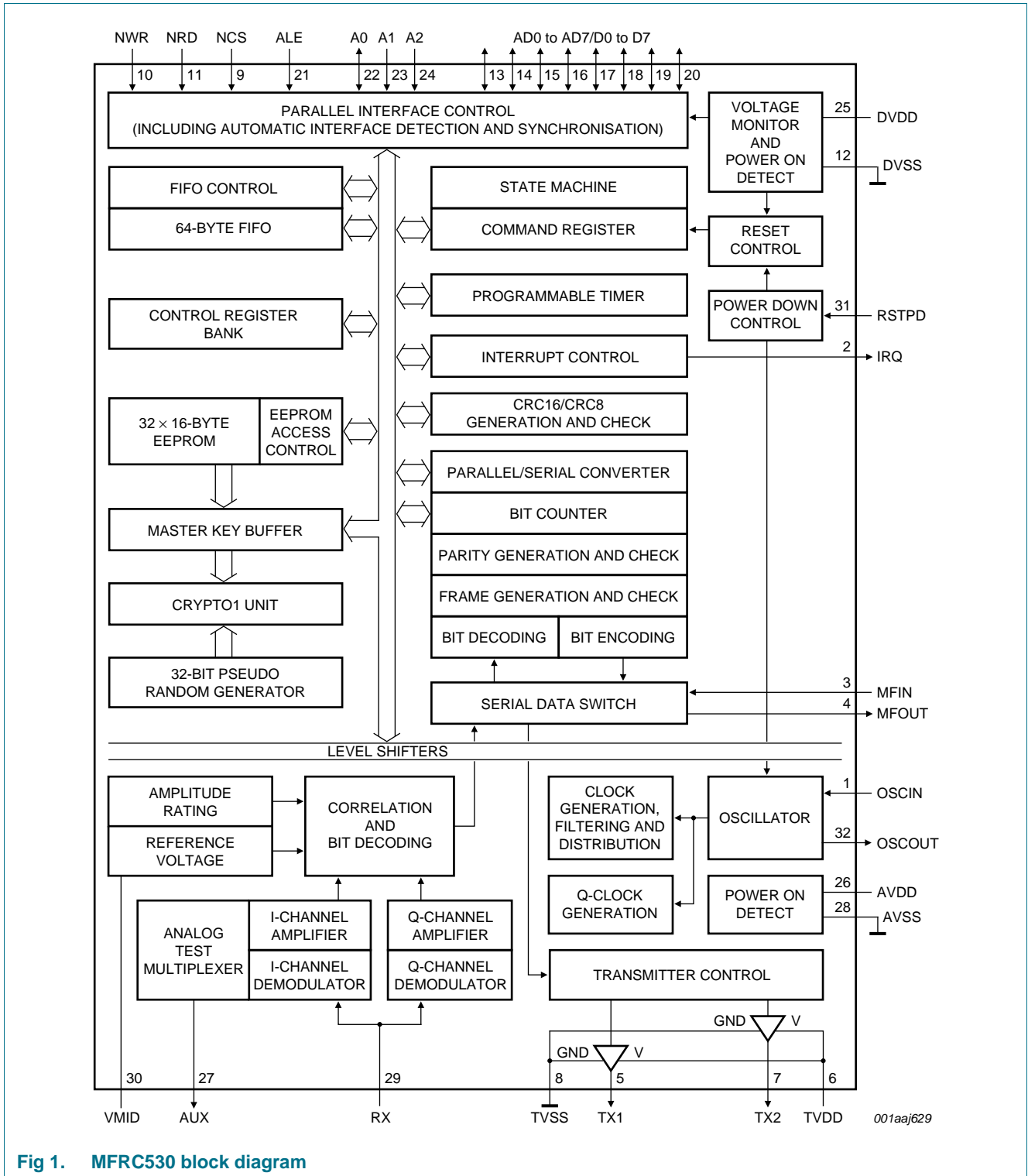
| Symbol         | Parameter                      | Conditions                 | Min  | Typ | Max             | Unit |
|----------------|--------------------------------|----------------------------|------|-----|-----------------|------|
| $T_{amb}$      | ambient temperature            |                            | -40  | -   | +150            | °C   |
| $T_{stg}$      | storage temperature            |                            | -40  | -   | +150            | °C   |
| $V_{DDD}$      | digital supply voltage         |                            | -0.5 | +5  | +6              | V    |
| $V_{DDA}$      | analog supply voltage          |                            | -0.5 | +5  | +6              | V    |
| $V_{DD(TVDD)}$ | TVDD supply voltage            |                            | -0.5 | +5  | +6              | V    |
| $ V_i $        | input voltage (absolute value) | on any digital pin to DVSS | -0.5 | -   | $V_{DDD} + 0.5$ | V    |
|                |                                | on pin RX to AVSS          | -0.5 | -   | $V_{DDA} + 0.5$ | V    |
| $I_{LI}$       | input leakage current          |                            | -1.0 | -   | +1.0            | mA   |
| $I_{DD(TVDD)}$ | TVDD supply current            | continuous wave            | -    | -   | 150             | mA   |

## 6. Ordering information

Table 2. Ordering information

| Type number    | Package |  | Version  |
|----------------|---------|--|----------|
|                | Name    | Description  |          |
| MFRC53001T/0FE | SO32    | plastic small outline package; 32 leads; body width 7.5 mm | SOT287-1 |

**7. Block diagram**



**Fig 1. MFRC530 block diagram**

## 8. Pinning information

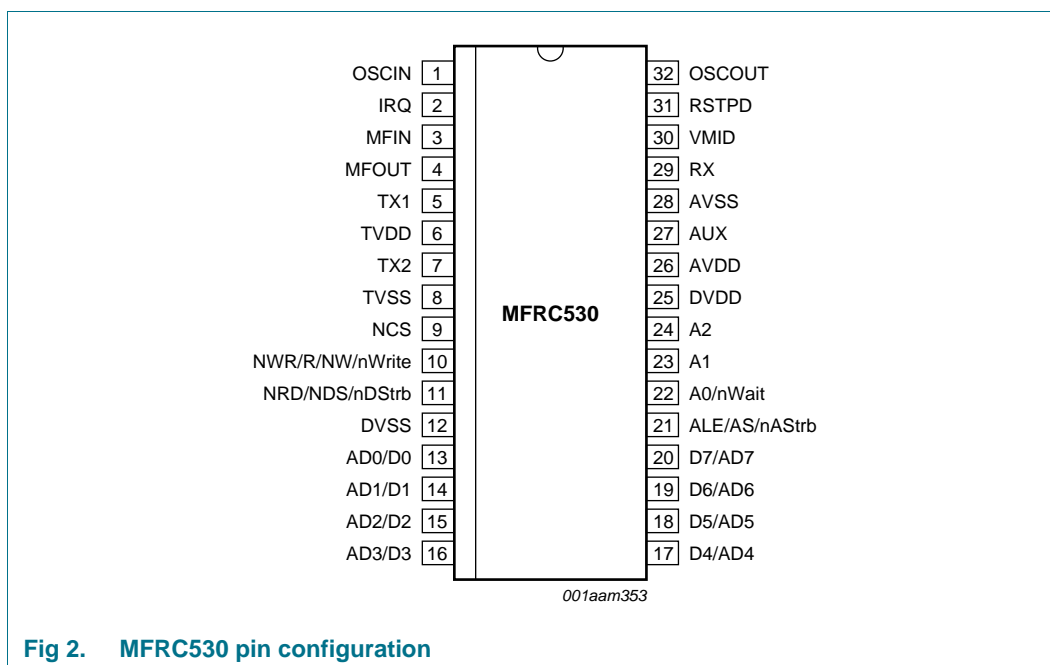


Fig 2. MFRC530 pin configuration

### 8.1 Pin description

Table 3. Pin description

| Pin               | Symbol | Type <sup>[1]</sup> | Description  |
|-------------------|--------|---------------------|--|
| 1                 | OSCIN  | I                   | oscillator/clock inputs:<br>crystal oscillator input to the oscillator's inverting amplifier<br>externally generated clock input; $f_{clk(ext)} = 13.56 \text{ MHz}$ |
| 2                 | IRQ    | O                   | interrupt request: generates an output signaling an interrupt event  |
| 3                 | MFIN   | I                   | ISO/IEC 14443 A MIFARE serial data interface input   |
| 4 <sup>[2]</sup>  | MFOUT  | O                   | serial data ISO/IEC 14443 A output   |
| 5                 | TX1    | O                   | transmitter 1 modulated carrier output; 13.56 MHz  |
| 6                 | TVDD   | P                   | transmitter power supply for the TX1 and TX2 output stages   |
| 7                 | TX2    | O                   | transmitter 2 modulated carrier output; 13.56 MHz  |
| 8                 | TVSS   | G                   | transmitter ground for the TX1 and TX2 output stages   |
| 9                 | NCS    | I                   | not chip select input is used to select and activate the MFRC530's microprocessor interface  |
| 10 <sup>[3]</sup> | NWR    | I                   | not write input generates the strobe signal for writing data to the MFRC530 registers when applied to pins D0 to D7  |
|                   | R/NW   | I                   | read not write input is used to switch between read or write cycles  |
|                   | nWrite | I                   | not write input selects the read or write cycle to be performed  |
| 11 <sup>[3]</sup> | NRD    | I                   | not read input generates the strobe signal for reading data from the MFRC530 registers when applied to pins D0 to D7   |
|                   | NDS    | I                   | not data strobe input generates the strobe signal for the read and write cycles  |
|                   | nDStrb | I                   | not data strobe input generates the strobe signal for the read and write cycles  |

Table 3. Pin description ...continued

| Pin                     | Symbol     | Type <sup>[1]</sup> | Description  |
|-------------------------|------------|---------------------|--|
| 12                      | DVSS       | G                   | digital ground   |
| 13                      | D0         | O                   | SPI master in, slave out output  |
| 13 to 20 <sup>[3]</sup> | D0 to D7   | I/O                 | 8-bit bidirectional data bus input/output on pins D0 to D7   |
|                         | AD0 to AD7 | I/O                 | 8-bit bidirectional address and data bus input/output on pins AD0 to AD7   |
| 21 <sup>[3]</sup>       | ALE        | I                   | address latch enable input for pins AD0 to AD5; HIGH latches the internal address  |
|                         | AS         | I                   | address strobe input for pins AD0 to AD5; HIGH latches the internal address  |
|                         | nAStrb     | I                   | not address strobe input for pins AD0 to AD5; LOW latches the internal address   |
|                         | NSS        | I                   | not slave select strobe input for SPI communication  |
| 22 <sup>[3]</sup>       | A0         | I                   | address line 0 is the address register bit 0 input   |
|                         | nWait      | O                   | not wait output:<br>LOW starts an access cycle<br>HIGH ends an access cycle  |
|                         | MOSI       | I                   | SPI master out, slave in   |
| 23                      | A1         | I                   | address line 1 is the address register bit 1 input   |
| 24 <sup>[3]</sup>       | A2         | I                   | address line 2 is the address register bit 2 input   |
|                         | SCK        | I                   | SPI serial clock input   |
| 25                      | DVDD       | P                   | digital power supply   |
| 26                      | AVDD       | P                   | analog power supply for pins OSCIN, OSCOUT, RX, VMID and AUX   |
| 27                      | AUX        | O                   | auxiliary output is used to generate analog test signals. The output signal is selected using the TestAnaSelect register's TestAnaOutSel[4:0] bits   |
| 28                      | AVSS       | G                   | analog ground  |
| 29                      | RX         | I                   | receiver input is used as the card response input. The carrier is load modulated at 13.56 MHz, drawn from the antenna circuit  |
| 30                      | VMID       | P                   | internal reference voltage pin provides the internal reference voltage as a supply<br><b>Remark:</b> It must be connected to a 100 nF block capacitor connected between pin VMID and ground          |
| 31                      | RSTPD      | I                   | reset and power-down input:<br>HIGH: the internal current sinks are switched off, the oscillator is inhibited and the input pads are disconnected<br>LOW (negative edge): start internal reset phase |
| 32                      | OSCOUT     | O                   | crystal oscillator output for the oscillator's inverting amplifier   |

[1] Pin types: I = Input, O = Output, I/O = Input/Output, P = Power and G = Ground.

[2] The SLRC400 uses pin name SIGOUT for pin MFOUT. The MFRC530 functionality includes test functions for the SLRC400 using pin MFOUT.

[3] These pins provide different functionality depending on the selected microprocessor interface type (see [Section 9.1 on page 7](#) for detailed information).

## 9. Functional description

### 9.1 Digital interface

#### 9.1.1 Overview of supported microprocessor interfaces

The MFRC530 supports direct interfacing to various 8-bit microprocessors. Alternatively, the MFRC530 can be connected to a PC's Enhanced Parallel Port (EPP). [Table 4](#) shows the parallel interface signals supported by the MFRC530.

**Table 4. Supported microprocessor and EPP interface signals**

| Bus control signals                               | Bus     | Separated address and data bus | Multiplexed address and data bus |
|---|---------|--------------------------------|----------------------------------|
| Separated read and write strobes                  | control | NRD, NWR, NCS                  | NRD, NWR, NCS, ALE               |
|   | address | A0, A1, A2                     | AD0, AD1, AD2, AD3, AD4, AD5     |
|   | data    | D0 to D7                       | AD0 to AD7                       |
| Common read and write strobe                      | control | R/NW, NDS, NCS                 | R/NW, NDS, NCS, AS               |
|   | address | A0, A1, A2                     | AD0, AD1, AD2, AD3, AD4, AD5     |
|   | data    | D0 to D7                       | AD0 to AD7                       |
| Common read and write strobe with handshake (EPP) | control | -                              | nWrite, nDStrb, nAStrb, nWait    |
|   | address | -                              | AD0, AD1, AD2, AD3, AD4, AD5     |
|   | data    | -                              | AD0 to AD7                       |

#### 9.1.2 Automatic microprocessor interface detection

After a Power-On or Hard reset, the MFRC530 resets the parallel microprocessor interface mode and detects the microprocessor interface type.

The MFRC530 identifies the microprocessor interface using the logic levels on the control pins. This is performed using a combination of fixed pin connections and the dedicated Initialization routine (see [Section 9.7.4 on page 27](#)).

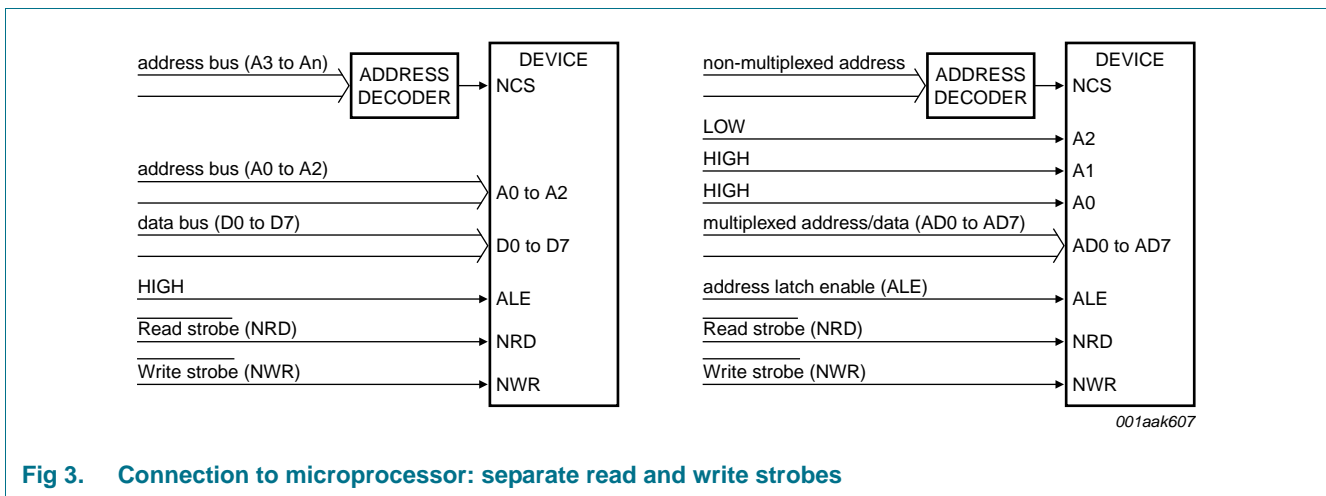
**9.1.3 Connection to different microprocessor types**

The connection to various microprocessor types is shown in [Table 5](#).

**Table 5. Connection scheme for detecting the parallel interface type**

| MFRC530 pins | Parallel interface type and signals |                         |                          |                         |  |
|--------------|-------------------------------------|-------------------------|--------------------------|-------------------------|--|
|              | Separated read/write strobe         |                         | Common read/write strobe |                         |  |
|              | Dedicated address bus               | Multiplexed address bus | Dedicated address bus    | Multiplexed address bus | Multiplexed address bus with handshake |
| ALE          | HIGH                                | ALE                     | HIGH                     | AS                      | nAStb                                  |
| A2           | A2                                  | LOW                     | A2                       | LOW                     | HIGH                                   |
| A1           | A1                                  | HIGH                    | A1                       | HIGH                    | HIGH                                   |
| A0           | A0                                  | HIGH                    | A0                       | LOW                     | nWait                                  |
| NRD          | NRD                                 | NRD                     | NDS                      | NDS                     | nDStb                                  |
| NWR          | NWR                                 | NWR                     | R/NW                     | R/NW                    | nWrite                                 |
| NCS          | NCS                                 | NCS                     | NCS                      | NCS                     | LOW                                    |
| D7 to D0     | D7 to D0                            | AD7 to AD0              | D7 to D0                 | AD7 to AD0              | AD7 to AD0                             |

**9.1.3.1 Separate read and write strobe**



Refer to [Section 13.4.1 on page 91](#) for timing specification.



9.1.3.2 Common read and write strobe

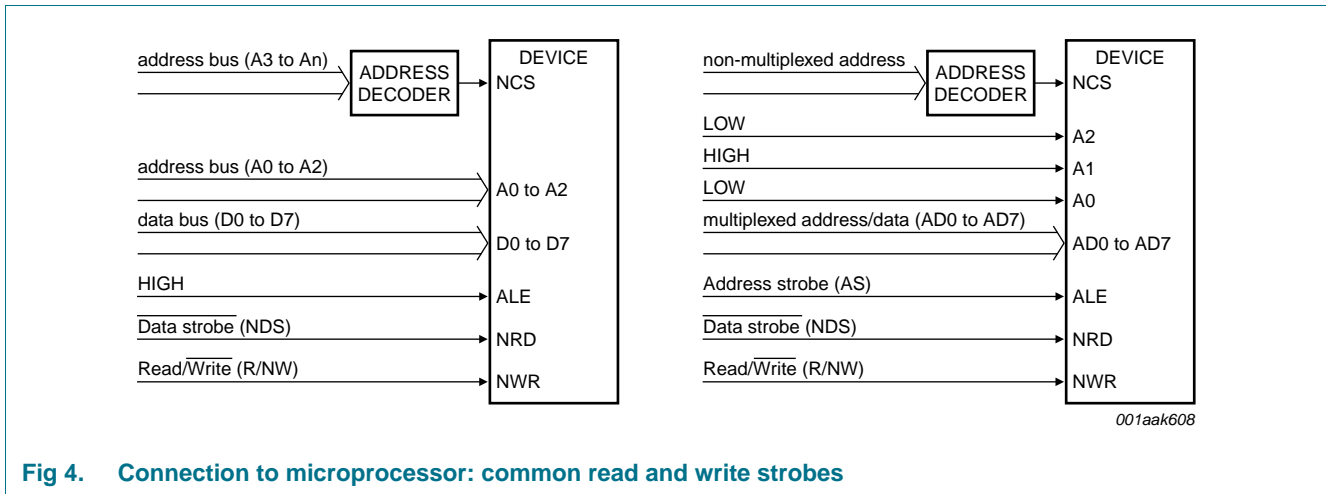


Fig 4. Connection to microprocessor: common read and write strobes

Refer to [Section 13.4.2 on page 92](#) for timing specification.

9.1.3.3 Common read and write strobe: EPP with handshake

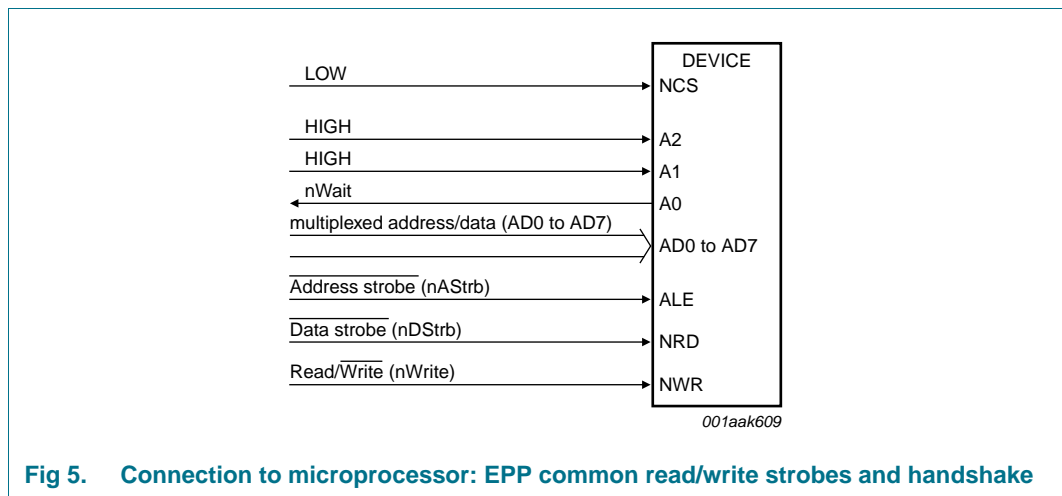


Fig 5. Connection to microprocessor: EPP common read/write strobes and handshake

Refer to [Section 13.4.3 on page 93](#) for timing specification.

**Remark:** In the EPP standard a chip select signal is not defined. To cover this situation, the status of the NCS pin can be used to inhibit the nDStrb signal. If this inhibitor is not used, it is mandatory that pin NCS is connected to pin DVSS.

**Remark:** After each Power-On or Hard reset, the nWait signal on pin A0 is high-impedance. nWait is defined as the first negative edge applied to the nAStrb pin after the reset phase. The MFRC530 does not support Read Address Cycle.

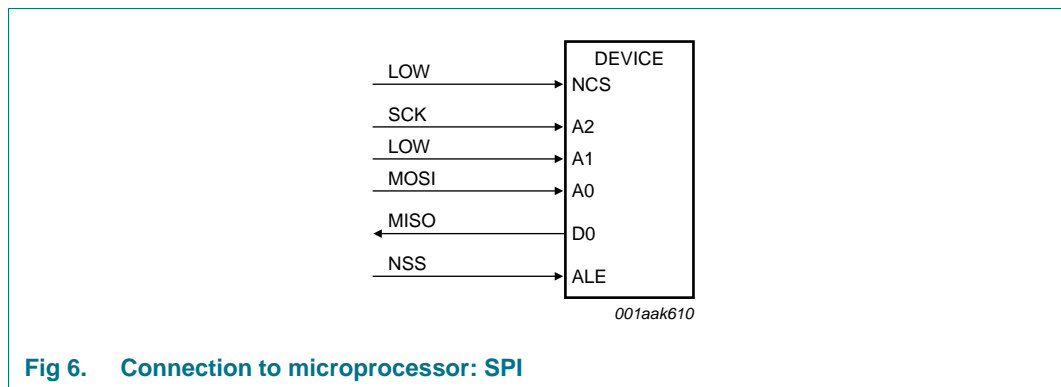
9.1.4 Serial Peripheral Interface

The MFRC530 provides compatibility with the 5-wire Serial Peripheral Interface (SPI) standard and acts as a slave during the SPI communication. The SPI clock signal SCK must be generated by the master. Data communication from the master to the slave uses the MOSI line. The MISO line sends data from the MFRC530 to the master.

**Table 6. SPI compatibility**

| MFRC530 pins | SPI pins       |
|--------------|----------------|
| ALE          | NSS            |
| A2           | SCK            |
| A1           | LOW            |
| A0           | MOSI           |
| NRD          | HIGH           |
| NWR          | HIGH           |
| NCS          | LOW            |
| D7 to D1     | do not connect |
| D0           | MISO           |

Figure 6 shows the microprocessor connection to the MFRC530 using SPI.



**Fig 6. Connection to microprocessor: SPI**

**Remark:** The SPI implementation for MFRC530 conforms to the SPI standard and ensures that the MFRC530 can only be addressed as a slave.

**9.1.4.1 SPI read data**

The structure shown in Table 7 must be used to read data using SPI. It is possible to read up to n-data bytes. The first byte sent defines both, the mode and the address.

**Table 7. SPI read data**

| Pin  | Byte 0    | Byte 1    | Byte 2    | ... | Byte n     | Byte n + 1 |
|------|-----------|-----------|-----------|-----|------------|------------|
| MOSI | address 0 | address 1 | address 2 | ... | address n  | 00         |
| MISO | XX        | data 0    | data 1    | ... | data n – 1 | data n     |

The address byte must meet the following criteria:

- the Most Significant Bit (MSB) of the first byte sets the mode. To read data from the MFRC530 the MSB is set to logic 1
- bits [6:1] define the address
- the Least Significant Bit (LSB) should be set to logic 0.

As shown in Table 8, all the bits of the last byte sent are set to logic 0.

**Table 8. SPI read address**

| Address (MOSI)   | Bit 7 (MSB) | Bit 6   | Bit 5   | Bit 4   | Bit 3   | Bit 2   | Bit 1   | Bit 0 (LSB) <sup>[1]</sup> |
|------------------|-------------|---------|---------|---------|---------|---------|---------|----------------------------|
| byte 0           | 1           | address | address | address | address | address | address | reserved                   |
| byte 1 to byte n | reserved    | address | address | address | address | address | address | reserved                   |
| byte n + 1       | 0           | 0       | 0       | 0       | 0       | 0       | 0       | 0                          |

[1] All reserved bits must be set to logic 0.

#### 9.1.4.2 SPI write data

The structure shown in [Table 9](#) must be used to write data using SPI. It is possible to write up to n-data bytes. The first byte sent defines both the mode and the address.

**Table 9. SPI write data**

|      | Byte 0  | Byte 1 | Byte 2 | ... | Byte n     | Byte n + 1 |
|------|---------|--------|--------|-----|------------|------------|
| MOSI | address | data 0 | data 1 | ... | data n – 1 | data n     |
| MISO | XX      | XX     | XX     | ... | XX         | XX         |

The address byte must meet the following criteria:

- the MSB of the first byte sets the mode. To write data to the MFRC530, the MSB is set to logic 0
- bits [6:1] define the address
- the LSB should be set to logic 0.

SPI write mode writes all data to the address defined in byte 0 enabling effective write cycles to the FIFO buffer.

**Table 10. SPI write address**

| Address line (MOSI)  | Bit 7 (MSB) | Bit 6   | Bit 5   | Bit 4   | Bit 3   | Bit 2   | Bit 1   | Bit 0 (LSB) <sup>[1]</sup> |
|----------------------|-------------|---------|---------|---------|---------|---------|---------|----------------------------|
| byte 0               | 0           | address | address | address | address | address | address | reserved                   |
| byte 1 to byte n + 1 | data        | data    | data    | data    | data    | data    | data    | data                       |

[1] All reserved bits must be set to logic 0.

**Remark:** The data bus pins D7 to D0 must be disconnected.

Refer to [Section 13.4.4 on page 95](#) for the timing specification.

## 9.2 Memory organization of the EEPROM

Table 11. EEPROM memory organization diagram

| Block    |         | Byte address | Access | Memory content                       | Refer to                                   |
|----------|---------|--------------|--------|--------------------------------------|--|
| Position | Address |              |        |                                      |  |
| 0        | 0       | 00h to 0Fh   | R      | product information field            | <a href="#">Section 9.2.1 on page 13</a>   |
| 1        | 1       | 10h to 1Fh   | R/W    | StartUp register initialization file | <a href="#">Section 9.2.2.1 on page 13</a> |
| 2        | 2       | 20h to 2Fh   | R/W    |                                      |  |
| 3        | 3       | 30h to 3Fh   | R/W    |                                      |  |
| 4        | 4       | 40h to 4Fh   | R/W    |                                      |  |
| 5        | 5       | 50h to 5Fh   | R/W    |                                      |  |
| 6        | 6       | 60h to 6Fh   | R/W    |                                      |  |
| 7        | 7       | 70h to 7Fh   | R/W    |                                      |  |
| 8        | 8       | 80h to 8Fh   | W      | keys for Crypto1                     | <a href="#">Section 9.2.3 on page 15</a>   |
| 9        | 9       | 90h to 9Fh   | W      |                                      |  |
| 10       | A       | A0h to AFh   | W      |                                      |  |
| 11       | B       | B0h to BFh   | W      |                                      |  |
| 12       | C       | C0h to CFh   | W      |                                      |  |
| 13       | D       | D0h to DFh   | W      |                                      |  |
| 14       | E       | E0h to EFh   | W      |                                      |  |
| 15       | F       | F0h to FFh   | W      |                                      |  |
| 16       | 10      | 100h to 10Fh | W      |                                      |  |
| 17       | 11      | 110h to 11Fh | W      |                                      |  |
| 18       | 12      | 120h to 12Fh | W      |                                      |  |
| 19       | 13      | 130h to 13Fh | W      |                                      |  |
| 20       | 14      | 140h to 14Fh | W      |                                      |  |
| 21       | 15      | 150h to 15Fh | W      |                                      |  |
| 22       | 16      | 160h to 16Fh | W      |                                      |  |
| 23       | 17      | 170h to 17Fh | W      |                                      |  |
| 24       | 18      | 180h to 18Fh | W      |                                      |  |
| 25       | 19      | 190h to 19Fh | W      |                                      |  |
| 26       | 1A      | 1A0h to 1AFh | W      |                                      |  |
| 27       | 1B      | 1B0h to 1BFh | W      |                                      |  |
| 28       | 1C      | 1C0h to 1CFh | W      |                                      |  |
| 29       | 1D      | 1D0h to 1DFh | W      |                                      |  |
| 30       | 1E      | 1E0h to 1EFh | W      |                                      |  |
| 31       | 1F      | 1F0h to 1FFh | W      |                                      |  |

## 9.2.1 Product information field (read only)

**Table 12. Product information field byte allocation**

| Byte   | 15  | 14       | 13 | 12 | 11                    | 10 | 9 | 8 | 7 | 6 | 5                           | 4 | 3 | 2 | 1 | 0 |
|--------|-----|----------|----|----|-----------------------|----|---|---|---|---|-----------------------------|---|---|---|---|---|
| Symbol | CRC | Internal |    |    | Product Serial Number |    |   |   | - |   | Product Type Identification |   |   |   |   |   |
| Access | R   | R        |    |    | R                     |    |   |   | R |   | R                           |   |   |   |   |   |

**Table 13. Product information field byte description**

| Byte     | Symbol                      | Access | Value | Description  |
|----------|-----------------------------|--------|-------|--|
| 15       | CRC                         | R      | -     | the content of the product information field is secured using a CRC byte which is checked during start-up  |
| 14 to 12 | Internal                    | R      | -     | three bytes for internal trimming parameters   |
| 11 to 8  | Product Serial Number       | R      | -     | a unique four byte serial number for the device  |
| 7 to 5   | reserved                    | R      | -     |  |
| 4 to 0   | Product Type Identification | R      | -     | the MFRC530 is a member of a new family of highly integrated reader ICs. Each member of the product family has a unique product type identification. The value of the product type identification is shown in <a href="#">Table 14</a> . |

**Table 14. Product type identification definition**

| Definition | Product type identification bytes |     |     |     |                  |
|------------|-----------------------------------|-----|-----|-----|------------------|
| Byte       | 0                                 | 1   | 2   | 3   | 4 <sup>[1]</sup> |
| Value      | 30h                               | 88h | FEh | 03h | XXh              |

[1] Byte 4 contains the current version number.

## 9.2.2 Register initialization files (read/write)

Register initialization from address 10h to address 2Fh is performed automatically during the initializing phase (see [Section 9.7.3 on page 27](#)) using the StartUp register initialization file.

In addition, the MFRC530 registers can be initialized using values from the register initialization file when the LoadConfig command is executed (see [Section 11.4.1 on page 84](#)).

**Remark:** The following points apply to initialization:

- the Page register (addressed using 10h, 18h, 20h, 28h) is skipped and not initialized.
- PreSetxx registers: do not change.
- all reserved register bits set to logic 0: do not change.

### 9.2.2.1 StartUp register initialization file (read/write)

The EEPROM memory block address 1 and 2 contents are used to automatically set the register subaddresses 10h to 2Fh during the initialization phase. The default values stored in the EEPROM during production are shown in [Section 9.2.2.2 “Factory default StartUp register initialization file”](#).

The byte assignment is shown in [Table 15](#).

**Table 15. Byte assignment for register initialization at start-up**

| EEPROM byte address    | Register address | Remark  |
|------------------------|------------------|---------|
| 10h (block 1, byte 0)  | 10h              | skipped |
| 11h                    | 11h              | copied  |
| ...                    | ...              | ...     |
| 2Fh (block 2, byte 15) | 2Fh              | copied  |

### 9.2.2.2 Factory default StartUp register initialization file

During the production tests, the StartUp register initialization file is initialized using the default values shown in [Table 16](#). During each power-up and initialization phase, these values are written to the MFRC530's registers.

**Table 16. Shipment content of StartUp configuration file**

| EEPROM byte address | Register address | Value | Symbol            | Description  |
|---------------------|------------------|-------|-------------------|--|
| 10h                 | 10h              | 00h   | Page              | free for user  |
| 11h                 | 11h              | 58h   | TxControl         | transmitter pins TX1 and TX2 are switched off, bridge driver configuration, modulator driven from internal digital circuitry |
| 12h                 | 12h              | 3Fh   | CwConductance     | source resistance of TX1 and TX2 is set to minimum   |
| 13h                 | 13h              | 3Fh   | PreSet13          | -  |
| 14h                 | 14h              | 19h   | CoderControl      | ISO/IEC 14443 A coding is set  |
| 15h                 | 15h              | 13h   | ModWidth          | pulse width for Miller pulse encoding is set to standard configuration   |
| 16h                 | 16h              | 00h   | PreSet16          | -  |
| 17h                 | 17h              | 00h   | PreSet17          | -  |
| 18h                 | 18h              | 00h   | Page              | free for user  |
| 19h                 | 19h              | 73h   | RxControl1        | ISO/IEC 14443 A is set and internal amplifier gain is maximum  |
| 1Ah                 | 1Ah              | 08h   | DecoderControl    | bit-collisions always evaluate to HIGH in the data bit stream  |
| 1Bh                 | 1Bh              | ADh   | BitPhase          | BitPhase[7:0] is set to standard configuration   |
| 1Ch                 | 1Ch              | FFh   | RxThreshold       | MinLevel[3:0] and CollLevel[3:0] are set to maximum  |
| 1Dh                 | 1Dh              | 1Eh   | BPSKDemControl    | ISO/IEC 14443 A is set   |
| 1Eh                 | 1Eh              | 41h   | RxControl2        | use Q-clock for the receiver, automatic receiver off is switched on, decoder is driven from internal analog circuitry        |
| 1Fh                 | 1Fh              | 00h   | ClockQControl     | automatic Q-clock calibration is switched on   |
| 20h                 | 20h              | 00h   | Page              | free for user  |
| 21h                 | 21h              | 06h   | RxWait            | frame guard time is set to six bit-clocks  |
| 22h                 | 22h              | 03h   | ChannelRedundancy | channel redundancy is set using ISO/IEC 14443 A  |
| 23h                 | 23h              | 63h   | CRCPreSetLSB      | CRC preset value is set using ISO/IEC 14443 A  |
| 24h                 | 24h              | 63h   | CRCPreSetMSB      | CRC preset value is set using ISO/IEC 14443 A  |
| 25h                 | 25h              | 00h   | PreSet25          | -  |
| 26h                 | 26h              | 00h   | MFOUSelect        | pin MFOUT is set LOW   |
| 27h                 | 27h              | 00h   | PreSet27          | -  |
| 28h                 | 28h              | 00h   | Page              | free for user  |

**Table 16. Shipment content of StartUp configuration file ...continued**

| EEPROM byte address | Register address | Value | Symbol       | Description   |
|---------------------|------------------|-------|--------------|---|
| 29h                 | 29h              | 08h   | FIFOLevel    | WaterLevel[5:0] FIFO buffer warning level is set to standard configuration                    |
| 2Ah                 | 2Ah              | 07h   | TimerClock   | TPreScaler[4:0] is set to standard configuration, timer unit restart function is switched off |
| 2Bh                 | 2Bh              | 06h   | TimerControl | Timer is started at the end of transmission, stopped at the beginning of reception            |
| 2Ch                 | 2Ch              | 0Ah   | TimerReload  | TReloadValue[7:0]: the timer unit preset value is set to standard configuration               |
| 2Dh                 | 2Dh              | 02h   | IRQPinConfig | pin IRQ is set to high-impedance  |
| 2Eh                 | 2Eh              | 00h   | PreSet2E     | -   |
| 2Fh                 | 2Fh              | 00h   | PreSet2F     | -   |

### 9.2.2.3 Register initialization file (read/write)

The EEPROM memory content from block address 3 to 7 can initialize register subaddresses 10h to 2Fh when the LoadConfig command is executed (see [Section 11.4.1 on page 84](#)). This command requires the EEPROM starting byte address as a two byte argument for the initialization procedure.

The byte assignment is shown in [Table 17](#).

**Table 17. Byte assignment for register initialization at startup**

| EEPROM byte address               | Register address | Remark  |
|-----------------------------------|------------------|---------|
| EEPROM starting byte address      | 10h              | skipped |
| EEPROM + 1 starting byte address  | 11h              | copied  |
|                                   | ...              | ...     |
| EEPROM + 31 starting byte address | 2Fh              | copied  |

The register initialization file is large enough to hold values for two initialization sets and up to one block (16-byte) of user data.

**Remark:** The register initialization file can be read/written by users and these bytes can be used to store other user data.

After each power-up, the default configuration enables the MIFARE and ISO/IEC 14443 A protocol.

## 9.2.3 Crypto1 keys (write only)

MIFARE security requires specific cryptographic keys to encrypt data stream communication on the contactless interface. These keys are called Crypto1 keys.

### 9.2.3.1 Key format

Keys stored in the EEPROM are written in a specific format. Each key byte must be split into lower four bits k0 to k3 (lower nibble) and the higher four bits k4 to k7 (higher nibble). Each nibble is stored twice in one byte and one of the two nibbles is bit-wise inverted. This format is a precondition for successful execution of the LoadKeyE2 (see [Section 11.6.1 on page 86](#)) and LoadKey commands (see [Section 11.6.2 on page 86](#)).

Using this format, 12 bytes of EEPROM memory are needed to store a 6-byte key. This is shown in [Figure 7](#).

|                     |                         |                         |                         |                         |                         |                         |  |  |         |  |  |  |
|---------------------|-------------------------|-------------------------|-------------------------|-------------------------|-------------------------|-------------------------|--|--|---------|--|--|--|
| Master key byte     | 0 (LSB)                 |                         |                         |                         | 1                       |                         |  |  | 5 (MSB) |  |  |  |
| Master key bits     | k7 k6 k5 k4 k7 k6 k5 k4 | k3 k2 k1 k0 k3 k2 k1 k0 | k7 k6 k5 k4 k7 k6 k5 k4 | k3 k2 k1 k0 k3 k2 k1 k0 | k7 k6 k5 k4 k7 k6 k5 k4 | k3 k2 k1 k0 k3 k2 k1 k0 |  |  |         |  |  |  |
| EEPROM byte address | n                       | n + 1                   | n + 2                   | n + 3                   | n + 10                  | n + 11                  |  |  |         |  |  |  |
| Example             | 5Ah                     | F0h                     | 5Ah                     | E1h                     | 5Ah                     | A5h                     |  |  |         |  |  |  |

001aak640

**Fig 7. Key storage format**

**Example:** The value for the key must be written to the EEPROM.

- If the key was: A0h A1h A2h A3h A4h A5h then
- 5Ah F0h 5Ah E1h 5Ah D2h 5Ah C3h 5Ah B4h 5Ah A5h would be written.

**Remark:** It is possible to load data for other key formats into the EEPROM key storage location. However, it is not possible to validate card authentication with data which will cause the LoadKeyE2 command (see [Section 11.6.1 on page 86](#)) to fail.

**9.2.3.2 Storage of keys in the EEPROM**

The MFRC530 reserves 384 bytes of memory in the EEPROM for the Crypto1 keys. No memory segmentation is used to mirror the 12-byte structure of key storage. Thus, every byte of the dedicated memory area can be the start of a key.

**Example:** If the key loading cycle starts at the last byte address of an EEPROM block, (for example, key byte 0 is stored at 12Fh), the next bytes are stored in the next EEPROM block, for example, key byte 1 is stored at 130h, byte 2 at 131h up to byte 11 at 13Ah.

Based on the 384 bytes of memory and a single key needing 12 bytes, then up to 32 different keys can be stored in the EEPROM.

**Remark:** It is not possible to load a key exceeding the EEPROM byte location 1FFh.

**9.3 FIFO buffer**

An 8 × 64 bit FIFO buffer is used in the MFRC530 to act as a parallel-to-parallel converter. It buffers both the input and output data streams between the microprocessor and the internal circuitry of the MFRC530. This makes it possible to manage data streams up to 64 bytes long without needing to take timing constraints into account.

**9.3.1 Accessing the FIFO buffer**

**9.3.1.1 Access rules**

The FIFO buffer input and output data bus is connected to the FIFOData register. Writing to this register stores one byte in the FIFO buffer and increments the FIFO buffer write pointer. Reading from this register shows the FIFO buffer contents stored at the FIFO buffer read pointer and increments the FIFO buffer read pointer. The distance between the write and read pointer can be obtained by reading the FIFOLength register.



When the microprocessor starts a command, the MFRC530 can still access the FIFO buffer while the command is running. Only one FIFO buffer has been implemented which is used for input and output. Therefore, the microprocessor must ensure that there are no inadvertent FIFO buffer accesses. [Table 18](#) gives an overview of FIFO buffer access during command processing.

**Table 18. FIFO buffer access**

| Active command | FIFO buffer |         | Remark  |
|----------------|-------------|---------|---|
|                | μp Write    | μp Read |   |
| StartUp        | -           | -       |   |
| Idle           | -           | -       |   |
| Transmit       | yes         | -       |   |
| Receive        | -           | yes     |   |
| Transceive     | yes         | yes     | the microprocessor has to know the state of the command (transmitting or receiving) |
| WriteE2        | yes         | -       |   |
| ReadE2         | yes         | yes     | the microprocessor has to prepare the arguments, afterwards only reading is allowed |
| LoadKeyE2      | yes         | -       |   |
| LoadKey        | yes         | -       |   |
| Authent1       | yes         | -       |   |
| Authent2       | -           | -       |   |
| LoadConfig     | yes         | -       |   |
| CalcCRC        | yes         | -       |   |

### 9.3.2 Controlling the FIFO buffer

In addition to writing to and reading from the FIFO buffer, the FIFO buffer pointers can be reset using the FlushFIFO bit. This changes the FIFOLength[6:0] value to zero, bit FIFOOvfl is cleared and the stored bytes are no longer accessible. This enables the FIFO buffer to be written with another 64 bytes of data.

### 9.3.3 FIFO buffer status information

The microprocessor can get the following FIFO buffer status data:

- the number of bytes stored in the FIFO buffer: bits FIFOLength[6:0]
- the FIFO buffer full warning: bit HiAlert
- the FIFO buffer empty warning: bit LoAlert
- the FIFO buffer overflow warning: bit FIFOOvfl.

**Remark:** Setting the FlushFIFO bit clears the FIFOOvfl bit.

The MFRC530 can generate an interrupt signal when:

- bit LoAlertIRQ is set to logic 1 and bit LoAlert = logic 1, pin IRQ is activated.
- bit HiAlertIRQ is set to logic 1 and bit HiAlert = logic 1, pin IRQ activated.

The HiAlert flag bit is set to logic 1 only when the WaterLevel[5:0] bits or less can be stored in the FIFO buffer. The trigger is generated by [Equation 1](#):

$$HiAlert = (64 - FIFOLength) \leq WaterLevel \quad (1)$$

The LoAlert flag bit is set to logic 1 when the FIFOLevel register's WaterLevel[5:0] bits or less are stored in the FIFO buffer. The trigger is generated by [Equation 2](#):

$$LoAlert = FIFOLength \leq WaterLevel \quad (2)$$

### 9.3.4 FIFO buffer registers and flags

[Table 19](#) shows the related FIFO buffer flags in alphabetic order.

**Table 19. Associated FIFO buffer registers and flags**

| Flags           | Register name | Bit    | Register address |
|-----------------|---------------|--------|------------------|
| FIFOLength[6:0] | FIFOLength    | 6 to 0 | 04h              |
| FIFOOvfl        | ErrorFlag     | 4      | 0Ah              |
| FlushFIFO       | Control       | 0      | 09h              |
| HiAlert         | PrimaryStatus | 1      | 03h              |
| HiAlertIEn      | InterruptEn   | 1      | 06h              |
| HiAlertIRq      | InterruptRq   | 1      | 07h              |
| LoAlert         | PrimaryStatus | 0      | 03h              |
| LoAlertIEn      | InterruptEn   | 0      | 06h              |
| LoAlertIRq      | InterruptRq   | 0      | 07h              |
| WaterLevel[5:0] | FIFOLevel     | 5 to 0 | 29h              |

## 9.4 Interrupt request system

The MFRC530 indicates interrupt events by setting the PrimaryStatus register bit IRq (see [Section 10.5.1.4 “PrimaryStatus register” on page 47](#)) and activating pin IRQ. The signal on pin IRQ can be used to interrupt the microprocessor using its interrupt handling capabilities ensuring efficient microprocessor software.

### 9.4.1 Interrupt sources overview

[Table 20](#) shows the integrated interrupt flags, related source and setting condition. The interrupt TimerIRq flag bit indicates an interrupt set by the timer unit. Bit TimerIRq is set when the timer decrements from one down to zero (bit TAutoRestart disabled) or from one to the TReLoadValue[7:0] with bit TAutoRestart enabled.

Bit TxIRq indicates interrupts from different sources and is set as follows:

- the transmitter automatically sets the bit TxIRq interrupt when it is active and its state changes from sending data to transmitting the end of frame pattern
- the CRC coprocessor sets the bit TxIRq after all data from the FIFO buffer has been processed indicated by bit CRCReady = logic 1
- when EEPROM programming is finished, the bit TxIRq is set and is indicated by bit E2Ready = logic 1

The RxIRq flag bit indicates an interrupt when the end of the received data is detected. The IdleIRq flag bit is set when a command finishes and the content of the Command register changes to Idle.

When the FIFO buffer reaches the HIGH-level indicated by the WaterLevel[5:0] value (see [Section 9.3.3 on page 17](#)) and bit HiAlert = logic 1, then the HiAlertIRq flag bit is set to logic 1.

When the FIFO buffer reaches the LOW-level indicated by the WaterLevel[5:0] value (see [Section 9.3.3](#)) and bit LoAlert = logic 1, then LoAlertIRq flag bit is set to logic 1.

**Table 20. Interrupt sources**

| Interrupt flag | Interrupt source | Trigger action                                    |
|----------------|------------------|---|
| TimerIRq       | timer unit       | timer counts from 1 to 0                          |
| TxIRq          | transmitter      | a data stream, transmitted to the card, ends      |
|                | CRC coprocessor  | all data from the FIFO buffer has been processed  |
|                | EEPROM           | all data from the FIFO buffer has been programmed |
| RxIRq          | receiver         | a data stream, received from the card, ends       |
| IdleIRq        | Command register | command execution finishes                        |
| HiAlertIRq     | FIFO buffer      | FIFO buffer is full                               |
| LoAlertIRq     | FIFO buffer      | FIFO buffer is empty                              |

## 9.4.2 Interrupt request handling

### 9.4.2.1 Controlling interrupts and getting their status

The MFRC530 informs the microprocessor about the interrupt request source by setting the relevant bit in the InterruptRq register. The relevance of each interrupt request bit as source for an interrupt can be masked by the InterruptEn register interrupt enable bits.

**Table 21. Interrupt control registers**

| Register    | Bit 7  | Bit 6    | Bit 5    | Bit 4 | Bit 3 | Bit 2   | Bit 1      | Bit 0      |
|-------------|--------|----------|----------|-------|-------|---------|------------|------------|
| InterruptEn | SetIEn | reserved | TimerIEn | TxIEn | RxIEn | IdleIEn | HiAlertIEn | LoAlertIEn |
| InterruptRq | SetIRq | reserved | TimerIRq | TxIRq | RxIRq | IdleIRq | HiAlertIRq | LoAlertIRq |

If an interrupt request flag is set to logic 1 (showing that an interrupt request is pending) and the corresponding interrupt enable flag is set, the PrimaryStatus register IRq flag bit is set to logic 1. Different interrupt sources can activate simultaneously and because of this, all interrupt request bits are OR'ed, coupled to the IRq flag and then forwarded to pin IRQ.

### 9.4.2.2 Accessing the interrupt registers

The interrupt request bits are automatically set by the MFRC530's internal state machines. In addition, the microprocessor can also set or clear the interrupt request bits as required.

A special implementation of the InterruptRq and InterruptEn registers enables changing an individual bit status without influencing any other bits. If an interrupt register is set to logic 1, bit Setlxx and the specific bit must both be set to logic 1 at the same time. Vice versa, if a specific interrupt flag is cleared, zero must be written to the Setlxx and the interrupt register address must be set to logic 1 at the same time.

If a content bit is not changed during the setting or clearing phase, zero must be written to the specific bit location.

**Example:** Writing 3Fh to the InterruptRq register clears all bits. SetIRq is set to logic 0 while all other bits are set to logic 1. Writing 81h to the InterruptRq register sets LoAlertIRq to logic 1 and leaves all other bits unchanged.

### 9.4.3 Configuration of pin IRQ

The logic level of the IRq flag bit is visible on pin IRQ. The signal on pin IRQ can also be controlled using the following IRQPinConfig register bits.

- bit IRQInv: the signal on pin IRQ is equal to the logic level of bit IRq when this bit is set to logic 0. When set to logic 1, the signal on pin IRQ is inverted with respect to bit IRq.
- bit IRQPushPull: when set to logic 1, pin IRQ has CMOS output characteristics. When it is set to logic 0, it is an open-drain output which requires an external resistor to achieve a HIGH-level at pin IRQ.

**Remark:** During the reset phase (see [Section 9.7.2 on page 26](#)) bit IRQInv is set to logic 1 and bit IRQPushPull is set to logic 0. This results in a high-impedance on pin IRQ.

### 9.4.4 Register overview interrupt request system

[Table 22](#) shows the related interrupt request system flags in alphabetically.

**Table 22. Associated Interrupt request system registers and flags**

| Flags       | Register name | Bit | Register address |
|-------------|---------------|-----|------------------|
| HiAlertEn   | InterruptEn   | 1   | 06h              |
| HiAlertIRq  | InterruptRq   | 1   | 07h              |
| IdleEn      | InterruptEn   | 2   | 06h              |
| IdleIRq     | InterruptRq   | 2   | 07h              |
| IRq         | PrimaryStatus | 3   | 03h              |
| IRQInv      | IRQPinConfig  | 1   | 07h              |
| IRQPushPull | IRQPinConfig  | 0   | 07h              |
| LoAlertEn   | InterruptEn   | 0   | 06h              |
| LoAlertIRq  | InterruptRq   | 0   | 07h              |
| RxEn        | InterruptEn   | 3   | 06h              |
| RxIRq       | InterruptRq   | 3   | 07h              |
| SetEn       | InterruptEn   | 7   | 06h              |
| SetIRq      | InterruptRq   | 7   | 07h              |
| TimerEn     | InterruptEn   | 5   | 06h              |
| TimerIRq    | InterruptRq   | 5   | 07h              |
| TxEn        | InterruptEn   | 4   | 06h              |
| TxIRq       | InterruptRq   | 4   | 07h              |

## 9.5 Timer unit

The timer derives its clock from the 13.56 MHz on-board chip clock. The microprocessor can use this timer to manage timing-relevant tasks.

The timer unit may be used in one of the following configurations:

- Timeout counter
- WatchDog counter
- Stopwatch
- Programmable one shot
- Periodical trigger

The timer unit can be used to measure the time interval between two events or to indicate that a specific timed event occurred. The timer is triggered by events but does not influence any event (e.g. a time-out during data receiving does not automatically influence the reception process). Several timer related flags can be set and these flags can be used to generate an interrupt.

### 9.5.1 Timer unit implementation

#### 9.5.1.1 Timer unit block diagram

[Figure 8](#) shows the block diagram of the timer module.

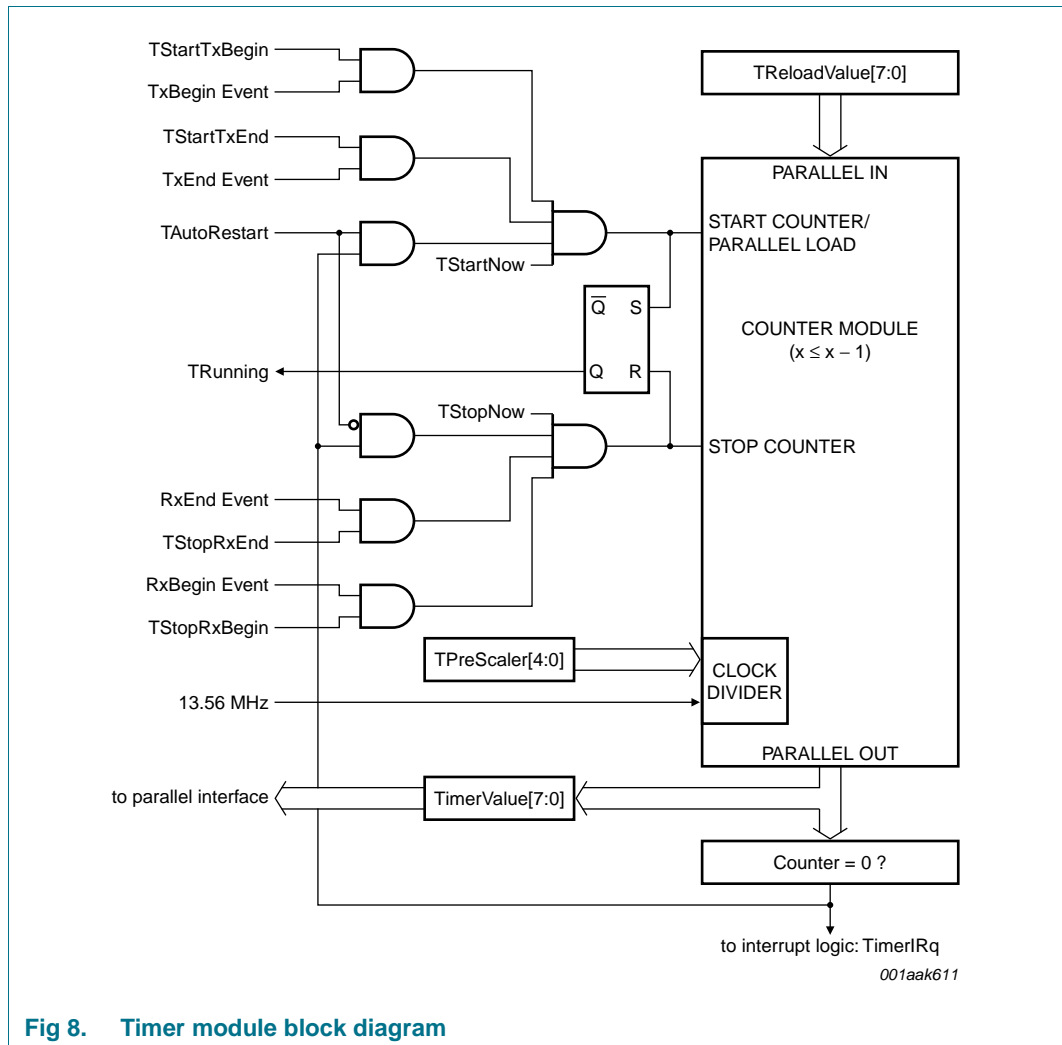


Fig 8. Timer module block diagram

The timer unit is designed, so that events when combined with enabling flags start or stop the counter. For example, setting bit TStartTxBegin = logic 1 enables control of received data with the timer unit. In addition, the first received bit is indicated by the TxBegin event. This combination starts the counter at the defined TReloadValue[7:0].

The timer stops automatically when the counter value is equal to zero or if a defined stop event happens.

### 9.5.1.2 Controlling the timer unit

The main part of the timer unit is a down counter. As long as the down counter value is not zero, it decrements its value with each timer clock cycle.

If the TAutoRestart flag is enabled, the timer does not decrement down to zero. On reaching value 1, the timer reloads the next clock function with the TReloadValue[7:0].

The timer is started immediately by loading a value from the TimerReload register into the counter module.

This is activated by one of the following events:

- transmission of the first bit to the card (TxBegin event) with bit TStartTxBegin = logic 1
- transmission of the last bit to the card (TxEnd event) with bit TStartTxEnd = logic 1
- bit TStartNow is set to logic 1 by the microprocessor

**Remark:** Every start event reloads the timer from the TimerReload register which re-triggers the timer unit.

The timer can be configured to stop on one of the following events:

- receipt of the first valid bit from the card (RxBegin event) with bit TStopRxBegin = logic 1
- receipt of the last bit from the card (RxEnd event) with bit TStopRxEnd = logic 1
- the counter module has decremented down to zero and bit TAutoRestart = logic 0
- bit TStopNow is set to logic 1 by the microprocessor.

Loading a new value, e.g. zero, into the TimerReload register or changing the timer unit while it is counting will not immediately influence the counter. In both cases, this is because this register only affects the counter content after a start event.

If the counter is stopped when bit TStopNow is set, no TimerIRq is flagged.

#### 9.5.1.3 Timer unit clock and period

The timer unit clock is derived from the 13.56 MHz on-board chip clock using the programmable divider. Clock selection is made using the TimerClock register TPreScaler[4:0] bits based on [Equation 3](#):

$$f_{TimerClock} = \frac{1}{T_{TimerClock}} = \frac{2^{TPreScaler}}{13.56} [MHz] \quad (3)$$

The values for the TPreScaler[4:0] bits are between 0 and 21 which results in a minimum periodic time ( $T_{TimerClock}$ ) of between 74 ns and 150 ms.

The time period elapsed since the last start event is calculated using [Equation 4](#):

$$t_{Timer} = \frac{TReLoadValue - TimerValue}{f_{TimerClock}} [s] \quad (4)$$

This results in a minimum time period ( $t_{Timer}$ ) of between 74 ns and 40 s.

#### 9.5.1.4 Timer unit status

The SecondaryStatus register's TRunning bit shows the timer's status. Configured start events start the timer at the TReloadValue[7:0] and change the status flag TRunning to logic 1. Conversely, configured stop events stop the timer and set the TRunning status flag to logic 0. As long as status flag TRunning is set to logic 1, the TimerValue register changes on the next timer unit clock cycle.

The TimerValue[7:0] bits can be read directly from the TimerValue register.

## 9.5.2 Using the timer unit functions

### 9.5.2.1 Time-out and WatchDog counters

After starting the timer using TReloadValue[7:0], the timer unit decrements the TimerValue register beginning with a given start event. If a given stop event occurs, such as a bit being received from the card, the timer unit stops without generating an interrupt.

If a stop event does not occur, such as the card not answering within the expected time, the timer unit decrements down to zero and generates a timer interrupt request. This signals to the microprocessor the expected event has not occurred within the given time ( $t_{Timer}$ ).

### 9.5.2.2 Stopwatch

The time ( $t_{Timer}$ ) between a start and stop event is measured by the microprocessor using the timer unit. Setting the TReloadValue register triggers the timer which in turn, starts to decrement. If the defined stop event occurs, the timer stops. The time between start and stop is calculated by the microprocessor using [Equation 5](#) when the time does not decrement down to zero.

$$\Delta t = (TReload_{value} - TimerValue) \times t_{Timer} \quad (5)$$

### 9.5.2.3 Programmable one shot timer and periodic trigger

**Programmable one shot timer:** The microprocessor starts the timer unit and waits for the timer interrupt. The interrupt occurs after the time specified by  $t_{Timer}$ .

**Periodic trigger:** If the microprocessor sets the TAutoRestart bit, it generates an interrupt request after every  $t_{Timer}$  cycle.

## 9.5.3 Timer unit registers

[Table 23](#) shows the related flags of the timer unit in alphabetical order.

**Table 23. Associated timer unit registers and flags**

| Flags             | Register name   | Bit    | Register address |
|-------------------|-----------------|--------|------------------|
| TAutoRestart      | TimerClock      | 5      | 2Ah              |
| TimerValue[7:0]   | TimerValue      | 7 to 0 | 0Ch              |
| TReloadValue[7:0] | TimerReload     | 7 to 0 | 2Ch              |
| TPreScaler[4:0]   | TimerClock      | 4 to 0 | 2Ah              |
| TRunning          | SecondaryStatus | 7      | 05h              |
| TStartNow         | Control         | 1      | 09h              |
| TStartTxBegin     | TimerControl    | 0      | 2Bh              |
| TStartTxEnd       | TimerControl    | 1      | 2Bh              |
| TStopNow          | Control         | 2      | 09h              |
| TStopRxBegin      | TimerControl    | 2      | 2Bh              |
| TStopRxEnd        | TimerControl    | 3      | 2Bh              |



## 9.6 Power reduction modes

### 9.6.1 Hard power-down

Hard power-down is enabled when pin RSTPD is HIGH. This turns off all internal current sinks including the oscillator. All digital input buffers are separated from the input pads and defined internally (except pin RSTPD itself). The output pins are frozen at a given value. The status of all pins during a hard power-down is shown in [Table 24](#).

**Table 24. Signal on pins during Hard power-down**

| Symbol   | Pin      | Type | Description   |
|----------|----------|------|---|
| OSCIN    | 1        | I    | not separated from input, pulled to AVSS                                      |
| IRQ      | 2        | O    | high-impedance  |
| MFIN     | 3        | I    | separated from input  |
| MFOUT    | 4        | O    | LOW   |
| TX1      | 5        | O    | HIGH, if bit TX1RFEn = logic 1<br>LOW, if bit TX1RFEn = logic 0               |
| TX2      | 7        | O    | HIGH, only if bit TX2RFEn = logic 1 and bit TX2Inv = logic 0<br>otherwise LOW |
| NCS      | 9        | I    | separated from input  |
| NWR      | 10       | I    | separated from input  |
| NRD      | 11       | I    | separated from input  |
| D0 to D7 | 13 to 20 | I/O  | separated from input  |
| ALE      | 21       | I    | separated from input  |
| A0       | 22       | I/O  | separated from input  |
| A1       | 23       | I    | separated from input  |
| A2       | 24       | I    | separated from input  |
| AUX      | 27       | O    | high-impedance  |
| RX       | 29       | I    | not changed   |
| VMID     | 30       | A    | pulled to V <sub>D<sub>DA</sub></sub>   |
| RSTPD    | 31       | I    | not changed   |
| OSCOUT   | 32       | O    | HIGH  |

### 9.6.2 Soft power-down mode

Soft power-down mode is entered immediately using the Control register bit PowerDown. All internal current sinks, including the oscillator buffer, are switched off. The digital input buffers are not separated from the input pads and keep their functionality. In addition, the digital output pins do not change their state.

After resetting the Control register bit PowerDown, the bit indicating Soft power-down mode is only cleared after 512 clock cycles. Resetting it does not immediately clear it. The PowerDown bit is automatically cleared when the Soft power-down mode is exited.

**Remark:** When the internal oscillator is used, time ( $t_{osc}$ ) is required for the oscillator to become stable. This is because the internal oscillator is supplied by V<sub>D<sub>DA</sub></sub> and any clock cycles will not be detected by the internal logic until V<sub>D<sub>DA</sub></sub> is stable.

**9.6.3 Standby mode**

The Standby mode is immediately entered when the Control register StandBy bit is set. All internal current sinks, including the internal digital clock buffer are switched off. However, the oscillator buffer is not switched off.

The digital input buffers are not separated by the input pads, keeping their functionality and the digital output pins do not change their state. In addition, the oscillator does not need time to wake-up.

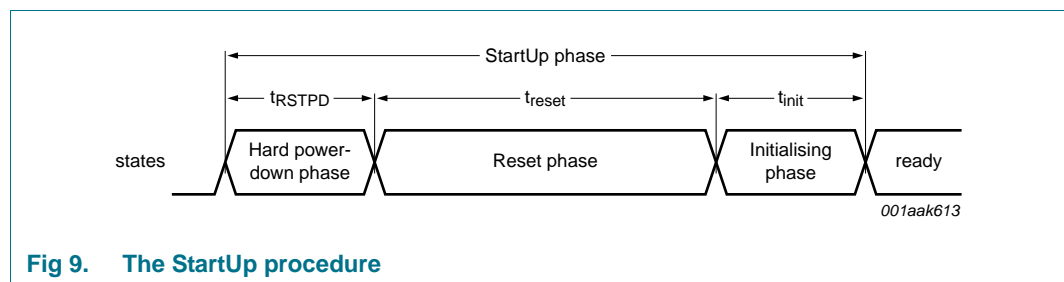
After resetting the Control register StandBy bit, it takes four clock cycles on pin OSCIN for Standby mode to exit. Resetting bit StandBy does not immediately clear it. It is automatically cleared when the Standby mode is exited.

**9.6.4 Automatic receiver power-down**

It is a power saving feature to switch off the receiver circuit when it is not needed. Setting bit RxAutoPD = logic 1, automatically powers down the receiver when it is not in use. Setting bit RxAutoPD = logic 0, keeps the receiver continuously powered up.

**9.7 StartUp phase**

The events executed during the StartUp phase are shown in [Figure 9](#).



**Fig 9. The StartUp procedure**

**9.7.1 Hard power-down phase**

The hard power-down phase is active during the following cases:

- a Power-On Reset (POR) caused by power-up on pins DVDD or AVDD activated when  $V_{DD}$  or  $V_{DDA}$  is below the relevant analog/digital reset threshold.
- a HIGH-level on pin RSTPD which is active while pin RSTPD is HIGH. The HIGH level period on pin RSTPD must be at least 100  $\mu$ s ( $t_{PD} \geq 100 \mu$ s). Shorter phases will not necessarily result in the reset phase ( $t_{reset}$ ). The rising or falling edge slew rate on pin RSTPD is not critical because pin RSTPD is a Schmitt trigger input.

**9.7.2 Reset phase**

The reset phase automatically follows the Hard power-down. Once the oscillator is running stably, the reset phase takes 512 clock cycles. During the reset phase, some register bits are preset by hardware. The respective reset values are given in the description of each register (see [Section 10.5 on page 46](#)).

**Remark:** When the internal oscillator is used, time ( $t_{osc}$ ) is required for the oscillator to become stable. This is because the internal oscillator is supplied by  $V_{DDA}$  and any clock cycles will not be detected by the internal logic until  $V_{DDA}$  is stable.

**9.7.3 Initialization phase**

The initialization phase automatically follows the reset phase and takes 128 clock cycles. During the initializing phase the content of the EEPROM blocks 1 and 2 is copied into the register subaddresses 10h to 2Fh (see [Section 9.2.2 on page 13](#)).

**Remark:** During the production test, the MFRC530 is initialized with default configuration values. This reduces the microprocessor’s configuration time to a minimum.

**9.7.4 Initializing the parallel interface type**

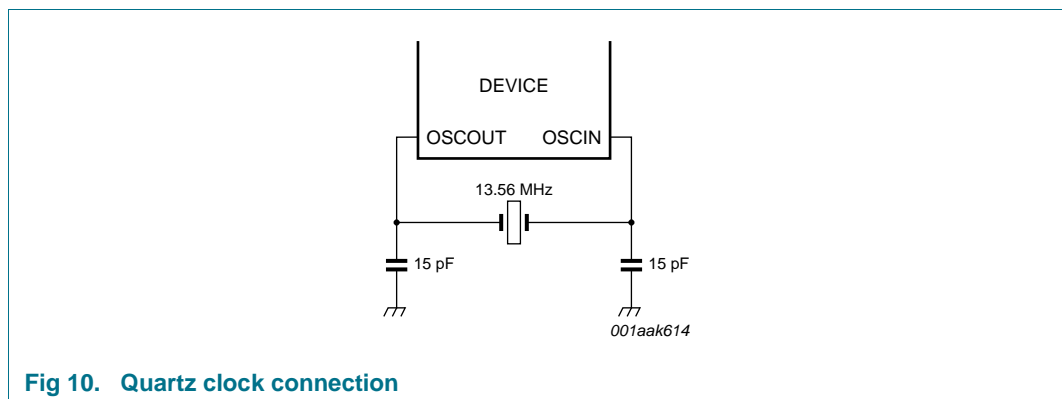
A different initialization sequence is used for each microprocessor. This enables detection of the correct microprocessor interface type and synchronization of the microprocessor’s and the MFRC530’s start-up. See [Section 9.1.3 on page 8](#) for detailed information on the different connections for each microprocessor interface type.

During StartUp phase, the command value is set to 3Fh once the oscillator attains clock frequency stability at an amplitude of > 90 % of the nominal 13.56 MHz clock frequency. At the end of the initialization phase, the MFRC530 automatically switches to idle and the command value changes to 00h.

To ensure correct detection of the microprocessor interface, the following sequence is executed:

- the Command register is read until the 6-bit register value is 00h. On reading the 00h value, the internal initialization phase is complete and the MFRC530 is ready to be controlled
- write 80h to the Page register to initialize the microprocessor interface
- read the Command register. If it returns a value of 00h, the microprocessor interface was successfully initialized
- write 00h to the Page registers to activate linear addressing mode.

**9.8 Oscillator circuit**



**Fig 10. Quartz clock connection**

The clock applied to the MFRC530 acts as a time basis for the synchronous system encoder and decoder. The stability of the clock frequency is an important factor for correct operation. To obtain highest performance, clock jitter must be as small as possible. This is best achieved by using the internal oscillator buffer with the recommended circuitry.

If an external clock source is used, the clock signal must be applied to pin OSCIN. In this case, be very careful in optimizing clock duty cycle and clock jitter. Ensure the clock quality has been verified. It must meet the specifications described in [Section 13.4.5 on page 95](#).

**Remark:** We do not recommend using an external clock source.

## 9.9 Transmitter pins TX1 and TX2

The signal on pins TX1 and TX2 is the 13.56 MHz energy carrier modulated by an envelope signal. It can be used to drive an antenna directly, using minimal passive components for matching and filtering (see [Section 15.1 on page 96](#)). To enable this, the output circuitry is designed with a very low-impedance source resistance. The TxControl register is used to control the TX1 and TX2 signals.

### 9.9.1 Configuring pins TX1 and TX2

TX1 pin configurations are described in [Table 25](#).

**Table 25. Pin TX1 configurations**

| TxControl register configuration | Envelope | TX1 signal               |
|----------------------------------|----------|--------------------------|
| TX1RFEn                          |          |                          |
| 0                                | X        | LOW (GND)                |
| 1                                | 0        | LOW                      |
| 1                                | 1        | 13.56 MHz energy carrier |

TX2 pin configurations are described in [Table 26](#).

**Table 26. Pin TX2 configurations**

| TxControl register configuration |       |        | Envelope | TX2 signal  |
|----------------------------------|-------|--------|----------|---|
| TX2RFEn                          | TX2CW | TX2Inv |          |   |
| 0                                | X     | X      | X        | LOW   |
| 1                                | 0     | 0      | 0        | LOW   |
| 1                                | 0     | 0      | 1        | 13.56 MHz carrier frequency                                   |
| 1                                | 0     | 1      | 0        | HIGH  |
| 1                                | 0     | 1      | 1        | 13.56 MHz carrier frequency, 180° phase-shift relative to TX1 |
| 1                                | 1     | 0      | X        | 13.56 MHz carrier frequency                                   |
| 1                                | 1     | 1      | X        | 13.56 MHz carrier frequency, 180° phase-shift relative to TX1 |

### 9.9.2 Antenna operating distance versus power consumption

Using different antenna matching circuits (by varying the supply voltage on the antenna driver supply pin TVDD), it is possible to find the trade-off between maximum effective operating distance and power consumption. Different antenna matching circuits are described in the Application note *"MIFARE Design Matching Circuit and Antennas"*.

### 9.9.3 Antenna driver output source resistance

The output source conductance of pins TX1 and TX2 can be adjusted between 1  $\Omega$  and 100  $\Omega$  using the CwConductance register GsCfgCW[5:0] bits.

The output source conductance of pins TX1 and TX2 during the modulation phase can be adjusted between 1  $\Omega$  and 100  $\Omega$  using the ModConductance register GsCfgMod[5:0] bits.

The values are relative to the reference resistance ( $R_{S(ref)}$ ) which is measured during the production test and stored in the MFRC530 EEPROM. It can be read from the product information field (see [Section 9.2.1 on page 13](#)). The electrical specification can be found in [Section 13.3.3 on page 91](#).

#### 9.9.3.1 Source resistance table

**Table 27. TX1 and TX2 source resistance of n-channel driver transistor against GsCfgCW or GsCfgMod**

MANT = Mantissa; EXP = Exponent.

| GsCfgCW,<br>GsCfgMod<br>(decimal) | EXP <sub>GsCfgCW</sub> ,<br>EXP <sub>GsCfgMod</sub><br>(decimal) | MANT <sub>GsCfgCW</sub> ,<br>MANT <sub>GsCfgMod</sub><br>(decimal) | R <sub>S(ref)</sub><br>( $\Omega$ ) | GsCfgCW,<br>GsCfgMod<br>(decimal) | EXP <sub>GsCfgCW</sub> ,<br>EXP <sub>GsCfgMod</sub><br>(decimal) | MANT <sub>GsCfgCW</sub> ,<br>MANT <sub>GsCfgMod</sub><br>(decimal) | R <sub>S(ref)</sub><br>( $\Omega$ ) |
|-----------------------------------|--|--|-------------------------------------|-----------------------------------|--|--|-------------------------------------|
| 0                                 | 0  | 0  | -                                   | 24                                | 1  | 8  | 0.0652                              |
| 16                                | 1  | 0  | -                                   | 25                                | 1  | 9  | 0.0580                              |
| 32                                | 2  | 0  | -                                   | 37                                | 2  | 5  | 0.0541                              |
| 48                                | 3  | 0  | -                                   | 26                                | 1  | 10   | 0.0522                              |
| 1                                 | 0  | 1  | 1.0000                              | 27                                | 1  | 11   | 0.0474                              |
| 17                                | 1  | 1  | 0.5217                              | 51                                | 3  | 3  | 0.0467                              |
| 2                                 | 0  | 2  | 0.5000                              | 38                                | 2  | 6  | 0.0450                              |
| 3                                 | 0  | 3  | 0.3333                              | 28                                | 1  | 12   | 0.0435                              |
| 33                                | 2  | 1  | 0.2703                              | 29                                | 1  | 13   | 0.0401                              |
| 18                                | 1  | 2  | 0.2609                              | 39                                | 2  | 7  | 0.0386                              |
| 4                                 | 0  | 4  | 0.2500                              | 30                                | 1  | 14   | 0.0373                              |
| 5                                 | 0  | 5  | 0.2000                              | 52                                | 3  | 4  | 0.0350                              |
| 19                                | 1  | 3  | 0.1739                              | 31                                | 1  | 15   | 0.0348                              |
| 6                                 | 0  | 6  | 0.1667                              | 40                                | 2  | 8  | 0.0338                              |
| 7                                 | 0  | 7  | 0.1429                              | 41                                | 2  | 9  | 0.0300                              |
| 49                                | 3  | 1  | 0.1402                              | 53                                | 3  | 5  | 0.0280                              |
| 34                                | 2  | 2  | 0.1351                              | 42                                | 2  | 10   | 0.0270                              |
| 20                                | 1  | 4  | 0.1304                              | 43                                | 2  | 11   | 0.0246                              |
| 8                                 | 0  | 8  | 0.1250                              | 54                                | 3  | 6  | 0.0234                              |
| 9                                 | 0  | 9  | 0.1111                              | 44                                | 2  | 12   | 0.0225                              |
| 21                                | 1  | 5  | 0.1043                              | 45                                | 2  | 13   | 0.0208                              |
| 10                                | 0  | 10   | 0.1000                              | 55                                | 3  | 7  | 0.0200                              |
| 11                                | 0  | 11   | 0.0909                              | 46                                | 2  | 14   | 0.0193                              |
| 35                                | 2  | 3  | 0.0901                              | 47                                | 2  | 15   | 0.0180                              |
| 22                                | 1  | 6  | 0.0870                              | 56                                | 3  | 8  | 0.0175                              |
| 12                                | 0  | 12   | 0.0833                              | 57                                | 3  | 9  | 0.0156                              |
| 13                                | 0  | 13   | 0.0769                              | 58                                | 3  | 10   | 0.0140                              |

**Table 27. TX1 and TX2 source resistance of n-channel driver transistor against GsCfgCW or GsCfgMod ...continued**  
 MANT = Mantissa; EXP= Exponent.

| GsCfgCW,<br>GsCfgMod<br>(decimal) | EXP <sub>GsCfgCW</sub> ,<br>EXP <sub>GsCfgMod</sub><br>(decimal) | MANT <sub>GsCfgCW</sub> ,<br>MANT <sub>GsCfgMod</sub><br>(decimal) | R <sub>S(ref)</sub><br>(Ω) | GsCfgCW,<br>GsCfgMod<br>(decimal) | EXP <sub>GsCfgCW</sub> ,<br>EXP <sub>GsCfgMod</sub><br>(decimal) | MANT <sub>GsCfgCW</sub> ,<br>MANT <sub>GsCfgMod</sub><br>(decimal) | R <sub>S(ref)</sub><br>(Ω) |
|-----------------------------------|--|--|----------------------------|-----------------------------------|--|--|----------------------------|
| 23                                | 1  | 7  | 0.0745                     | 59                                | 3  | 11   | 0.0127                     |
| 14                                | 0  | 14   | 0.0714                     | 60                                | 3  | 12   | 0.0117                     |
| 50                                | 3  | 2  | 0.0701                     | 61                                | 3  | 13   | 0.0108                     |
| 36                                | 2  | 4  | 0.0676                     | 62                                | 3  | 14   | 0.0100                     |
| 15                                | 0  | 15   | 0.0667                     | 63                                | 3  | 15   | 0.0093                     |

**9.9.3.2 Calculating the relative source resistance**

The reference source resistance R<sub>S(ref)</sub> can be calculated using [Equation 6](#).

$$R_{S(ref)} = \frac{I}{MANT_{GsCfgCW} \cdot \left(\frac{77}{40}\right)^{EXP_{GsCfgCW}}} \tag{6}$$

The reference source resistance (R<sub>S(ref)</sub>) during the modulation phase can be calculated using ModConductance register’s GsCfgMod[5:0].

**9.9.3.3 Calculating the effective source resistance**

**Wiring resistance (R<sub>S(wire)</sub>):** Wiring and bonding add a constant offset to the driver resistance that is relevant when pins TX1 and TX2 are switched to low-impedance. The additional resistance for pin TX1 (R<sub>S(wire)TX1</sub>) can be set approximately as shown in [Equation 7](#).

$$R_{S(wire)TX1} \approx 500 \text{ m}\Omega \tag{7}$$

**Effective resistance (R<sub>Sx</sub>):** The source resistances of the driver transistors (RsMaxP byte) read from the Product Information Field (see [Section 9.2.1 on page 13](#)) are measured during the production test with CwConductance register’s GsCfgCW[5:0] = 01h.

To calculate the driver resistance for a specific value set in GsCfgMod[5:0], use [Equation 8](#).

$$R_{Sx} = (R_{S(ref)maxP} - R_{S(wire)TX1}) \cdot R_{S(rel)} + R_{S(wire)TX1} \tag{8}$$

**9.9.4 Pulse width**

The envelope carries the data signal information that is transmitted to the card. It is an encoded data signal based on the Miller code. In addition, each pause of the Miller encoded signal is again encoded as a pulse of a fixed width. The width of the pulse is adjusted using the ModWidth register. The pulse width (t<sub>w</sub>) is calculated using [Equation 9](#) where the frequency constant (f<sub>clk</sub>) = 13.56 MHz.

$$t_w = 2 \frac{ModWidth + 1}{f_{clk}} \tag{9}$$

**9.10 Receiver circuit**

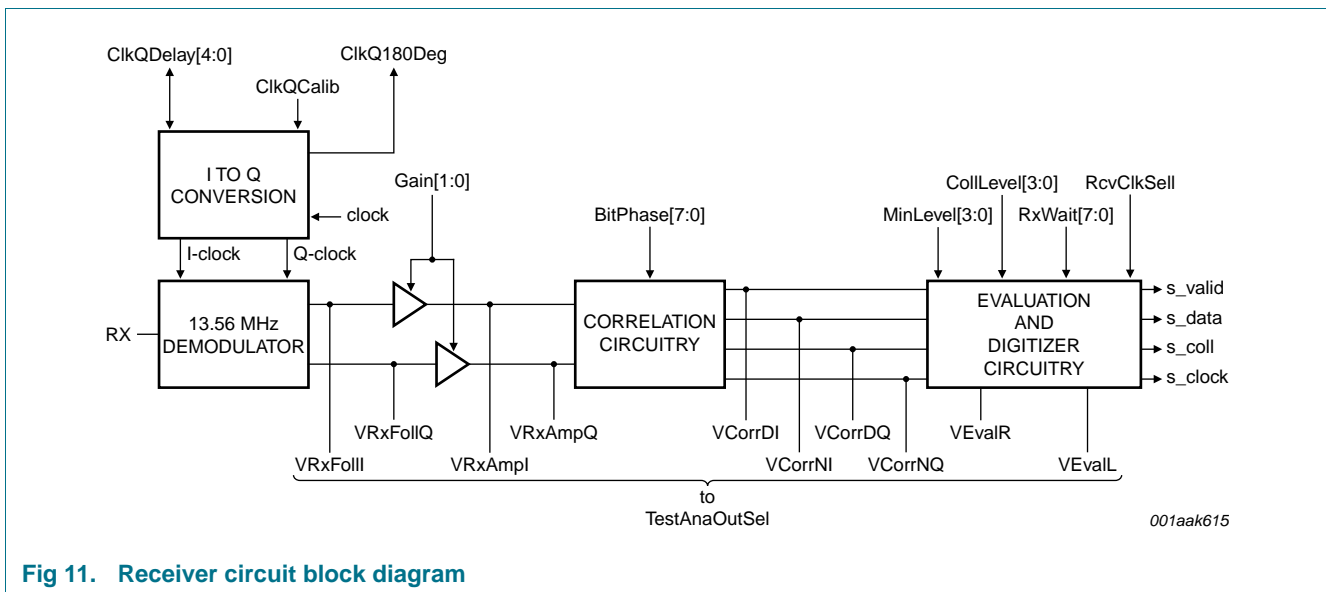
The MFRC530 uses an integrated quadrature demodulation circuit enabling it to extract the ISO/IEC 14443 A compliant subcarrier from the 13.56 MHz ASK modulated signal applied to pin RX.

The quadrature demodulator uses two different clocks (Q-clock and I-clock) with a phase-shift of 90° between them. Both resulting subcarrier signals are amplified, filtered and forwarded to the correlation circuitry. The correlation results are evaluated, digitized and then passed to the digital circuitry. Various adjustments can be made to obtain optimum performance for all processing units.

**9.10.1 Receiver circuit block diagram**

[Figure 11](#) shows the block diagram of the receiver circuit. The receiving process can be broken down in to several steps. Quadrature demodulation of the 13.56 MHz carrier signal is performed. To achieve the optimum performance, automatic Q-clock calibration is recommended (see [Section 9.10.2.1 on page 32](#)).

The demodulated signal is amplified by an adjustable amplifier. A correlation circuit calculates the degree of similarity between the expected and the received signal. The BitPhase register enables correlation interval position alignment with the received signal's bit grid. In the evaluation and digitizer circuitry, the valid bits are detected and the digital results are sent to the FIFO buffer. Several tuning steps are possible for this circuit.



**Fig 11. Receiver circuit block diagram**

The signal can be observed on its way through the receiver as shown in [Figure 11](#). One signal at a time can be routed to pin AUX using the TestAnaSelect register as described in [Section 15.2.2 on page 101](#).

**9.10.2 Receiver operation**

In general, the default settings programmed in the StartUp initialization file are suitable for use with the MFRC530 to MIFARE card data communication. However, in some environments specific user settings will achieve better performance.

### 9.10.2.1 Automatic Q-clock calibration

The quadrature demodulation concept of the receiver generates a phase signal (I-clock) and a 90° phase-shifted quadrature signal (Q-clock). To achieve the optimum demodulator performance, the Q-clock and the I-clock must be phase-shifted by 90°. After the reset phase, a calibration procedure is automatically performed.

Automatic calibration can be set-up to execute at the end of each Transceive command if bit ClkQCalib = logic 0. Setting bit ClkQCalib = logic 1 disables all automatic calibrations except after the reset sequence. Automatic calibration can also be triggered by the software when bit ClkQCalib has a logic 1 to logic 0 transition.

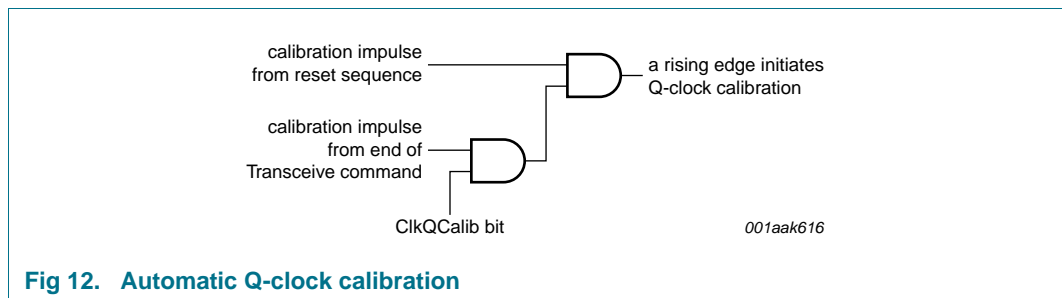


Fig 12. Automatic Q-clock calibration

**Remark:** The duration of the automatic Q-clock calibration is 65 oscillator periods or approximately 4.8  $\mu$ s.

The ClockQControl register's ClkQDelay[4:0] value is proportional to the phase-shift between the Q-clock and the I-clock. The ClkQ180Deg status flag bit is set when the phase-shift between the Q-clock and the I-clock is greater than 180°.

**Remark:**

- The StartUp configuration file enables automatic Q-clock calibration after a reset
- If bit ClkQCalib = logic 1, automatic calibration is not performed. Leaving this bit set to logic 1 can be used to permanently disable automatic calibration.
- It is possible to write data to the ClkQDelay[4:0] bits using the microprocessor. The aim could be to disable automatic calibration and set the delay using the software. Configuring the delay value using the software requires bit ClkQCalib to have been previously set to logic 1 and a time interval of at least 4.8  $\mu$ s has elapsed. Each delay value must be written with bit ClkQCalib set to logic 1. If bit ClkQCalib is logic 0, the configured delay value is overwritten by the next automatic calibration interval.



### 9.10.2.2 Amplifier

The demodulated signal must be amplified by the variable amplifier to achieve the best performance. The gain of the amplifiers can be adjusted using the RxControl1 register Gain[1:0] bits; see [Table 28](#).

**Table 28. Gain factors for the internal amplifier**

See [Table 84 "RxControl1 register bit descriptions" on page 59](#) for additional information.

| Register setting | Gain factor (dB)<br>(simulation results) |
|------------------|--|
| 00               | 20                                       |
| 01               | 24                                       |
| 10               | 31                                       |
| 11               | 35                                       |

### 9.10.2.3 Correlation circuitry

The correlation circuitry calculates the degree of matching between the received and an expected signal. The output is a measure of the amplitude of the expected signal in the received signal. This is done for both, the Q and I-channels. The correlator provides two outputs for each of the two input channels, resulting in a total of four output signals.

The correlation circuitry needs the phase information for the incoming card signal for optimum performance. This information is defined for the microprocessor using the BitPhase register. This value defines the phase relationship between the transmitter and receiver clock in multiples of the BitPhase time ( $t_{\text{BitPhase}} = 1 / 13.56 \text{ MHz}$ ).

### 9.10.2.4 Evaluation and digitizer circuitry

The correlation results are evaluated for each bit-half of the Manchester encoded signal. The evaluation and digitizer circuit decides from the signal strengths of both bit-halves, if the current bit is valid

- If the bit is valid, its value is identified
- If the bit is not valid, it is checked to identify if it contains a bit-collision

Select the following levels for optimal performance using RxThreshold register bits:

- **MinLevel[3:0]:** defines the minimum signal strength of the stronger bit-half's signal which is considered valid.
- **CollLevel[3:0]:** defines the minimum signal strength relative to the amplitude of the stronger half-bit that has to be exceeded by the weaker half-bit of the Manchester encoded signal to generate a bit-collision. If the signal's strength is below this value, logic 1 and logic 0 can be determined unequivocally.

After data transmission, the card is not allowed to send its response before a preset time period which is called the frame guard time in the ISO/IEC 14443 standard. The length of this time period is set using the RxWait register's RxWait[7:0] bits. The RxWait register defines when the receiver is switched on after data transmission to the card in multiples of one bit duration.

If bit RcvClkSell is set to logic 1, the I-clock is used to clock the correlator and evaluation circuits. If bit RcvClkSell is set to logic 0, the Q-clock is used.

**Remark:** It is recommended to use the Q-clock.

## 9.11 Serial signal switch

The MFRC530 comprises two main blocks:

- **digital circuitry:** comprising the state machines, encoder and decoder logic etc.
- **analog circuitry:** comprising the modulator, antenna drivers, receiver and amplification circuitry

The interface between these two blocks can be configured so that the interface signals are routed to pins MFIN and MFOUT. This makes it possible to connect the analog part of one MFRC530 to the digital part of another device.

**Remark:** The MFIN pin can only be accessed by 106 kBd. The Manchester with subcarrier and the Manchester encoded signal can only be accessed by the MFOUT pin at 106 kBd

### 9.11.1 Serial signal switch block diagram

[Figure 13](#) shows the serial signal switches. Three different switches are implemented in the serial signal switch enabling the MFRC530 to be used in different configurations.

The serial signal switch can also be used to check the transmitted and received data during the design-in phase or for test purposes. [Section 15.2.1 on page 99](#) describes the analog test signals and measurements at the serial signal switch.

**Remark:** The SLR400 uses pin name SIGOUT for pin MFOUT. The MFRC530 functionality includes the test modes for the SLRC400 using pin MFOUT.

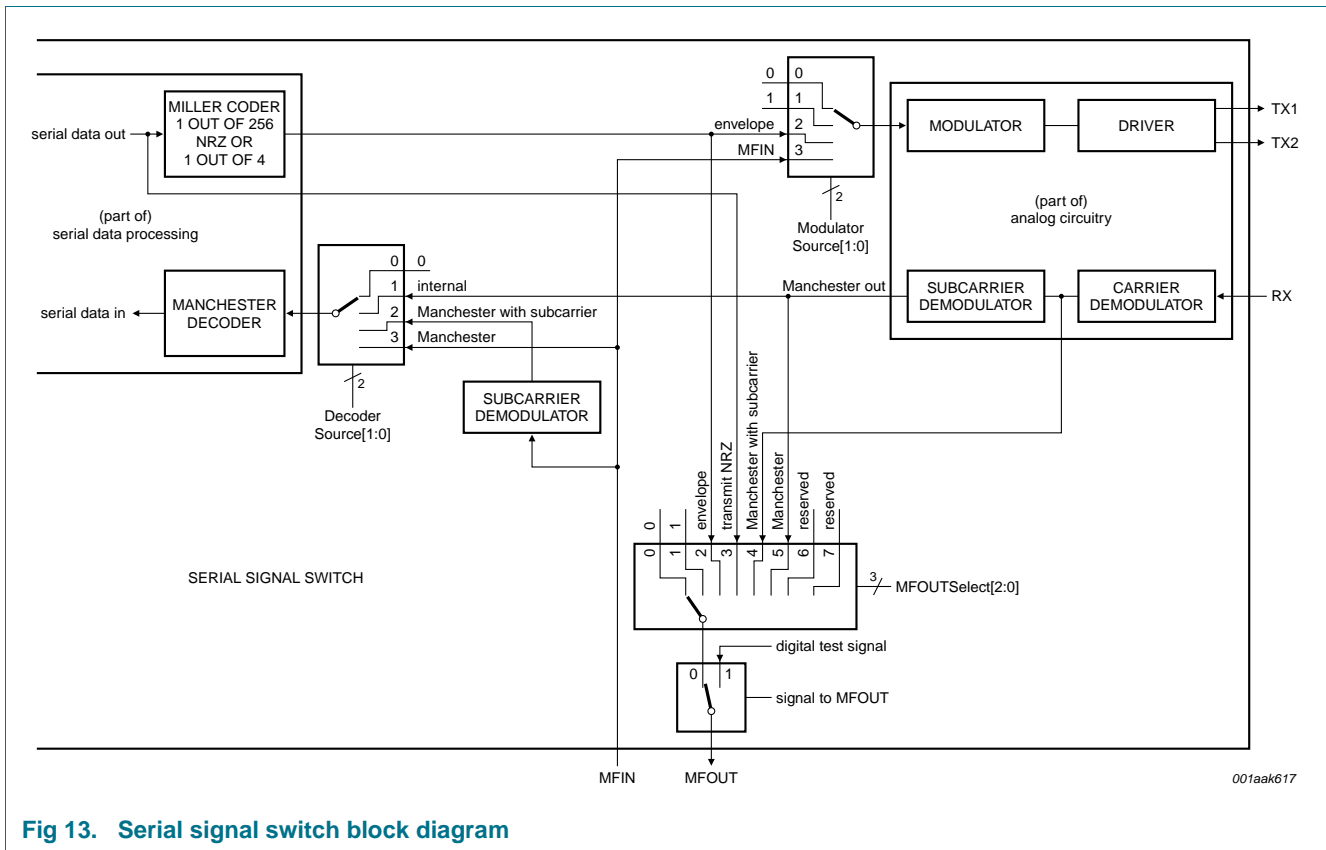


Fig 13. Serial signal switch block diagram

Section 9.11.2, Section 9.11.2.1 and Section 9.11.2.2 describe the relevant registers and settings used to configure and control the serial signal switch.

### 9.11.2 Serial signal switch registers

The RxControl2 register DecoderSource[1:0] bits define the input signal for the internal Manchester decoder and are described in Table 29.

Table 29. DecoderSource[1:0] values

See Table 94 on page 62 for additional information.

| Number | DecoderSource[1:0] | Input signal to decoder  |
|--------|--------------------|--|
| 0      | 00                 | constant 0   |
| 1      | 01                 | output of the analog part. This is the default configuration   |
| 2      | 10                 | direct connection to pin MFIN; expects an 847.5 kHz subcarrier signal modulated by a Manchester encoded signal |
| 3      | 11                 | direct connection to pin MFIN; expects a Manchester encoded signal   |

The TxControl register ModulatorSource[1:0] bits define the signal used to modulate the transmitted 13.56 MHz energy carrier. The modulated signal drives pins TX1 and TX2.

**Table 30. ModulatorSource[1:0] values**See [Table 94 on page 62](#) for additional information.

| Number | ModulatorSource[1:0] | Input signal to modulator  |
|--------|----------------------|--|
| 0      | 00                   | constant 0 (energy carrier off on pins TX1 and TX2)  |
| 1      | 01                   | constant 1 (continuous energy carrier on pins TX1 and TX2)                                 |
| 2      | 10                   | modulation signal (envelope) from the internal encoder. This is the default configuration. |
| 3      | 11                   | direct connection to MFIN; expects a Miller pulse coded signal                             |

The MFOUTSelect register MFOUTSelect[2:0] bits select the output signal which is to be routed to pin MFOUT.

**Table 31. MFOUTSelect[2:0] values**See [Table 108 on page 65](#) for additional information.

| Number | MFOUTSelect[2:0] | Signal routed to pin MFOUT   |
|--------|------------------|--|
| 0      | 000              | constant LOW   |
| 1      | 001              | constant HIGH  |
| 2      | 010              | modulation signal (envelope) from the internal encoder   |
| 3      | 011              | serial data stream to be transmitted; the same as for MFOUTSelect[2:0] = 010 but not encoded by the selected pulse encoder |
| 4      | 100              | output signal of the receiver circuit; card modulation signal regenerated and delayed                                      |
| 5      | 101              | output signal of the subcarrier demodulator; Manchester coded card signal  |
| 6      | 110              | reserved   |
| 7      | 111              | reserved   |

**Remark:** To use the MFOUTSelect[2:0] bits, the TestDigiSelect register SignalToMFOUT bit must be logic 0.

### 9.11.2.1 Active antenna concept

The MFRC530 analog and digital circuitry is accessed using pins MFIN and MFOUT.

[Table 32](#) lists the required settings.

**Table 32. Register settings to enable use of the analog circuitry**

| Register                          | Number <sup>[1]</sup> | Signal                             | MFRC530 pin |
|-----------------------------------|-----------------------|------------------------------------|-------------|
| <b>Analog circuitry settings</b>  |                       |                                    |             |
| ModulatorSource                   | 3                     | Miller pulse encoded               | MFIN        |
| MFOUTSelect                       | 4                     | Manchester encoded with subcarrier | MFOUT       |
| DecoderSource                     | X                     | -                                  | -           |
| <b>Digital circuitry settings</b> |                       |                                    |             |
| ModulatorSource                   | X                     | -                                  | -           |
| MFOUTSelect                       | 2                     | Miller pulse encoded               | MFOUT       |
| DecoderSource                     | 2                     | Manchester encoded with subcarrier | MFIN        |

[1] The number column refers to the value in the number column of [Table 29](#), [Table 30](#) and [Table 31](#).

Two MFRC530 devices configured as described in [Table 32](#) can be connected to each other using pins MFOUT and MFIN.

**Remark:** The active antenna concept can only be used at a baud rate of 106 kBd.

### 9.11.2.2 Driving both RF parts

It is possible to connect both passive and active antennas to a single IC. The passive antenna pins TX1, TX2 and RX are connected using the appropriate filter and matching circuit. At the same time an active antenna is connected to pins MFOUT and MFIN. In this configuration, two RF parts can be driven, one after another, by one microprocessor.

## 9.12 MIFARE higher baud rates

The MIFARE system is specified with a fixed baud rate of 106 kBd for communication on the RF interface. The current version of ISO/IEC 14443 A also defines 106 kBd for the initial phase of a communication between Proximity Integrated Circuit Cards (PICC) and Proximity Coupling Devices (PCD).

To cover requirements of large data transmissions and to speed up terminal to card communication, the MFRC530 supports communication at higher MIFARE baud rates in combination with a microcontroller IC such as the MIFARE ProX.

**Table 33. MIFARE higher baud rates**

| Communication direction   | Baud rates (kBd) |
|---|------------------|
| MFRC530 based PCD → microcontroller PICC supporting higher baud rates | 106, 212, 424    |
| Microcontroller PICC supporting higher baud rates → MFRC530 based PCD | 106, 212, 424    |

The MIFARE higher baud rates concept is described in the application note: *MIFARE Implementation of Higher Baud rates* [Ref. 5](#). This application covers the integration of the MIFARE higher baud rates communication concept in current applications.

## 9.13 MIFARE authentication and Crypto1

The security algorithm used in the MIFARE products is called Crypto1. It is based on a proprietary stream cipher with a 48-bit key length. To access data on MIFARE cards, knowledge of the key format is needed. The correct key must be available in the MFRC530 to enable successful card authentication and access to the card's data stored in the EEPROM.

After a card is selected as defined in ISO/IEC 14443 A standard, the user can continue with the MIFARE protocol. It is mandatory that the card authentication is performed.

Crypto1 authentication is a 3-pass authentication which is automatically performed when the Authent1 and Authent2 commands are executed (see [Section 11.6.3 on page 87](#) and [Section 11.6.4 on page 87](#)).

During the card authentication procedure, the security algorithm is initialized. After a successful authentication, communication with the MIFARE card is encrypted.

### 9.13.1 Crypto1 key handling

On execution of the authentication command, the MFRC530 reads the key from the key buffer. The key is always read from the key buffer and ensures Crypto1 authentication commands do not require addressing of a key. The user must ensure the correct key is prepared in the key buffer before triggering card authentication.

The key buffer can be loaded from:

- the EEPROM using the LoadKeyE2 command (see [Section 11.6.1 on page 86](#))
- the microprocessor's FIFO buffer using the LoadKey command (see [Section 11.6.2 on page 86](#)). This is shown in [Figure 14](#).

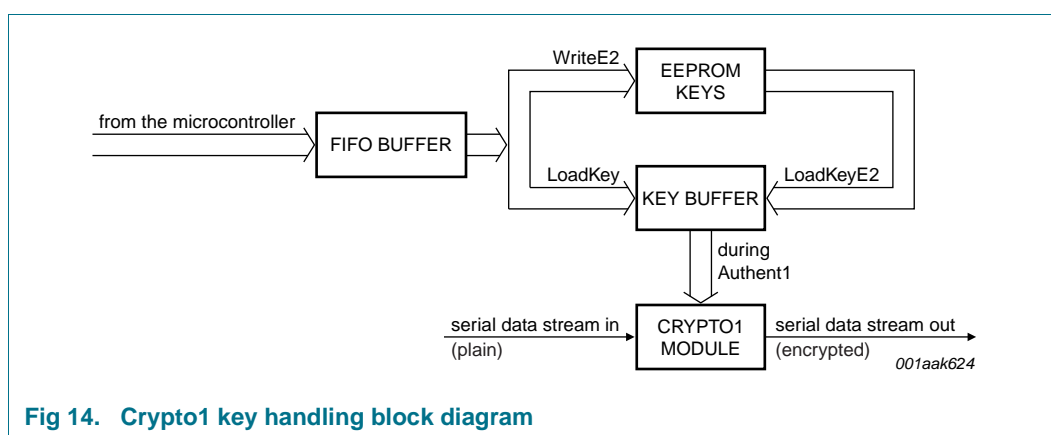


Fig 14. Crypto1 key handling block diagram

### 9.13.2 Authentication procedure

The Crypto1 security algorithm enables authentication of MIFARE cards. To obtain valid authentication, the correct key has to be available in the key buffer of the MFRC530. This can be ensured as follows:

1. Load the internal key buffer by using the LoadKeyE2 (see [Section 11.6.1 on page 86](#)) or the LoadKey (see [Section 11.6.2 on page 86](#)) commands.
2. Start the Authent1 command (see [Section 11.6.3 on page 87](#)). When finished, check the error flags to obtain the command execution status.
3. Start the Authent2 command (see [Section 11.6.4 on page 87](#)). When finished, check the error flags and bit Crypto1On to obtain the command execution status.

## 10. MFRC530 registers

### 10.1 Register addressing modes

Three methods can be used to operate the MFRC530:

- initiating functions and controlling data by executing commands
- configuring the functional operation using a set of configuration bits
- monitoring the state of the MFRC530 by reading status flags

The commands, configuration bits and flags are accessed using the microprocessor interface. The MFRC530 can internally address 64 registers using six address lines.

#### 10.1.1 Page registers

The MFRC530 register set is segmented into eight pages contain eight registers each. A Page register can always be addressed, irrespective of which page is currently selected.

#### 10.1.2 Dedicated address bus

When using the MFRC530 with the dedicated address bus, the microprocessor defines three address lines using address pins A0, A1 and A2. This enables addressing within a page. To switch between registers in different pages a paging mechanism needs to be used.

[Table 34](#) shows how the register address is assembled.

**Table 34. Dedicated address bus: assembling the register address**

| Register bit: UsePageSelect | Register address |             |             |    |    |    |
|-----------------------------|------------------|-------------|-------------|----|----|----|
| 1                           | PageSelect2      | PageSelect1 | PageSelect0 | A2 | A1 | A0 |

#### 10.1.3 Multiplexed address bus

The microprocessor may define all six address lines at once using the MFRC530 with a multiplexed address bus. In this case either the paging mechanism or linear addressing can be used.

[Table 35](#) shows how the register address is assembled.

**Table 35. Multiplexed address bus: assembling the register address**

| Multiplexed address bus type | UsePage Select | Register address |             |             |     |     |     |
|------------------------------|----------------|------------------|-------------|-------------|-----|-----|-----|
| Paging mode                  | 1              | PageSelect2      | PageSelect1 | PageSelect0 | AD2 | AD1 | AD0 |
| Linear addressing            | 0              | AD5              | AD4         | AD3         | AD2 | AD1 | AD0 |

## 10.2 Register bit behavior

Bits and flags for different registers behave differently, depending on their functions. In principle, bits with same behavior are grouped in common registers. [Table 36](#) describes the function of the Access column in the register tables.

**Table 36. Behavior and designation of register bits**

| Abbreviation | Behavior       | Description  |
|--------------|----------------|--|
| R/W          | read and write | <p>These bits can be read and written by the microprocessor. Since they are only used for control, their content is not influenced by internal state machines.</p> <p><b>Example:</b> TimerReload register may be read and written by the microprocessor. It will also be read by internal state machines but never changed by them.</p> |
| D            | dynamic        | <p>These bits can be read and written by the microprocessor. Nevertheless, they may also be written automatically by internal state machines.</p> <p><b>Example:</b> the Command register changes its value automatically after the execution of the command.</p>  |
| R            | read only      | <p>These registers hold flags which have a value determined by internal states only.</p> <p><b>Example:</b> the ErrorFlag register cannot be written externally but shows internal states.</p>   |
| W            | write only     | <p>These registers are used for control only. They may be written by the microprocessor but cannot be read. Reading these registers returns an undefined value.</p> <p><b>Example:</b> The TestAnaSelect register is used to determine the signal on pin AUX however, it is not possible to read its content.</p>                        |
| 0, 1 or x    | generic value  | <p>Where applicable, the values 0 and 1 indicate the <b>expected</b> logic value for a given bit. Where X is used, any logic value can be entered</p>  |



## 10.3 Register overview

**Table 37. MFRC530 register overview**

| Sub address (Hex)                            | Register name   | Function   | Refer to                            |
|--|-----------------|--|-------------------------------------|
| <b>Page 0: Command and status</b>            |                 |  |                                     |
| 00h  | Page            | selects the page register  | <a href="#">Table 39 on page 46</a> |
| 01h  | Command         | starts and stops command execution                                   | <a href="#">Table 41 on page 46</a> |
| 02h  | FIFOData        | input and output of 64-byte FIFO buffer                              | <a href="#">Table 43 on page 47</a> |
| 03h  | PrimaryStatus   | receiver and transmitter and FIFO buffer status flags                | <a href="#">Table 45 on page 47</a> |
| 04h  | FIFOLength      | number of bytes buffered in the FIFO buffer                          | <a href="#">Table 47 on page 48</a> |
| 05h  | SecondaryStatus | secondary status flags   | <a href="#">Table 49 on page 49</a> |
| 06h  | InterruptEn     | enable and disable interrupt request control bits                    | <a href="#">Table 51 on page 49</a> |
| 07h  | InterruptRq     | interrupt request flags  | <a href="#">Table 53 on page 50</a> |
| <b>Page 1: Control and status</b>            |                 |  |                                     |
| 08h  | Page            | selects the page register  | <a href="#">Table 39 on page 46</a> |
| 09h  | Control         | control flags for timer unit, power saving etc                       | <a href="#">Table 55 on page 51</a> |
| 0Ah  | ErrorFlag       | show the error status of the last command executed                   | <a href="#">Table 57 on page 52</a> |
| 0Bh  | CollPos         | bit position of the first bit-collision detected on the RF interface | <a href="#">Table 59 on page 53</a> |
| 0Ch  | TimerValue      | value of the timer   | <a href="#">Table 61 on page 53</a> |
| 0Dh  | CRCResultLSB    | LSB of the CRC coprocessor register                                  | <a href="#">Table 63 on page 53</a> |
| 0Eh  | CRCResultMSB    | MSB of the CRC coprocessor register                                  | <a href="#">Table 65 on page 54</a> |
| 0Fh  | BitFraming      | adjustments for bit oriented frames                                  | <a href="#">Table 67 on page 54</a> |
| <b>Page 2: Transmitter and coder control</b> |                 |  |                                     |
| 10h  | Page            | selects the page register  | <a href="#">Table 39 on page 46</a> |
| 11h  | TxControl       | controls the operation of the antenna driver pins TX1 and TX2        | <a href="#">Table 69 on page 55</a> |
| 12h  | CwConductance   | selects the conductance of the antenna driver pins TX1 and TX2       | <a href="#">Table 71 on page 56</a> |
| 13h  | PreSet13        | do not change these values   | <a href="#">Table 73 on page 56</a> |
| 14h  | CoderControl    | sets the clock frequency and the encoding                            | <a href="#">Table 75 on page 57</a> |
| 15h  | ModWidth        | selects the modulation pulse width                                   | <a href="#">Table 77 on page 57</a> |
| 16h  | PreSet16        | do not change these values   | <a href="#">Table 79 on page 58</a> |
| 17h  | PreSet17        | do not change these values   | <a href="#">Table 81 on page 58</a> |
| <b>Page 3: Receiver and decoder control</b>  |                 |  |                                     |
| 18   | Page            | selects the page register  | <a href="#">Table 39 on page 46</a> |
| 19   | RxControl1      | controls receiver behavior   | <a href="#">Table 83 on page 59</a> |
| 1A   | DecoderControl  | controls decoder behavior  | <a href="#">Table 85 on page 60</a> |
| 1B   | BitPhase        | selects the bit-phase between transmitter and receiver clock         | <a href="#">Table 87 on page 60</a> |
| 1C   | RxThreshold     | selects thresholds for the bit decoder                               | <a href="#">Table 89 on page 61</a> |
| 1D   | BPSKDemControl  | controls BPSK receiver behavior                                      | <a href="#">Table 91 on page 61</a> |
| 1Eh  | RxControl2      | controls decoder and defines the receiver input source               | <a href="#">Table 93 on page 62</a> |
| 1Fh  | ClockQControl   | clock control for the 90° phase-shifted Q-channel clock              | <a href="#">Table 95 on page 62</a> |

Table 37. MFRC530 register overview ...continued

| Sub address (Hex)                                    | Register name     | Function   | Refer to                             |
|--|-------------------|--|--------------------------------------|
| <b>Page 4: RF Timing and channel redundancy</b>      |                   |  |                                      |
| 20h  | Page              | selects the page register  | <a href="#">Table 39 on page 46</a>  |
| 21h  | RxWait            | selects the interval after transmission before the receiver starts   | <a href="#">Table 97 on page 63</a>  |
| 22h  | ChannelRedundancy | selects the method and mode used to check data integrity on the RF channel   | <a href="#">Table 99 on page 63</a>  |
| 23h  | CRCPresetLSB      | preset LSB value for the CRC register  | <a href="#">Table 101 on page 64</a> |
| 24h  | CRCPresetMSB      | preset MSB value for the CRC register  | <a href="#">Table 103 on page 64</a> |
| 25h  | PreSet25          | do not change these values   | <a href="#">Table 105 on page 64</a> |
| 26h  | MFOUSelect        | selects internal signal applied to pin MFOUT, includes the MSB of value TimeSlotPeriod; see <a href="#">Table 107 on page 65</a> | <a href="#">Table 107 on page 65</a> |
| 27h  | PreSet27          | do not change these values   | <a href="#">Table 109 on page 65</a> |
| <b>Page 5: FIFO, timer and IRQ pin configuration</b> |                   |  |                                      |
| 28h  | Page              | selects the page register  | <a href="#">Table 39 on page 46</a>  |
| 29h  | FIFOLevel         | defines the FIFO buffer overflow and underflow warning levels  | <a href="#">Table 47 on page 48</a>  |
| 2Ah  | TimerClock        | selects the timer clock divider  | <a href="#">Table 113 on page 66</a> |
| 2Bh  | TimerControl      | selects the timer start and stop conditions  | <a href="#">Table 115 on page 67</a> |
| 2Ch  | TimerReload       | defines the timer preset value   | <a href="#">Table 117 on page 67</a> |
| 2Dh  | IRQPinConfig      | configures pin IRQ output stage  | <a href="#">Table 119 on page 68</a> |
| 2Eh  | PreSet2E          | do not change these values   | <a href="#">Table 121 on page 68</a> |
| 2Fh  | PreSet2F          | do not change these values   | <a href="#">Table 122 on page 68</a> |
| <b>Page 6: reserved registers</b>                    |                   |  |                                      |
| 30h  | Page              | selects the page register  | <a href="#">Table 39 on page 46</a>  |
| 31h  | reserved          | reserved   | <a href="#">Table 123 on page 68</a> |
| 32h  | reserved          | reserved   |                                      |
| 33h  | reserved          | reserved   |                                      |
| 34h  | reserved          | reserved   |                                      |
| 35h  | reserved          | reserved   |                                      |
| 36h  | reserved          | reserved   |                                      |
| 37h  | reserved          | reserved   |                                      |
| <b>Page 7: Test control</b>                          |                   |  |                                      |
| 38h  | Page              | selects the page register  | <a href="#">Table 39 on page 46</a>  |
| 39h  | reserved          | reserved   | <a href="#">Table 124 on page 69</a> |
| 3Ah  | TestAnaSelect     | selects analog test mode   | <a href="#">Table 125 on page 69</a> |
| 3Bh  | reserved          | reserved   | <a href="#">Table 127 on page 70</a> |
| 3Ch  | reserved          | reserved   | <a href="#">Table 128 on page 70</a> |
| 3Dh  | TestDigiSelect    | selects digital test mode  | <a href="#">Table 129 on page 70</a> |
| 3Eh  | reserved          | reserved   | <a href="#">Table 131 on page 71</a> |
| 3Fh  | reserved          | reserved   |                                      |

## 10.4 MFRC530 register flags overview

Table 38. MFRC530 register flags overview

| Flag(s)            | Register          | Bit    | Address |
|--------------------|-------------------|--------|---------|
| AccessErr          | ErrorFlag         | 5      | 0Ah     |
| BitPhase[7:0]      | BitPhase          | 7 to 0 | 1Bh     |
| ClkQ180Deg         | ClockQControl     | 7      | 1Fh     |
| ClkQCalib          | ClockQControl     | 6      | 1Fh     |
| ClkQDelay[4:0]     | ClockQControl     | 4 to 0 | 1Fh     |
| CoderRate[2:0]     | CoderControl      | 5 to 3 | 14h     |
| CollErr            | ErrorFlag         | 0      | 0Ah     |
| CollLevel[3:0]     | RxThreshold       | 3 to 0 | 1Ch     |
| CollPos[7:0]       | CollPos           | 7 to 0 | 0Bh     |
| Command[5:0]       | Command           | 5 to 0 | 01h     |
| CRC3309            | ChannelRedundancy | 5      | 22h     |
| CRC8               | ChannelRedundancy | 4      | 22h     |
| CRCErr             | ErrorFlag         | 3      | 0Ah     |
| CRCPresetLSB[7:0]  | CRCPresetLSB      | 7 to 0 | 23h     |
| CRCPresetMSB[7:0]  | CRCPresetMSB      | 7 to 0 | 24h     |
| CRCReady           | SecondaryStatus   | 5      | 05h     |
| CRCResultMSB[7:0]  | CRCResultMSB      | 7 to 0 | 0Eh     |
| CRCResultLSB[7:0]  | CRCResultLSB      | 7 to 0 | 0Dh     |
| Crypto1On          | Control           | 3      | 09h     |
| DecoderSource[1:0] | RxControl2        | 1 to 0 | 1Eh     |
| E2Ready            | SecondaryStatus   | 6      | 05h     |
| Err                | PrimaryStatus     | 2      | 03h     |
| FIFOData[7:0]      | FIFOData          | 7 to 0 | 02h     |
| FIFOLength[6:0]    | FIFOLength        | 6 to 0 | 04h     |
| FIFOovfl           | ErrorFlag         | 4      | 0Ah     |
| FilterAmpDet       | BPSKDemControl    | 4      | 1Dh     |
| FlushFIFO          | Control           | 0      | 09h     |
| FramingErr         | ErrorFlag         | 2      | 0Ah     |
| Gain[1:0]          | RxControl1        | 1 to 0 | 19h     |
| GsCfgCW[5:0]       | CwConductance     | 5 to 0 | 12h     |
| HiAlert            | PrimaryStatus     | 1      | 03h     |
| HiAlertIEn         | InterruptEn       | 1      | 06h     |
| HiAlertIRq         | InterruptRq       | 1      | 07h     |
| IdleIEn            | InterruptEn       | 2      | 06h     |
| IdleIRq            | InterruptRq       | 2      | 07h     |
| IFDetectBusy       | Command           | 7      | 01h     |
| IRq                | PrimaryStatus     | 3      | 03h     |
| IRQInv             | IRQPinConfig      | 1      | 2Dh     |
| IRQPushPull        | IRQPinConfig      | 0      | 2Dh     |

Table 38. MFRC530 register flags overview ...continued

| Flag(s)                | Register          | Bit    | Address                                   |
|------------------------|-------------------|--------|---|
| KeyErr                 | ErrorFlag         | 6      | 0Ah                                       |
| LoAlert                | PrimaryStatus     | 0      | 03h                                       |
| LoAlertIE              | InterruptEn       | 0      | 06h                                       |
| LoAlertIRq             | InterruptRq       | 0      | 07h                                       |
| LPOff                  | RxControl1        | 2      | 19h                                       |
| MFOUSelect[2:0]        | MFOUSelect        | 2 to 0 | 26h                                       |
| MinLevel[3:0]          | RxThreshold       | 7 to 4 | 1Ch                                       |
| ModemState[2:0]        | PrimaryStatus     | 6 to 4 | 03h                                       |
| ModulatorSource[1:0]   | TxControl         | 6 to 5 | 11h                                       |
| ModWidth[7:0]          | ModWidth          | 7 to 0 | 15h                                       |
| PageSelect[2:0]        | Page              | 2 to 0 | 00h, 08h, 10h, 18h, 20h, 28h, 30h and 38h |
| ParityEn               | ChannelRedundancy | 0      | 22h                                       |
| ParityErr              | ErrorFlag         | 1      | 0Ah                                       |
| ParityOdd              | ChannelRedundancy | 1      | 22h                                       |
| PowerDown              | Control           | 4      | 09h                                       |
| RcvClkSell             | RxControl2        | 7      | 1Eh                                       |
| RxAlign[2:0]           | BitFraming        | 6 to 4 | 0Fh                                       |
| RxAutoPD               | RxControl2        | 6      | 1Eh                                       |
| RxCRCEn                | ChannelRedundancy | 3      | 22h                                       |
| RxCoding               | DecoderControl    | 0      | 1Ah                                       |
| RxIE                   | InterruptEn       | 3      | 06h                                       |
| RxIRq                  | InterruptRq       | 3      | 07h                                       |
| RxLastBits[2:0]        | SecondaryStatus   | 2 to 0 | 05h                                       |
| RxMultiple             | DecoderControl    | 6      | 1Ah                                       |
| RxWait[7:0]            | RxWait            | 7 to 0 | 21h                                       |
| SetIE                  | InterruptEn       | 7      | 06h                                       |
| SetIRq                 | InterruptRq       | 7      | 07h                                       |
| SignalToMFOU           | TestDigiSelect    | 7      | 3Dh                                       |
| StandBy                | Control           | 5      | 09h                                       |
| SubCPulses[2:0]        | RxControl1        | 7 to 5 | 19h                                       |
| TauB[1:0]              | BPSKDemControl    | 1 to 0 | 1Dh                                       |
| TauD[1:0]              | BPSKDemControl    | 3 to 2 | 1Dh                                       |
| TAutoRestart           | TimerClock        | 5      | 2Ah                                       |
| TestAnaOutSel[4:0]     | TestAnaSelect     | 3 to 0 | 3Ah                                       |
| TestDigiSignalSel[6:0] | TestDigiSelect    | 6 to 0 | 3Dh                                       |
| TimerIE                | InterruptEn       | 5      | 06h                                       |
| TimerIRq               | InterruptRq       | 5      | 07h                                       |
| TimerValue[7:0]        | TimerValue        | 7 to 0 | 0Ch                                       |
| TPreScaler[4:0]        | TimerClock        | 4 to 0 | 2Ah                                       |
| TReloadValue[7:0]      | TimerReload       | 7 to 0 | 2Ch                                       |

Table 38. MFRC530 register flags overview ...continued

| Flag(s)         | Register          | Bit    | Address                                   |
|-----------------|-------------------|--------|---|
| TRunning        | SecondaryStatus   | 7      | 05h                                       |
| TStartTxBegin   | TimerControl      | 0      | 2Bh                                       |
| TStartTxEnd     | TimerControl      | 1      | 2Bh                                       |
| TStartNow       | Control           | 1      | 09h                                       |
| TStopRxBegin    | TimerControl      | 2      | 2Bh                                       |
| TStopRxEnd      | TimerControl      | 3      | 2Bh                                       |
| TStopNow        | Control           | 2      | 09h                                       |
| TX1RFEn         | TxControl         | 0      | 11h                                       |
| TX2Cw           | TxControl         | 3      | 11h                                       |
| TX2Inv          | TxControl         | 3      | 11h                                       |
| TX2RFEn         | TxControl         | 1      | 11h                                       |
| TxCRCEn         | ChannelRedundancy | 2      | 22h                                       |
| TxIEn           | InterruptEn       | 4      | 06h                                       |
| TxIRq           | InterruptRq       | 4      | 07h                                       |
| TxLastBits[2:0] | BitFraming        | 2 to 0 | 0Fh                                       |
| UsePageSelect   | Page              | 7      | 00h, 08h, 10h, 18h, 20h, 28h, 30h and 38h |
| WaterLevel[5:0] | FIFOLevel         | 5 to 0 | 29h                                       |
| ZeroAfterColl   | DecoderControl    | 5      | 1Ah                                       |

## 10.5 Register descriptions

### 10.5.1 Page 0: Command and status

#### 10.5.1.1 Page register

Selects the page register.

**Table 39. Page register (address: 00h, 08h, 10h, 18h, 20h, 28h, 30h, 38h)  
reset value: 1000 0000b, 80h bit allocation**

| Bit    | 7             | 6    | 5 | 4 | 3               | 2   | 1   | 0 |
|--------|---------------|------|---|---|-----------------|-----|-----|---|
| Symbol | UsePageSelect | 0000 |   |   | PageSelect[2:0] |     |     |   |
| Access | R/W           | R/W  |   |   | R/W             | R/W | R/W |   |

**Table 40. Page register bit descriptions**

| Bit    | Symbol          | Value | Description   |
|--------|-----------------|-------|---|
| 7      | UsePageSelect   | 1     | the value of PageSelect[2:0] is used as the register address A5, A4, and A3. The LSBs of the register address are defined using the address pins or the internal address latch, respectively. |
|        |                 | 0     | the complete content of the internal address latch defines the register address. The address pins are used as described in <a href="#">Table 5 on page 8</a> .                                |
| 6 to 3 | 0000            | -     | reserved  |
| 2 to 0 | PageSelect[2:0] | -     | when UsePageSelect = logic 1, the value of PageSelect is used to specify the register page (A5, A4 and A3 of the register address)  |

#### 10.5.1.2 Command register

Starts and stops the command execution.

**Table 41. Command register (address: 01h) reset value: x000 0000b, x0h bit allocation**

| Bit    | 7            | 6 | 5            | 4 | 3 | 2 | 1 | 0 |
|--------|--------------|---|--------------|---|---|---|---|---|
| Symbol | IFDetectBusy | 0 | Command[5:0] |   |   |   |   |   |
| Access | R            | R | D            |   |   |   |   |   |

**Table 42. Command register bit descriptions**

| Bit    | Symbol       | Value | Description   |
|--------|--------------|-------|---|
| 7      | IFDetectBusy |       | shows the status of interface detection logic   |
|        |              | 0     | interface detection finished successfully   |
|        |              | 1     | interface detection ongoing   |
| 6      | 0            | -     | reserved  |
| 5 to 0 | Command[5:0] | -     | activates a command based on the Command code. Reading this register shows which command is being executed. |

### 10.5.1.3 FIFOData register

Input and output of the 64 byte FIFO buffer.

**Table 43. FIFOData register (address: 02h) reset value: xxxx xxxxb, xxh bit allocation**

| Bit    | 7             | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------|---------------|---|---|---|---|---|---|---|
| Symbol | FIFOData[7:0] |   |   |   |   |   |   |   |
| Access | D             |   |   |   |   |   |   |   |

**Table 44. FIFOData register bit descriptions**

| Bit    | Symbol        | Description  |
|--------|---------------|--|
| 7 to 0 | FIFOData[7:0] | data input and output port for the internal 64-byte FIFO buffer. The FIFO buffer acts as a parallel in to parallel out converter for all data streams. |

### 10.5.1.4 PrimaryStatus register

Bits relating to receiver, transmitter and FIFO buffer status flags.

**Table 45. PrimaryStatus register (address: 03h) reset value: 0000 0101b, 05h bit allocation**

| Bit    | 7 | 6               | 5 | 4   | 3   | 2       | 1       | 0 |
|--------|---|-----------------|---|-----|-----|---------|---------|---|
| Symbol | 0 | ModemState[2:0] |   | IRq | Err | HiAlert | LoAlert |   |
| Access | R | R               |   | R   | R   | R       | R       |   |

**Table 46. PrimaryStatus register bit descriptions**

| Bit    | Symbol          | Value | Status     | Description   |
|--------|-----------------|-------|------------|---|
| 7      | 0               | -     |            | reserved  |
| 6 to 4 | ModemState[2:0] |       |            | shows the state of the transmitter and receiver state machines:                                   |
|        |                 | 000   | Idle       | neither the transmitter or receiver are operating; neither of them are started or have input data |
|        |                 | 001   | TxSOF      | transmit start of frame pattern   |
|        |                 | 010   | TxData     | transmit data from the FIFO buffer (or redundancy CRC check bits)                                 |
|        |                 | 011   | TxEOF      | transmit End Of Frame (EOF) pattern   |
|        |                 | 100   | GoToRx1    | intermediate state 1; receiver starts   |
|        |                 |       | GoToRx2    | intermediate state 2; receiver finishes   |
|        |                 | 101   | PrepareRx  | waiting until the RxWait register time period expires   |
|        |                 | 110   | AwaitingRx | receiver activated; waiting for an input signal on pin RX   |
|        |                 | 111   | Receiving  | receiving data  |
| 3      | IRq             | -     |            | shows any interrupt source requesting attention based on the InterruptEn register flag settings   |

**Table 46. PrimaryStatus register bit descriptions ...continued**

| Bit | Symbol  | Value | Status | Description  |
|-----|---------|-------|--------|--|
| 2   | Err     | 1     |        | any error flag in the ErrorFlag register is set  |
| 1   | HiAlert | 1     |        | the alert level for the number of bytes in the FIFO buffer (FIFOLength[6:0]) is:<br>$HiAlert = (64 - FIFOLength) \leq WaterLevel$<br>otherwise value = logic 0<br>Example:<br>FIFOLength = 60, WaterLevel = 4 then<br>HiAlert = logic 1<br>FIFOLength = 59, WaterLevel = 4 then<br>HiAlert = logic 0 |
| 0   | LoAlert | 1     |        | the alert level for number of bytes in the FIFO buffer (FIFOLength[6:0]) is:<br>$LoAlert = FIFOLength \leq WaterLevel$ otherwise<br>value = logic 0<br>Example:<br>FIFOLength = 4, WaterLevel = 4 then<br>LoAlert = logic 1<br>FIFOLength = 5, WaterLevel = 4 then<br>LoAlert = logic 0              |

**10.5.1.5 FIFOLength register**

Number of bytes in the FIFO buffer.

**Table 47. FIFOLength register (address: 04h) reset value: 0000 0000b, 00h bit allocation**

| Bit    | 7 | 6               | 5 | 4 | 3 | 2 | 1 | 0 |
|--------|---|-----------------|---|---|---|---|---|---|
| Symbol | 0 | FIFOLength[6:0] |   |   |   |   |   |   |
| Access | R | R               |   |   |   |   |   |   |

**Table 48. FIFOLength bit descriptions**

| Bit    | Symbol          | Description  |
|--------|-----------------|--|
| 7      | 0               | reserved   |
| 6 to 0 | FIFOLength[6:0] | gives the number of bytes stored in the FIFO buffer. Writing increments the FIFOLength register value while reading decrements the FIFOLength register value |



**10.5.1.6 SecondaryStatus register**

Various secondary status flags.

**Table 49. SecondaryStatus register (address: 05h) reset value: 01100 000b, 60h bit allocation**

| Bit    | 7        | 6       | 5        | 4  | 3 | 2               | 1 | 0 |
|--------|----------|---------|----------|----|---|-----------------|---|---|
| Symbol | TRunning | E2Ready | CRCReady | 00 |   | RxLastBits[2:0] |   |   |
| Access | R        | R       | R        | R  |   | R               |   |   |

**Table 50. SecondaryStatus register bit descriptions**

| Bit    | Symbol          | Value | Description  |
|--------|-----------------|-------|--|
| 7      | TRunning        | 1     | the timer unit is running and the counter decrements the TimerValue register on the next timer clock cycle |
|        |                 | 0     | the timer unit is not running  |
| 6      | E2Ready         | 1     | EEPROM programming is finished   |
|        |                 | 0     | EEPROM programming is ongoing  |
| 5      | CRCReady        | 1     | CRC calculation is finished  |
|        |                 | 0     | CRC calculation is ongoing   |
| 4 to 3 | 00              | -     | reserved   |
| 2 to 0 | RxLastBits[2:0] | -     | shows the number of valid bits in the last received byte. If zero, the whole byte is valid                 |

**10.5.1.7 InterruptEn register**

Control bits to enable and disable passing of interrupt requests.

**Table 51. InterruptEn register (address: 06h) reset value: 0000 0000b, 00h bit allocation**

| Bit    | 7      | 6   | 5        | 4     | 3     | 2       | 1          | 0          |
|--------|--------|-----|----------|-------|-------|---------|------------|------------|
| Symbol | SetIEn | 0   | TimerIEn | TxIEn | RxIEn | IdleIEn | HiAlertIEn | LoAlertIEn |
| Access | W      | R/W | R/W      | R/W   | R/W   | R/W     | R/W        | R/W        |

**Table 52. InterruptEn register bit descriptions**

| Bit | Symbol     | Value | Description   |
|-----|------------|-------|---|
| 7   | SetIEn     | 1     | indicates that the marked bits in the InterruptEn register are set          |
|     |            | 0     | clears the marked bits  |
| 6   | 0          | -     | reserved  |
| 5   | TimerIEn   | -     | sends the TimerIRq timer interrupt request to pin IRQ <sup>[1]</sup>        |
| 4   | TxIEn      | -     | sends the TxIRq transmitter interrupt request to pin IRQ <sup>[1]</sup>     |
| 3   | RxIEn      | -     | sends the RxIRq receiver interrupt request to pin IRQ <sup>[1]</sup>        |
| 2   | IdleIEn    | -     | sends the IdleIRq idle interrupt request to pin IRQ <sup>[1]</sup>          |
| 1   | HiAlertIEn | -     | sends the HiAlertIRq high alert interrupt request to pin IRQ <sup>[1]</sup> |
| 0   | LoAlertIEn | -     | sends the LoAlertIRq low alert interrupt request to pin IRQ <sup>[1]</sup>  |

[1] This bit can only be set or cleared using bit SetIEn.

### 10.5.1.8 InterruptRq register

Interrupt request flags.

**Table 53. InterruptRq register (address: 07h) reset value: 0000 0000b, 00h bit allocation**

| Bit    | 7      | 6   | 5        | 4     | 3     | 2       | 1          | 0          |
|--------|--------|-----|----------|-------|-------|---------|------------|------------|
| Symbol | SetIRq | 0   | TimerIRq | TxIRq | RxIRq | IdleIRq | HiAlertIRq | LoAlertIRq |
| Access | W      | R/W | D        | D     | D     | D       | D          | D          |

**Table 54. InterruptRq register bit descriptions**

| Bit | Symbol     | Value | Description   |
|-----|------------|-------|---|
| 7   | SetIRq     | 1     | sets the marked bits in the InterruptRq register  |
|     |            | 0     | clears the marked bits in the InterruptRq register  |
| 6   | 0          | -     | reserved  |
| 5   | TimerIRq   | 1     | timer decrements the TimerValue register to zero  |
|     |            | 0     | timer decrements are still greater than zero  |
| 4   | TxIRq      | 1     | TxIRq is set to logic 1 if one of the following events occurs:<br>Transceive command; all data transmitted<br>Authent1 and Authent2 commands; all data transmitted<br>WriteE2 command; all data is programmed<br>CalcCRC command; all data is processed         |
|     |            | 0     | when not acted on by Transceive, Authent1, Authent2, WriteE2 or CalcCRC commands  |
| 3   | RxIRq      | 1     | the receiver terminates   |
|     |            | 0     | reception still ongoing   |
| 2   | IdleIRq    | 1     | command terminates correctly. For example; when the Command register changes its value from any command to the Idle command. If an unknown command is started the IdleIRq bit is set. Microprocessor start-up of the Idle command does not set the IdleIRq bit. |
|     |            | 0     | IdleIRq = logic 0 in all other instances  |
| 1   | HiAlertIRq | 1     | PrimaryStatus register HiAlert bit is set <sup>[1]</sup>  |
|     |            | 0     | PrimaryStatus register HiAlert bit is not set   |
| 0   | LoAlertIRq | 1     | PrimaryStatus register LoAlert bit is set <sup>[1]</sup>  |
|     |            | 0     | PrimaryStatus register LoAlert bit is not set   |

[1] PrimaryStatus register bit HiAlertIRq stores this event and it can only be reset using bit SetIRq.

## 10.5.2 Page 1: Control and status

### 10.5.2.1 Page register

Selects the page register; see [Section 10.5.1.1 “Page register” on page 46](#).

### 10.5.2.2 Control register

Various control flags, for timer, power saving, etc.

**Table 55. Control register (address: 09h) reset value: 0000 0000b, 00h bit allocation**

| Bit    | 7   | 6 | 5       | 4         | 3         | 2        | 1         | 0         |
|--------|-----|---|---------|-----------|-----------|----------|-----------|-----------|
| Symbol | 00  |   | StandBy | PowerDown | Crypto1On | TStopNow | TStartNow | FlushFIFO |
| Access | R/W |   | D       | D         | D         | W        | W         | W         |

**Table 56. Control register bit descriptions**

| Bit    | Symbol    | Value | Description   |
|--------|-----------|-------|---|
| 7 to 6 | 00        | -     | reserved  |
| 5      | StandBy   | 1     | activates Standby mode. The current consuming blocks are switched off but the clock keeps running   |
| 4      | PowerDown | 1     | activates Power-down mode. The current consuming blocks are switched off including the clock  |
| 3      | Crypto1On | 1     | Crypto1 unit is switched on and all data communication with the card is encrypted <sup>[1]</sup>  |
|        |           | 0     | Crypto1 unit is switched off. All data communication with the card is unencrypted (plain)   |
| 2      | TStopNow  | 1     | immediately stops the timer <sup>[2]</sup>  |
| 1      | TStartNow | 1     | immediately starts the timer <sup>[2]</sup>   |
| 0      | FlushFIFO | 1     | immediately clears the internal FIFO buffer's read and write pointer, the FIFOLength[6:0] bits are set to logic 0 and the FIFOvfl flag <sup>[2]</sup> |

[1] This bit can only be set to logic 1 by successful execution of the Authent2 command

[2] Reading this bit always returns logic 0

### 10.5.2.3 ErrorFlag register

Error flags show the error status of the last executed command.

**Table 57. ErrorFlag register (address: 0Ah) reset value: 0100 0000b, 40h bit allocation**

| Bit    | 7 | 6      | 5         | 4       | 3      | 2          | 1         | 0       |
|--------|---|--------|-----------|---------|--------|------------|-----------|---------|
| Symbol | 0 | KeyErr | AccessErr | FIFOvfl | CRCErr | FramingErr | ParityErr | CollErr |
| Access | R | R      | R         | R       | R      | R          | R         | R       |

**Table 58. ErrorFlag register bit descriptions**

| Bit | Symbol     | Value | Description  |
|-----|------------|-------|--|
| 7   | 0          | -     | reserved   |
| 6   | KeyErr     | 1     | set when the LoadKeyE2 or LoadKey command recognize that the input data is not encoded based on the key format definition            |
|     |            | 0     | set when the LoadKeyE2 or the LoadKey command starts   |
| 5   | AccessErr  | 1     | set when the access rights to the EEPROM are violated  |
|     |            | 0     | set when an EEPROM related command starts  |
| 4   | FIFOvfl    | 1     | set when the microprocessor or MFRC530 internal state machine (e.g. receiver) tries to write data to the FIFO buffer when it is full |
| 3   | CRCErr     | 1     | set when RxCRCEn is set and the CRC fails  |
|     |            | 0     | automatically set during the PrepareRx state in the receiver start phase   |
| 2   | FramingErr | 1     | set when the SOF is incorrect  |
|     |            | 0     | automatically set during the PrepareRx state in the receiver start phase   |
| 1   | ParityErr  | 1     | set when the parity check fails  |
|     |            | 0     | automatically set during the PrepareRx state in the receiver start phase   |
| 0   | CollErr    | 1     | set when a bit-collision is detected   |
|     |            | 0     | automatically set during the PrepareRx state in the receiver start phase   |

**10.5.2.4 CollPos register**

Bit position of the first bit-collision detected on the RF interface.

**Table 59. CollPos register (address: 0Bh) reset value: 0000 0000b, 00h bit allocation**

| Bit    | 7            | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------|--------------|---|---|---|---|---|---|---|
| Symbol | CollPos[7:0] |   |   |   |   |   |   |   |
| Access | R            |   |   |   |   |   |   |   |

**Table 60. CollPos register bit descriptions**

| Bit    | Symbol       | Description  |
|--------|--------------|--|
| 7 to 0 | CollPos[7:0] | this register shows the bit position of the first detected collision in a received frame.<br>Example:<br>00h indicates a bit collision in the start bit<br>01h indicates a bit collision in the 1 <sup>st</sup> bit<br>...<br>08h indicates a bit collision in the 8 <sup>th</sup> bit |

**10.5.2.5 TimerValue register**

Value of the timer.

**Table 61. TimerValue register (address: 0Ch) reset value: xxxx xxxxb, xxh bit allocation**

| Bit    | 7               | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------|-----------------|---|---|---|---|---|---|---|
| Symbol | TimerValue[7:0] |   |   |   |   |   |   |   |
| Access | R               |   |   |   |   |   |   |   |

**Table 62. TimerValue register bit descriptions**

| Bit    | Symbol          | Description                                 |
|--------|-----------------|---|
| 7 to 0 | TimerValue[7:0] | this register shows the timer counter value |

**10.5.2.6 CRCResultLSB register**

LSB of the CRC coprocessor register.

**Table 63. CRCResultLSB register (address: 0Dh) reset value: xxxx xxxxb, xxh bit allocation**

| Bit    | 7                 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------|-------------------|---|---|---|---|---|---|---|
| Symbol | CRCResultLSB[7:0] |   |   |   |   |   |   |   |
| Access | R                 |   |   |   |   |   |   |   |

**Table 64. CRCResultLSB register bit descriptions**

| Bit    | Symbol            | Description   |
|--------|-------------------|---|
| 7 to 0 | CRCResultLSB[7:0] | gives the CRC register's least significant byte value; only valid if CRCReady = logic 1 |

### 10.5.2.7 CRCResultMSB register

MSB of the CRC coprocessor register.

**Table 65. CRCResultMSB register (address: 0Eh) reset value: xxxx xxxxb, xxh bit allocation**

| Bit    | 7                 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------|-------------------|---|---|---|---|---|---|---|
| Symbol | CRCResultMSB[7:0] |   |   |   |   |   |   |   |
| Access | R                 |   |   |   |   |   |   |   |

**Table 66. CRCResultMSB register bit descriptions**

| Bit    | Symbol            | Description   |
|--------|-------------------|---|
| 7 to 0 | CRCResultMSB[7:0] | gives the CRC register's most significant byte value; only valid if CRCReady = logic 1.<br>The register's value is undefined for 8-bit CRC calculation. |

### 10.5.2.8 BitFraming register

Adjustments for bit oriented frames.

**Table 67. BitFraming register (address: 0Fh) reset value: 0000 0000b, 00h bit allocation**

| Bit    | 7   | 6            | 5 | 4 | 3   | 2               | 1 | 0 |
|--------|-----|--------------|---|---|-----|-----------------|---|---|
| Symbol | 0   | RxAlign[2:0] |   |   | 0   | TxLastBits[2:0] |   |   |
| Access | R/W | D            |   |   | R/W | D               |   |   |

**Table 68. BitFraming register bit descriptions**

| Bit    | Symbol          | Value | Description   |
|--------|-----------------|-------|---|
| 7      | 0               | -     | reserved  |
| 6 to 4 | RxAlign[2:0]    |       | defines the bit position in the FIFO buffer for the first bit received and stored. Additional received bits are stored in the next subsequent bit positions. After reception, RxAlign[2:0] is automatically cleared. For example: |
|        |                 | 000   | the LSB of the received bit is stored in bit position 0 and the second received bit is stored in bit position 1   |
|        |                 | 001   | the LSB of the received bit is stored in bit position 1, the second received bit is stored in bit position 2  |
|        |                 | ...   |   |
|        |                 | 111   | the LSB of the received bit is stored in bit position 7, the second received bit is stored in the next byte in bit position 0   |
| 3      | 0               | -     | reserved  |
| 2 to 0 | TxLastBits[2:0] | -     | defines the number of bits of the last byte that shall be transmitted. 000 indicates that all bits of the last byte will be transmitted. TxLastBits[2:0] is automatically cleared after transmission.                             |

### 10.5.3 Page 2: Transmitter and control

#### 10.5.3.1 Page register

Selects the page register; see [Section 10.5.1.1 “Page register” on page 46](#).

#### 10.5.3.2 TxControl register

Controls the logical behavior of the antenna pins TX1 and TX2.

**Table 69. TxControl register (address: 11h) reset value: 0101 1000b, 58h bit allocation**

| Bit    | 7   | 6                    | 5 | 4   | 3      | 2     | 1       | 0       |
|--------|-----|----------------------|---|-----|--------|-------|---------|---------|
| Symbol | 0   | ModulatorSource[1:0] |   | 1   | TX2Inv | TX2Cw | TX2RFEn | TX1RFEn |
| Access | R/W | R/W                  |   | R/W | R/W    | R/W   | R/W     | R/W     |

**Table 70. TxControl register bit descriptions**

| Bit    | Symbol               | Value | Description   |
|--------|----------------------|-------|---|
| 7      | 0                    | -     | this value must not be changed  |
| 6 to 5 | ModulatorSource[1:0] |       | selects the source for the modulator input:   |
|        |                      | 00    | modulator input is LOW  |
|        |                      | 01    | modulator input is HIGH   |
|        |                      | 10    | modulator input is the internal encoder   |
|        |                      | 11    | modulator input is pin MFIN   |
| 4      | 1                    | -     | this value must not be changed  |
| 3      | TX2Inv               | 1     | delivers an inverted 13.56 MHz energy carrier output signal on pin TX2                          |
| 2      | TX2Cw                | 1     | delivers a continuously unmodulated 13.56 MHz energy carrier output signal on pin TX2           |
|        |                      | 0     | enables modulation of the 13.56 MHz energy carrier  |
| 1      | TX2RFEn              | 1     | the output signal on pin TX2 is the 13.56 MHz energy carrier modulated by the transmission data |
|        |                      | 0     | TX2 is driven at a constant output level  |
| 0      | TX1RFEn              | 1     | the output signal on pin TX1 is the 13.56 MHz energy carrier modulated by the transmission data |
|        |                      | 0     | TX1 is driven at a constant output level  |

**10.5.3.3 CwConductance register**

Selects the conductance of the antenna driver pins TX1 and TX2.

**Table 71. CwConductance register (address: 12h) reset value: 0011 1111b, 3Fh bit allocation**

| Bit    | 7   | 6 | 5            | 4 | 3 | 2 | 1 | 0 |
|--------|-----|---|--------------|---|---|---|---|---|
| Symbol | 00  |   | GsCfgCW[5:0] |   |   |   |   |   |
| Access | R/W |   | R/W          |   |   |   |   |   |

**Table 72. CwConductance register bit descriptions**

| Bit    | Symbol       | Value | Description   |
|--------|--------------|-------|---|
| 7 to 6 | 00           | 0     | these values must not be changed  |
| 5 to 0 | GsCfgCW[5:0] | -     | defines the conductance register value for the output driver. This can be used to regulate the output power/current consumption and operating distance. |

See [Section 9.9.3.1](#) for detailed information about GsCfgCW[5:0].

**10.5.3.4 PreSet13 register**

These bit settings must not be changed.

**Table 73. PreSet13 register (address: 13h) reset value: 0011 1111b, 3Fh bit allocation**

| Bit    | 7   | 6 | 5     | 4 | 3 | 2 | 1 | 0 |
|--------|-----|---|-------|---|---|---|---|---|
| Symbol | 00  |   | 11111 |   |   |   |   |   |
| Access | R/W |   | R/W   |   |   |   |   |   |

**Table 74. PreSet13 register bit descriptions**

| Bit    | Symbol | Value | Description                      |
|--------|--------|-------|----------------------------------|
| 7 to 6 | 00     | 0     | these values must not be changed |
| 5 to 0 | 11111  | -     | these values must not be changed |



### 10.5.3.5 CoderControl register

Sets the clock rate and the coding mode.

**Table 75. CoderControl register (address: 14h) reset value: 0001 1001b, 19h bit allocation**

| Bit    | 7   | 6 | 5              | 4 | 3 | 2   | 1 | 0 |
|--------|-----|---|----------------|---|---|-----|---|---|
| Symbol | 00  |   | CoderRate[2:0] |   |   | 001 |   |   |
| Access | R/W |   | R/W            |   |   | R/W |   |   |

**Table 76. CoderControl register bit descriptions**

| Bit    | Symbol         | Value | Description  |
|--------|----------------|-------|--|
| 7 to 6 | 00             | -     | these values must not be changed                             |
| 6      | 0              | -     | this value must not be changed                               |
| 5 to 3 | CoderRate[2:0] |       | this register defines the clock rate for the encoder circuit |
|        |                | 000   | MIFARE 848 kBd   |
|        |                | 001   | MIFARE 424 kBd   |
|        |                | 010   | MIFARE 212 kBd   |
|        |                | 011   | MIFARE 106 kBd; ISO/IEC 14443 A                              |
|        |                | 100   | reserved   |
|        |                | 101   | reserved   |
|        |                | 110   | reserved   |
|        |                | 111   | reserved   |
| 2 to 0 | 001            | -     | these values must not be changed                             |

### 10.5.3.6 ModWidth register

Selects the pulse-modulation width.

**Table 77. ModWidth register (address: 15h) reset value: 0001 0011b, 13h bit allocation**

| Bit    | 7             | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------|---------------|---|---|---|---|---|---|---|
| Symbol | ModWidth[7:0] |   |   |   |   |   |   |   |
| Access | R/W           |   |   |   |   |   |   |   |

**Table 78. ModWidth register bit descriptions**

| Bit    | Symbol        | Description   |
|--------|---------------|---|
| 7 to 0 | ModWidth[7:0] | defines the width of the modulation pulse based on<br>$t_{\text{mod}} = 2 \cdot (\text{ModWidth} + 1) / f_{\text{clk}}$ |

### 10.5.3.7 PreSet16 register

These bit settings must not be changed.

**Table 79. PreSet16 register (address: 16h) reset value: 0000 0000b, 00h bit allocation**

| Bit    | 7 | 6 | 5 | 4 | 3        | 2 | 1 | 0 |
|--------|---|---|---|---|----------|---|---|---|
| Symbol |   |   |   |   | 00000000 |   |   |   |
| Access |   |   |   |   | R/W      |   |   |   |

**Table 80. PreSet16 register bit descriptions**

| Bit    | Symbol   | Value | Description                      |
|--------|----------|-------|----------------------------------|
| 7 to 0 | 00000000 | 0     | these values must not be changed |

### 10.5.3.8 PreSet17 register

These bit settings must not be changed.

**Table 81. PreSet17 register (address: 17h) reset value: 0000 0000b, 00h bit allocation**

| Bit    | 7 | 6 | 5 | 4 | 3        | 2 | 1 | 0 |
|--------|---|---|---|---|----------|---|---|---|
| Symbol |   |   |   |   | 00000000 |   |   |   |
| Access |   |   |   |   | R/W      |   |   |   |

**Table 82. PreSet17 register bit descriptions**

| Bit    | Symbol   | Value | Description                      |
|--------|----------|-------|----------------------------------|
| 7 to 0 | 00000000 | 0     | these values must not be changed |

## 10.5.4 Page 3: Receiver and decoder control

### 10.5.4.1 Page register

Selects the page register; see [Section 10.5.1.1 “Page register” on page 46](#).

### 10.5.4.2 RxControl1 register

Controls receiver operation.

**Table 83. RxControl1 register (address: 19h) reset value: 0111 0011b, 73h bit allocation**

| Bit    | 7               | 6 | 5 | 4   | 3   | 2     | 1         | 0 |
|--------|-----------------|---|---|-----|-----|-------|-----------|---|
| Symbol | SubCPulses[2:0] |   |   | 1   | 0   | LPOff | Gain[1:0] |   |
| Access | R/W             |   |   | R/W | R/W | R/W   | R/W       |   |

**Table 84. RxControl1 register bit descriptions**

| Bit    | Symbol          | Value | Description  |
|--------|-----------------|-------|--|
| 7 to 5 | SubCPulses[2:0] |       | defines the number of subcarrier pulses for each bit     |
|        |                 | 000   | 1 pulse for each bit                                     |
|        |                 | 001   | 2 pulses for each bit                                    |
|        |                 | 010   | 4 pulses for each bit                                    |
|        |                 | 011   | 8 pulses for each bit ISO/IEC 14443 A                    |
|        |                 | 100   | 16 pulses  |
|        |                 | 101   | reserved   |
|        |                 | 110   | reserved   |
|        |                 | 111   | reserved   |
| 4      | 1               | 1     | this value must not be changed                           |
| 3      | 0               | 0     | this value must not be changed                           |
| 2      | LPOff           | 0     | switches off a low-pass filter at the internal amplifier |
| 1 to 0 | Gain[1:0]       |       | defines the receiver's signal voltage gain factor        |
|        |                 | 00    | 20 dB gain factor  |
|        |                 | 01    | 24 dB gain factor  |
|        |                 | 10    | 31 dB gain factor  |
|        |                 | 11    | 35 dB gain factor  |

### 10.5.4.3 DecoderControl register

Controls decoder operation.

**Table 85. DecoderControl register (address: 1Ah) reset value: 0000 1000b, 08h bit allocation**

| Bit    | 7   | 6          | 5             | 4   | 3   | 2   | 1 | 0        |
|--------|-----|------------|---------------|-----|-----|-----|---|----------|
| Symbol | 0   | RxMultiple | ZeroAfterColl | 0   | 1   | 00  |   | RxCoding |
| Access | R/W | R/W        | R/W           | R/W | R/W | R/W |   | R/W      |

**Table 86. DecoderControl register bit descriptions**

| Bit    | Symbol        | Value | Description  |
|--------|---------------|-------|--|
| 7      | 0             | -     | this value must not be changed   |
| 6      | RxMultiple    | 0     | after receiving one frame, the receiver is deactivated   |
|        |               | 1     | enables reception of more than one frame   |
| 5      | ZeroAfterColl | 1     | any bits received after a bit-collision are masked to zero. This helps to resolve the anti-collision procedure as defined in ISO/IEC 14443 A |
| 4      | 0             | 0     | this value must not be changed   |
| 3      | 1             | 1     | this value must not be changed   |
| 2 to 1 | 00            | 0     | these values must not be changed   |
| 0      | RxCoding      | 0     | Manchester encoding  |
|        |               | 1     | BPSK encoding  |

### 10.5.4.4 BitPhase register

Selects the bit-phase between transmitter and receiver clock.

**Table 87. BitPhase register (address: 1Bh) reset value: 1010 1101b, ADh bit allocation**

| Bit    | 7             | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------|---------------|---|---|---|---|---|---|---|
| Symbol | BitPhase[7:0] |   |   |   |   |   |   |   |
| Access | R/W           |   |   |   |   |   |   |   |

**Table 88. BitPhase register bit descriptions**

| Bit    | Symbol        | Description   |
|--------|---------------|---|
| 7 to 0 | BitPhase[7:0] | defines the phase relationship between transmitter and receiver clock<br><b>Remark:</b> The correct value of this register is essential for proper operation. |

#### 10.5.4.5 RxThreshold register

Selects thresholds for the bit decoder.

**Table 89. RxThreshold register (address: 1Ch) reset value: 1111 1111b, FFh bit allocation**

| Bit    | 7             | 6 | 5 | 4 | 3              | 2 | 1 | 0 |
|--------|---------------|---|---|---|----------------|---|---|---|
| Symbol | MinLevel[3:0] |   |   |   | CollLevel[3:0] |   |   |   |
| Access | R/W           |   |   |   | R/W            |   |   |   |

**Table 90. RxThreshold register bit descriptions**

| Bit    | Symbol         | Description   |
|--------|----------------|---|
| 7 to 4 | MinLevel[3:0]  | the minimum signal strength the decoder will accept. If the signal strength is below this level, it is not evaluated.   |
| 3 to 0 | CollLevel[3:0] | the minimum signal strength the decoder input that must be reached by the weaker half-bit of the Manchester encoded signal to generate a bit-collision (relative to the amplitude of the stronger half-bit) |

#### 10.5.4.6 BPSKDemControl

Controls BPSK demodulation.

**Table 91. BPSKDemControl register (address: 1Dh) reset value: 0001 1110b, 1Eh bit allocation**

| Bit    | 7   | 6 | 5 | 4            | 3         | 2 | 1         | 0 |
|--------|-----|---|---|--------------|-----------|---|-----------|---|
| Symbol | 000 |   |   | FilterAmpDet | TauD[1:0] |   | TauB[1:0] |   |
| Access | R/W |   |   | R/W          | R/W       |   | R/W       |   |

**Table 92. BPSKDemControl register bit descriptions**

| Bit    | Symbol       | Value | Description   |
|--------|--------------|-------|---|
| 7 to 5 | 000          | 0     | these values must not be changed                                    |
| 4      | FilterAmpDet | -     | switches on a high-pass filter for amplitude detection              |
| 3 to 2 | TauD[1:0]    | -     | changes the time constant of the internal PLL whilst receiving data |
| 1 to 0 | TauB[1:0]    | -     | changes the time constant of the internal PLL during data bursts    |

### 10.5.4.7 RxControl2 register

Controls decoder behavior and defines the input source for the receiver.

**Table 93. RxControl2 register (address: 1Eh) reset value: 0100 0001b, 41h bit allocation**

| Bit    | 7          | 6        | 5 | 4    | 3 | 2 | 1                  | 0 |
|--------|------------|----------|---|------|---|---|--------------------|---|
| Symbol | RcvClkSell | RxAutoPD |   | 0000 |   |   | DecoderSource[1:0] |   |
| Access | R/W        | R/W      |   | R/W  |   |   | R/W                |   |

**Table 94. RxControl2 register bit descriptions**

| Bit    | Symbol             | Value | Description   |
|--------|--------------------|-------|---|
| 7      | RcvClkSell         | 1     | I-clock is used as the receiver clock <sup>[1]</sup>  |
|        |                    | 0     | Q-clock is used as the receiver clock <sup>[1]</sup>  |
| 6      | RxAutoPD           | 1     | receiver circuit is automatically switched on before receiving and switched off afterwards. This can be used to reduce current consumption. |
|        |                    | 0     | receiver is always activated  |
| 5 to 2 | 0000               | 0     | these values must not be changed  |
| 1 to 0 | DecoderSource[1:0] |       | selects the source for the decoder input  |
|        |                    | 00    | LOW   |
|        |                    | 01    | internal demodulator  |
|        |                    | 10    | a subcarrier modulated Manchester encoded signal on pin MFIN  |
|        |                    | 11    | a baseband Manchester encoded signal on pin MFIN  |

[1] I-clock and Q-clock are 90° phase-shifted from each other.

### 10.5.4.8 ClockQControl register

Controls clock generation for the 90° phase-shifted Q-clock.

**Table 95. ClockQControl register (address: 1Fh) reset value: 000x xxxxb, xxh bit allocation**

| Bit    | 7          | 6         | 5   | 4 | 3 | 2              | 1 | 0 |
|--------|------------|-----------|-----|---|---|----------------|---|---|
| Symbol | ClkQ180Deg | ClkQCalib | 0   |   |   | ClkQDelay[4:0] |   |   |
| Access | R          | R/W       | R/W |   |   | D              |   |   |

**Table 96. ClockQControl register bit descriptions**

| Bit    | Symbol         | Value | Description   |
|--------|----------------|-------|---|
| 7      | ClkQ180Deg     | 1     | Q-clock is phase-shifted more than 180° compared to the I-clock   |
|        |                | 0     | Q-clock is phase-shifted less than 180° compared to the I-clock   |
| 6      | ClkQCalib      | 0     | Q-clock is automatically calibrated after the reset phase and after data reception from the card  |
|        |                | 1     | no calibration is performed automatically   |
| 5      | 0              | 0     | this value must not be changed  |
| 4 to 0 | ClkQDelay[4:0] | -     | this register shows the number of delay elements used to generate a 90° phase-shift of the I-clock to obtain the Q-clock. It can be written directly by the microprocessor or by the automatic calibration cycle. |

### 10.5.5 Page 4: RF Timing and channel redundancy

#### 10.5.5.1 Page register

Selects the page register; see [Section 10.5.1.1 “Page register” on page 46](#).

#### 10.5.5.2 RxWait register

Selects the time interval after transmission, before the receiver starts.

**Table 97. RxWait register (address: 21h) reset value: 0000 0101b, 06h bit allocation**

| Bit    | 7           | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------|-------------|---|---|---|---|---|---|---|
| Symbol | RxWait[7:0] |   |   |   |   |   |   |   |
| Access | R/W         |   |   |   |   |   |   |   |

**Table 98. RxWait register bit descriptions**

| Bit    | Symbol      | Function  |
|--------|-------------|---|
| 7 to 0 | RxWait[7:0] | after data transmission, the activation of the receiver is delayed for RxWait bit-clock cycles. During this frame guard time any signal on pin RX is ignored. |

#### 10.5.5.3 ChannelRedundancy register

Selects kind and mode of checking the data integrity on the RF channel.

**Table 99. ChannelRedundancy register (address: 22h) reset value: 0000 0011b, 03h bit allocation**

| Bit    | 7   | 6       | 5    | 4       | 3       | 2         | 1        | 0   |
|--------|-----|---------|------|---------|---------|-----------|----------|-----|
| Symbol | 00  | CRC3309 | CRC8 | RxCRCEn | TxCRCEn | ParityOdd | ParityEn |     |
| Access | R/W | R/W     | R/W  | R/W     | R/W     | R/W       | R/W      | R/W |

**Table 100. ChannelRedundancy bit descriptions**

| Bit    | Symbol                 | Value | Function  |
|--------|------------------------|-------|---|
| 7 to 6 | 00                     | 0     | this value must not be changed  |
| 5      | CRC3309 <sup>[1]</sup> | 1     | CRC calculation is performed using ISO/IEC 3309 and ISO/IEC 15693   |
|        |                        | 0     | CRC calculation is performed using ISO/IEC 14443 A  |
| 4      | CRC8                   | 1     | an 8-bit CRC is calculated  |
|        |                        | 0     | a 16-bit CRC is calculated  |
| 3      | RxCRCEn                | 1     | the last byte(s) of a received frame are interpreted as CRC bytes. If the CRC is correct, the CRC bytes are not passed to the FIFO. If the CRC bytes are incorrect, the CRCErr flag is set. |
|        |                        | 0     | no CRC is expected  |
| 2      | TxCRCEn                | 1     | a CRC is calculated over the transmitted data and the CRC bytes are appended to the data stream   |
|        |                        | 0     | no CRC is transmitted   |

**Table 100. ChannelRedundancy bit descriptions ...continued**

| Bit | Symbol    | Value | Function   |
|-----|-----------|-------|--|
| 1   | ParityOdd | 1     | odd parity is generated or expected <sup>[2]</sup>   |
|     |           | 0     | even parity is generated or expected   |
| 0   | ParityEn  | 1     | a parity bit is inserted in the transmitted data stream after each byte and expected in the received data stream after each byte (MIFARE, ISO/IEC 14443 A) |
|     |           | 0     | no parity bit is inserted or expected  |

[1] With ISO/IEC 14443 A, this bit must be set to logic 0.

[2] With ISO/IEC 14443 A, this bit must be set to logic 1.

### 10.5.5.4 CRCPresetLSB register

LSB of the preset value for the CRC register.

**Table 101. CRCPresetLSB register (address: 23h) reset value: 0101 0011b, 63h bit allocation**

| Bit    | 7                 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------|-------------------|---|---|---|---|---|---|---|
| Symbol | CRCPresetLSB[7:0] |   |   |   |   |   |   |   |
| Access | R/W               |   |   |   |   |   |   |   |

**Table 102. CRCPresetLSB register bit descriptions**

| Bit    | Symbol            | Description   |
|--------|-------------------|---|
| 7 to 0 | CRCPresetLSB[7:0] | defines the start value for CRC calculation. This value is loaded into the CRC at the beginning of transmission, reception and the CalcCRC command (if CRC calculation is enabled). |

### 10.5.5.5 CRCPresetMSB register

MSB of the preset value for the CRC register.

**Table 103. CRCPresetMSB register (address: 24h) reset value: 0101 0011b, 63h bit allocation**

| Bit    | 7                 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------|-------------------|---|---|---|---|---|---|---|
| Symbol | CRCPresetMSB[7:0] |   |   |   |   |   |   |   |
| Access | R/W               |   |   |   |   |   |   |   |

**Table 104. CRCPresetMSB bit descriptions**

| Bit    | Symbol            | Description  |
|--------|-------------------|--|
| 7 to 0 | CRCPresetMSB[7:0] | defines the starting value for CRC calculation. This value is loaded into the CRC at the beginning of transmission, reception and the CalcCRC command (if the CRC calculation is enabled)<br><b>Remark:</b> This register is not relevant if CRC8 is set to logic 1. |

### 10.5.5.6 PreSet25 register

These values must not be changed.

**Table 105. PreSet25 register (address: 25h) reset value: 0000 0000b, 00h bit allocation**

| Bit    | 7        | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------|----------|---|---|---|---|---|---|---|
| Symbol | 00000000 |   |   |   |   |   |   |   |
| Access | R/W      |   |   |   |   |   |   |   |



**Table 106. PreSet25 register bit descriptions**

| Bit    | Symbol   | Value | Description                      |
|--------|----------|-------|----------------------------------|
| 7 to 0 | 00000000 | 0     | these values must not be changed |

**10.5.5.7 MFOUSelect register**

Selects the internal signal applied to pin MFOUT.

**Table 107. MFOUSelect register (address: 26h) reset value: 0000 0000b, 00h bit allocation**

| Bit    | 7     | 6 | 5 | 4 | 3 | 2               | 1 | 0 |
|--------|-------|---|---|---|---|-----------------|---|---|
| Symbol | 00000 |   |   |   |   | MFOUSelect[2:0] |   |   |
| Access | R/W   |   |   |   |   | R/W             |   |   |

**Table 108. MFOUSelect register bit descriptions**

| Bit    | Symbol          | Value | Description   |
|--------|-----------------|-------|---|
| 7 to 3 | 00000           | 0     | these values must not be changed  |
| 2 to 0 | MFOUSelect[2:0] |       | defines which signal is routed to pin MFOUT:  |
|        |                 | 000   | constant LOW  |
|        |                 | 001   | constant HIGH   |
|        |                 | 010   | modulation signal (envelope) from the internal encoder, (Miller coded)                      |
|        |                 | 011   | serial data stream, not Miller encoded  |
|        |                 | 100   | output signal of the energy carrier demodulator (card modulation signal) <sup>[1]</sup>     |
|        |                 | 101   | output signal of the subcarrier demodulator (Manchester encoded card signal) <sup>[1]</sup> |
|        |                 | 110   | reserved  |
|        |                 | 111   | reserved  |

[1] Only valid for MIFARE and ISO/IEC 14443 A communication at 106 kBd.

**10.5.5.8 PreSet27 register****Table 109. PreSet27 (address: 27h) reset value: xxxx xxxxb, xxh bit allocation**

| Bit    | 7       | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------|---------|---|---|---|---|---|---|---|
| Symbol | xxxxxxx |   |   |   |   |   |   |   |
| Access | W       |   |   |   |   |   |   |   |

**Table 110. PreSet27 register bit descriptions**

| Bit    | Symbol  | Value | Description  |
|--------|---------|-------|--|
| 7 to 0 | xxxxxxx | 0     | these values can be logic 1 or logic 0. These values must not be changed |

**10.5.6 Page 5: FIFO, timer and IRQ pin configuration**

**10.5.6.1 Page register**

Selects the page register; see [Section 10.5.1.1 “Page register” on page 46](#).

**10.5.6.2 FIFOLevel register**

Defines the levels for FIFO underflow and overflow warning.

**Table 111. FIFOLevel register (address: 29h) reset value: 0000 1000b, 08h bit allocation**

| Bit    | 7   | 6 | 5               | 4 | 3 | 2 | 1 | 0 |
|--------|-----|---|-----------------|---|---|---|---|---|
| Symbol | 00  |   | WaterLevel[5:0] |   |   |   |   |   |
| Access | R/W |   | R/W             |   |   |   |   |   |

**Table 112. FIFOLevel register bit descriptions**

| Bit    | Symbol          | Description   |
|--------|-----------------|---|
| 7 to 6 | 00              | these values must not be changed  |
| 5 to 0 | WaterLevel[5:0] | defines, the warning level of a FIFO buffer overflow or underflow:<br>HiAlert is set to logic 1 if the remaining FIFO buffer space is equal to, or less than, WaterLevel[5:0] bits in the FIFO buffer.<br>LoAlert is set to logic 1 if equal to, or less than, WaterLevel[5:0] bits in the FIFO buffer. |

**10.5.6.3 TimerClock register**

Selects the divider for the timer clock.

**Table 113. TimerClock register (address: 2Ah) reset value: 0000 0111b, 07h bit allocation**

| Bit    | 7  | 6 | 5            | 4               | 3 | 2 | 1 | 0 |
|--------|----|---|--------------|-----------------|---|---|---|---|
| Symbol | 00 |   | TAutoRestart | TPreScaler[4:0] |   |   |   |   |
| Access | RW |   | RW           | RW              |   |   |   |   |

**Table 114. TimerClock register bit descriptions**

| Bit    | Symbol          | Value | Function   |
|--------|-----------------|-------|--|
| 7 to 6 | 00              | 0     | these values must not be changed   |
| 5      | TAutoRestart    | 1     | the timer automatically restarts its countdown from the TReloadValue[7:0] instead of counting down to zero   |
|        |                 | 0     | the timer decrements to zero and register InterruptIRq TimerIRq bit is set to logic 1  |
| 4 to 0 | TPreScaler[4:0] | -     | defines the timer clock frequency ( $f_{\text{TimerClock}}$ ). The TPreScaler[4:0] can be adjusted from 0 to 21. The following formula is used to calculate the TimerClock frequency ( $f_{\text{TimerClock}}$ ):<br>$f_{\text{TimerClock}} = 13.56 \text{ MHz} / 2^{\text{TPreScaler}} \text{ [MHz]}$ |

#### 10.5.6.4 TimerControl register

Selects start and stop conditions for the timer.

**Table 115. TimerControl register (address: 2Bh) reset value: 0000 0110b, 06h bit allocation**

| Bit    | 7    | 6 | 5 | 4 | 3          | 2            | 1           | 0             |
|--------|------|---|---|---|------------|--------------|-------------|---------------|
| Symbol | 0000 |   |   |   | TStopRxEnd | TStopRxBegin | TStartTxEnd | TStartTxBegin |
| Access | R/W  |   |   |   | R/W        | R/W          | R/W         | R/W           |

**Table 116. TimerControl register bit descriptions**

| Bit    | Symbol        | Value | Description  |
|--------|---------------|-------|--|
| 7 to 4 | 0000          | 0     | these values must not be changed   |
| 3      | TStopRxEnd    | 1     | the timer automatically stops when data reception ends   |
|        |               | 0     | the timer is not influenced by this condition  |
| 2      | TStopRxBegin  | 1     | the timer automatically stops when the first valid bit is received   |
|        |               | 0     | the timer is not influenced by this condition  |
| 1      | TStartTxEnd   | 1     | the timer automatically starts when data transmission ends. If the timer is already running, the timer restarts by loading TReloadValue[7:0] into the timer.       |
|        |               | 0     | the timer is not influenced by this condition  |
| 0      | TStartTxBegin | 1     | the timer automatically starts when the first bit is transmitted. If the timer is already running, the timer restarts by loading TReloadValue[7:0] into the timer. |
|        |               | 0     | the timer is not influenced by this condition  |

#### 10.5.6.5 TimerReload register

Defines the preset value for the timer.

**Table 117. TimerReload register (address: 2Ch) reset value: 0000 1010b, 0Ah bit allocation**

| Bit    | 7                 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------|-------------------|---|---|---|---|---|---|---|
| Symbol | TReloadValue[7:0] |   |   |   |   |   |   |   |
| Access | R/W               |   |   |   |   |   |   |   |

**Table 118. TimerReload register bit descriptions**

| Bit    | Symbol            | Description  |
|--------|-------------------|--|
| 7 to 0 | TReloadValue[7:0] | on a start event, the timer loads the TReloadValue[7:0] value. Changing this register only affects the timer on the next start event. If TReloadValue[7:0] is set to logic 0 the timer cannot start. |

### 10.5.6.6 IRQPinConfig register

Configures the output stage for pin IRQ.

**Table 119. IRQPinConfig register (address: 2Dh) reset value: 0000 0010b, 02h bit allocation**

| Bit    | 7      | 6 | 5 | 4 | 3 | 2 | 1      | 0           |
|--------|--------|---|---|---|---|---|--------|-------------|
| Symbol | 000000 |   |   |   |   |   | IRQInv | IRQPushPull |
| Access | R/W    |   |   |   |   |   | R/W    | R/W         |

**Table 120. IRQPinConfig register bit descriptions**

| Bit    | Symbol      | Value | Description  |
|--------|-------------|-------|--|
| 7 to 2 | 000000      | 0     | these values must not be changed                                 |
| 1      | IRQInv      | 1     | inverts the signal on pin IRQ with respect to bit IRq            |
|        |             | 0     | the signal on pin IRQ is not inverted and is the same as bit IRq |
| 0      | IRQPushPull | 1     | pin IRQ functions as a standard CMOS output pad                  |
|        |             | 0     | pin IRQ functions as an open-drain output pad                    |

### 10.5.6.7 PreSet2E register

**Table 121. PreSet2E register (address: 2Eh) reset value: xxxx xxxxb, xxh bit allocation**

| Bit    | 7        | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------|----------|---|---|---|---|---|---|---|
| Symbol | xxxxxxxx |   |   |   |   |   |   |   |
| Access | W        |   |   |   |   |   |   |   |

### 10.5.6.8 PreSet2F register

**Table 122. PreSet2F register (address: 2Fh) reset value: xxxx xxxxb, xxh bit allocation**

| Bit    | 7        | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------|----------|---|---|---|---|---|---|---|
| Symbol | xxxxxxxx |   |   |   |   |   |   |   |
| Access | W        |   |   |   |   |   |   |   |

## 10.5.7 Page 6: reserved

### 10.5.7.1 Page register

Selects the page register; see [Section 10.5.1.1 “Page register” on page 46](#).

### 10.5.7.2 Reserved registers 31h, 32h, 33h, 34h, 35h, 36h and 37h

**Table 123. Reserved registers (address: 31h, 32h, 33h, 34h, 35h, 36h, 37h) reset value: xxxx xxxxb, xxh bit allocation**

| Bit    | 7        | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------|----------|---|---|---|---|---|---|---|
| Symbol | xxxxxxxx |   |   |   |   |   |   |   |
| Access | W        |   |   |   |   |   |   |   |

**Remark:** These registers are reserved for future use.

**10.5.8 Page 7: Test control**

**10.5.8.1 Page register**

Selects the page register; see [Section 10.5.1.1 “Page register” on page 46](#).

**10.5.8.2 Reserved register 39h**

**Table 124. Reserved register (address: 39h) reset value: xxxx xxxxb, xxh bit allocation**

| Bit    | 7        | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------|----------|---|---|---|---|---|---|---|
| Symbol | xxxxxxxx |   |   |   |   |   |   |   |
| Access | W        |   |   |   |   |   |   |   |

**Remark:** This register is reserved for future use.

**10.5.8.3 TestAnaSelect register**

Selects analog test signals.

**Table 125. TestAnaSelect register (address: 3Ah) reset value: 0000 0000b, 00h bit allocation**

| Bit    | 7    | 6 | 5 | 4 | 3                  | 2 | 1 | 0 |
|--------|------|---|---|---|--------------------|---|---|---|
| Symbol | 0000 |   |   |   | TestAnaOutSel[4:0] |   |   |   |
| Access | W    |   |   |   | W                  |   |   |   |

**Table 126. TestAnaSelect bit descriptions**

| Bit    | Symbol             | Value | Description   |
|--------|--------------------|-------|---|
| 7 to 4 | 0000               | 0     | these values must not be changed  |
| 3 to 0 | TestAnaOutSel[4:0] |       | selects the internal analog signal to be routed to the AUX pin. See <a href="#">Section 15.2.2 on page 101</a> for detailed information. The settings are as follows: |
|        |                    | 0     | VMID  |
|        |                    | 1     | Vbandgap  |
|        |                    | 2     | VRxFollI  |
|        |                    | 3     | VRxFollQ  |
|        |                    | 4     | VRxAmpI   |
|        |                    | 5     | VRxAmpQ   |
|        |                    | 6     | VCorrNI   |
|        |                    | 7     | VCorrNQ   |
|        |                    | 8     | VCorrDI   |
|        |                    | 9     | VCorrDQ   |
|        |                    | A     | VEvalL  |
|        |                    | B     | VEvalR  |
|        |                    | C     | VTemp   |
|        |                    | D     | reserved  |
|        |                    | E     | reserved  |
|        |                    | F     | reserved  |

#### 10.5.8.4 Reserved register 3Bh

**Table 127. Reserved register (address: 3Bh) reset value: xxxx xxxxb, xxh bit allocation**

| Bit    | 7        | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------|----------|---|---|---|---|---|---|---|
| Symbol | xxxxxxxx |   |   |   |   |   |   |   |
| Access | W        |   |   |   |   |   |   |   |

**Remark:** This register is reserved for future use.

#### 10.5.8.5 Reserved register 3Ch

**Table 128. Reserved register (address: 3Ch) reset value: xxxx xxxxb, xxh bit allocation**

| Bit    | 7        | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------|----------|---|---|---|---|---|---|---|
| Symbol | xxxxxxxx |   |   |   |   |   |   |   |
| Access | W        |   |   |   |   |   |   |   |

**Remark:** This register is reserved for future use.

#### 10.5.8.6 TestDigiSelect register

Selects digital test mode.

**Table 129. TestDigiSelect register (address: 3Dh) reset value: xxxx xxxxb, xxh bit allocation**

| Bit    | 7             | 6                      | 5 | 4 | 3 | 2 | 1 | 0 |
|--------|---------------|------------------------|---|---|---|---|---|---|
| Symbol | SignalToMFOUT | TestDigiSignalSel[6:0] |   |   |   |   |   |   |
| Access | W             | W                      |   |   |   |   |   |   |

**Table 130. TestDigiSelect register bit descriptions**

| Bit    | Symbol                 | Value | Description   |
|--------|------------------------|-------|---|
| 7      | SignalToMFOUT          | 1     | overrides the MFOUTSelect[2:0] setting and routes the digital test signal defined with the TestDigiSignalSel[6:0] bits to pin MFOUT   |
|        |                        | 0     | MFOUTSelect[2:0] defines the signal on pin MFOUT  |
| 6 to 0 | TestDigiSignalSel[6:0] | -     | selects the digital test signal to be routed to pin MFOUT. Refer to <a href="#">Section 15.2.3 on page 102</a> for detailed information. The following lists the signal names for the TestDigiSignalSel[6:0] addresses: |
|        |                        | F4h   | s_data  |
|        |                        | E4h   | s_valid   |
|        |                        | D4h   | s_coll  |
|        |                        | C4h   | s_clock   |
|        |                        | B5h   | rd_sync   |
|        |                        | A5h   | wr_sync   |
|        |                        | 96h   | int_clock   |
|        |                        | 83h   | BPSK_out  |
|        |                        | E2h   | BPSK_sig  |

10.5.8.7 Reserved registers 3Eh, 3Fh

Table 131. Reserved register (address: 3Eh, 3Fh) reset value: xxxx xxxxb, xxh bit allocation

| Bit    | 7 | 6 | 5 | 4       | 3 | 2 | 1 | 0 |
|--------|---|---|---|---------|---|---|---|---|
| Symbol |   |   |   | xxxxxxx |   |   |   |   |
| Access |   |   |   | W       |   |   |   |   |

**Remark:** This register is reserved for future use.

11. MFRC530 command set

MFRC530 operation is determined by an internal state machine capable of performing a command set. The commands can be started by writing the command code to the Command register. Arguments and/or data necessary to process a command are mainly exchanged using the FIFO buffer.

- Each command needing a data stream (or data byte stream) as an input immediately processes the data in the FIFO buffer
- Each command that requires arguments only starts processing when it has received the correct number of arguments from the FIFO buffer
- The FIFO buffer is not automatically cleared at the start of a command. It is, therefore, possible to write command arguments and/or the data bytes into the FIFO buffer before starting a command.
- Each command (except the StartUp command) can be interrupted by the microprocessor writing a new command code to the Command register e.g. the Idle command.

11.1 MFRC530 command overview

Table 132. MFRC530 commands overview

| Command  | Value | Action   | FIFO communication      |               |
|----------|-------|--|-------------------------|---------------|
|          |       |  | Arguments and data sent | Data received |
| StartUp  | 3Fh   | runs the reset and initialization phase. See <a href="#">Section 11.1.2 on page 73</a> .<br><b>Remark:</b> This command can only be activated by Power-On or Hard resets.  | -                       | -             |
| Idle     | 00h   | no action; cancels execution of the current command. See <a href="#">Section 11.1.3 on page 73</a>   | -                       | -             |
| Transmit | 1Ah   | transmits data from the FIFO buffer to the card. See <a href="#">Section 11.2.1 on page 74</a>   | data stream             | -             |
| Receive  | 16h   | activates receiver circuitry. Before the receiver starts, the state machine waits until the time defined in the RxWait register has elapsed. See <a href="#">Section 11.2.2 on page 77</a> .<br><b>Remark:</b> This command may be used for test purposes only, since there is no timing relationship to the Transmit command. | -                       | data stream   |

Table 132. MFRC530 commands overview ...continued

| Command                   | Value | Action   | FIFO communication  |               |
|---------------------------|-------|--|---|---------------|
|                           |       |  | Arguments and data sent   | Data received |
| Transceive <sup>[1]</sup> | 1Eh   | transmits data from FIFO buffer to the card and automatically activates the receiver after transmission. The receiver waits until the time defined in the RxWait register has elapsed before starting. See <a href="#">Section 11.2.3 on page 80</a> .   | data stream   | data stream   |
| WriteE2                   | 01h   | reads data from the FIFO buffer and writes it to the EEPROM. See <a href="#">Section 11.3.1 on page 82</a> .   | start address LSB<br>start address MSB<br>data byte stream  | -             |
| ReadE2                    | 03h   | reads data from the EEPROM and sends it to the FIFO buffer. See <a href="#">Section 11.3.2 on page 84</a> .<br><b>Remark:</b> Keys cannot be read back   | start address LSB<br>start address MSB<br>number of data bytes  | data bytes    |
| LoadKeyE2                 | 0Bh   | copies a key from the EEPROM into the key buffer. See <a href="#">Section 11.6.1 on page 86</a> .  | start address LSB<br>start address MSB  | -             |
| LoadKey                   | 19h   | reads a key from the FIFO buffer and loads it into the key buffer. See <a href="#">Section 11.6.2 on page 86</a> .<br><b>Remark:</b> The key has to be prepared in a specific format (refer to <a href="#">Section 9.2.3.1 "Key format" on page 15</a> ) | byte 0 LSB<br>byte 1<br>...<br>byte 10<br>byte 11 MSB   | -             |
| Authent1                  | 0Ch   | performs the first part of card authentication using the Crypto1 algorithm. See <a href="#">Section 11.6.3 on page 87</a> .  | card Authent1 command<br>card block address<br>card serial number LSB<br>card serial number byte 1<br>card serial number byte 2<br>card serial number MSB | -             |
| Authent2                  | 14h   | performs the second part of card authentication using the Crypto1 algorithm. See <a href="#">Section 11.6.4 on page 87</a> .   | -   | -             |
| LoadConfig                | 07h   | reads data from EEPROM and initializes the MFRC530 registers. See <a href="#">Section 11.4.1 on page 84</a> .  | start address LSB<br>start address MSB  | -             |
| CalcCRC                   | 12h   | activates the CRC coprocessor<br><b>Remark:</b> The result of the CRC calculation is read from the CRCResultLSB and CRCResultMSB registers. See <a href="#">Section 11.4.2 on page 85</a> .  | data byte stream  | -             |

[1] This command is the combination of the Transmit and Receive commands.



### 11.1.1 Basic states

#### 11.1.2 StartUp command 3Fh

Table 133. StartUp command 3Fh

| Command | Value | Action                                  | Arguments and data | Returned data |
|---------|-------|---|--------------------|---------------|
| StartUp | 3Fh   | runs the reset and initialization phase | -                  | -             |

**Remark:** This command can only be activated by a Power-On or Hard reset.

The StartUp command runs the reset and initialization phases. It does not need or return, any data. It cannot be activated by the microprocessor but is automatically started after one of the following events:

- Power-On Reset (POR) caused by power-up on pin DVDD or on pin AVDD
- Negative edge on pin RSTPD

The reset phase comprises an asynchronous reset and configuration of certain register bits. The initialization phase configures several registers with values stored in the EEPROM.

When the StartUp command finishes, the Idle command is automatically executed.

**Remark:**

- The microprocessor must not write to the MFRC530 while it is still executing the StartUp command. To avoid this, the microprocessor polls for the Idle command to determine when the initialization phase has finished; see [Section 9.7.4 on page 27](#).
- When the StartUp command is active, it is only possible to read from the Page 0 register.
- The StartUp command cannot be interrupted by the microprocessor.

#### 11.1.3 Idle command 00h

Table 134. Idle command 00h

| Command | Value | Action                                       | Arguments and data | Returned data |
|---------|-------|--|--------------------|---------------|
| Idle    | 00h   | no action; cancels current command execution | -                  | -             |

The Idle command switches the MFRC530 to its inactive state where it waits for the next command. It does not need or return, any data.

The device automatically enters the idle state when a command finishes. When this happens, the MFRC530 sends an interrupt request by setting bit IdleIRq. When triggered by the microprocessor, the Idle command can be used to stop execution of all other commands (except the StartUp command) but this does not generate an interrupt request (IdleIRq).

**Remark:** Stopping command execution with the Idle command does not clear the FIFO buffer.

## 11.2 Commands for card communication

The MFRC530 is a fully ISO/IEC 14443 A compliant reader IC. This enables the command set to be more flexible and generalized when compared to dedicated MIFARE reader ICs. [Section 11.2.1](#) to [Section 11.2.5](#) describe the command set for ISO/IEC 14443 A card communication and related communication protocols.

### 11.2.1 Transmit command 1Ah

Table 135. Transmit command 1Ah

| Command  | Value | Action                                  | Arguments and data | Returned data |
|----------|-------|---|--------------------|---------------|
| Transmit | 1Ah   | transmits data from FIFO buffer to card | data stream        | -             |

The Transmit command reads data from the FIFO buffer and sends it to the transmitter. It does not return any data. The Transmit command can only be started by the microprocessor.

#### 11.2.1.1 Using the Transmit command

To transmit data, one of the following sequences can be used:

1. All data to be transmitted to the card is written to the FIFO buffer while the Idle command is active. Then the command code for the Transmit command is written to the Command register.
 

**Remark:** This is possible for transmission of a data stream up to 64 bytes.
2. The command code for the Transmit command is stored in the Command register. Since there is not any data available in the FIFO buffer, the command is only enabled but transmission is not activated. Data transmission starts when the first data byte is written to the FIFO buffer. To generate a continuous data stream on the RF interface, the microprocessor must write the subsequent data bytes into the FIFO buffer in time.
 

**Remark:** This allows transmission of any data stream length but it requires data to be written to the FIFO buffer in time.
3. Part of the data transmitted to the card is written to the FIFO buffer while the Idle command is active. Then the command code for the Transmit command is written to the Command register. While the Transmit command is active, the microprocessor can send further data to the FIFO buffer. This is then appended by the transmitter to the transmitted data stream.
 

**Remark:** This allows transmission of any data stream length but it requires data to be written to the FIFO buffer in time.

When the transmitter requests the next data byte to ensure the data stream on the RF interface is continuous and the FIFO buffer is empty, the Transmit command automatically exits. This causes the internal state machine to change its state from transmit to idle.

When the data transmission to the card is finished, the TxIRq flag is set by the MFRC530 to indicate to the microprocessor transmission is complete.

**Remark:** If the microprocessor overwrites the transmit code in the Command register with another command, transmission stops immediately on the next clock cycle. This can produce output signals that are not in accordance with ISO/IEC 14443 A.

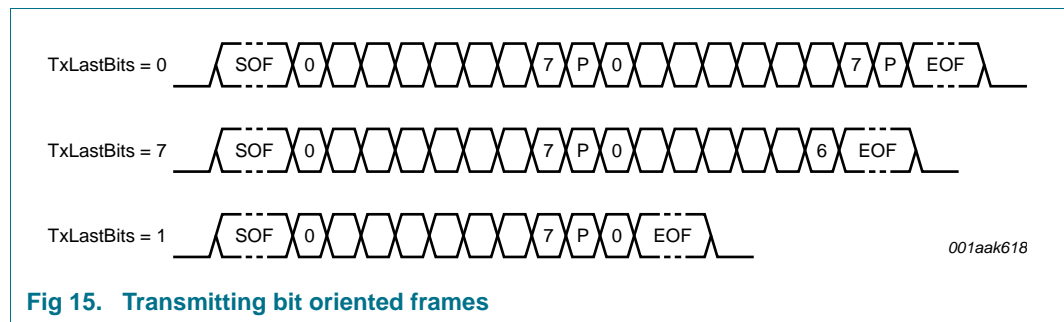
**11.2.1.2 RF channel redundancy and framing**

Each ISO/IEC 14443 A frame transmitted consists of a Start Of Frame (SOF) pattern, followed by the data stream and is closed by an End Of Frame (EOF) pattern. These different phases of the transmission sequence can be monitored using the PrimaryStatus register ModemState[2:0] bits; see [Section 11.2.4 on page 80](#).

Depending on the setting of the ChannelRedundancy register bit TxCRCEn, the CRC is calculated and appended to the data stream. The CRC is calculated according to the settings in the ChannelRedundancy register. Parity generation is handled according to the ChannelRedundancy register ParityEn and ParityOdd bits settings.

**11.2.1.3 Transmission of bit oriented frames**

The transmitter can be configured to send an incomplete last byte. To achieve this the BitFraming register's TxLastBits[2:0] bits must be set at above zero (for example, 1). This is shown in [Figure 15](#).



**Fig 15. Transmitting bit oriented frames**

[Figure 15](#) shows the data stream when bit ParityEn is set in the ChannelRedundancy register. All fully transmitted bytes are followed by a parity check bit but the incomplete byte is not followed by a parity check bit. After transmission, the TxLastBits[2:0] bits are automatically cleared.

**Remark:** If the TxLastBits[2:0] bits are not equal to zero, CRC generation must be disabled. This is done by clearing the ChannelRedundancy register TxCRCEn bit.

**11.2.1.4 Transmission of frames with more than 64 bytes**

To generate frames of more than 64 bytes, the microprocessor must write data to the FIFO buffer while the Transmit command is active. The state machine checks the FIFO buffer status when it starts transmitting the last bit of the data stream; the check time is shown in [Figure 16](#) with arrows.

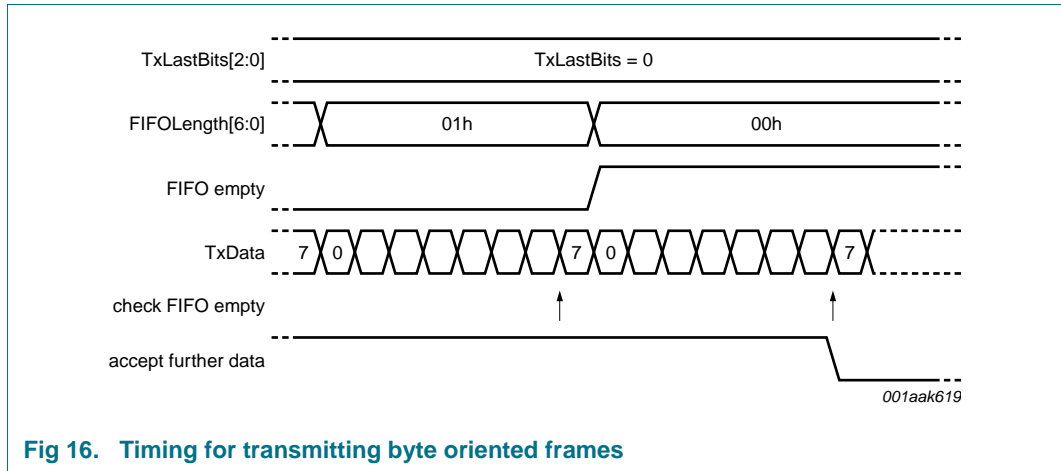


Fig 16. Timing for transmitting byte oriented frames

As long as the internal signal accept further data is logic 1, data can be written to the FIFO buffer. The MFRC530 appends this data to the data stream transmitted using the RF interface.

If the internal accept further data signal is logic 0, the transmission terminates. All data written to the FIFO buffer after the accept further data signal was set to logic 0 is not transmitted, however, it remains in the FIFO buffer.

**Remark:** If parity generation is enabled (ParityEn = logic 1), the parity bit is the last bit transmitted. This delays the accept further data signal by a duration of one bit.

If the TxLastBits[2:0] bits are not zero, the last byte is not transmitted completely. Only the number of bits set by TxLastBits[2:0], starting with the least significant bit are transmitted. This means that the internal state machine has to check the FIFO buffer status at an earlier point in time; see Figure 17.

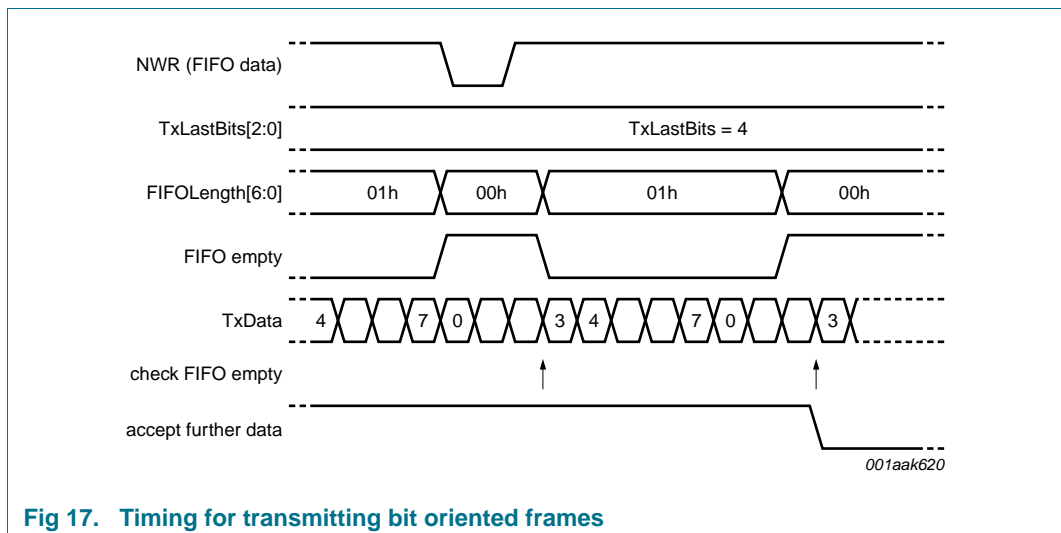


Fig 17. Timing for transmitting bit oriented frames

Since in this example TxLastBits[2:0] = 4, transmission stops after bit 3 is transmitted and the frame is completed with an EOF, if configured.

[Figure 17](#) also shows write access to the FIFOData register just before the FIFO buffer's status is checked. This leads to FIFO empty state being held LOW which keeps the accept further data active. The new byte written to the FIFO buffer is transmitted using the RF interface.

Accept further data is only changed by the check FIFO empty function. This function verifies FIFO empty for one bit duration before the last expected bit transmission.

**Table 136. Transmission of frames of more than 64 bytes**

| Frame definition     | Verification at:          |
|----------------------|---------------------------|
| 8-bit with parity    | 8 <sup>th</sup> bit       |
| 8-bit without parity | 7 <sup>th</sup> bit       |
| x-bit without parity | (x – 1) <sup>th</sup> bit |

### 11.2.2 Receive command 16h

**Table 137. Receive command 16h**

| Command | Value | Action                       | Arguments and data | Returned data |
|---------|-------|------------------------------|--------------------|---------------|
| Receive | 16h   | activates receiver circuitry | -                  | data stream   |

The Receive command activates the receiver circuitry. All data received from the RF interface is written to the FIFO buffer. The Receive command can be started either using the microprocessor or automatically during execution of the Transceive command.

**Remark:** This command can only be used for test purposes since there is no timing relationship to the Transmit command.

#### 11.2.2.1 Using the Receive command

After starting the Receive command, the internal state machine decrements to the RxWait register value on every bit-clock. The analog receiver circuitry is prepared and activated from 3 down to 1. When the counter reaches 0, the receiver starts monitoring the incoming signal at the RF interface.

When the signal strength reaches a level higher than the RxThreshold register MinLevel[3:0] bits value, it starts decoding. The decoder stops when the signal can longer be detected on the receiver input pin RX. The decoder sets bit RxIRq indicating receive termination.

The different phases of the receive sequence are monitored using the PrimaryStatus register ModemState[2:0] bits; see [Section 11.2.4 on page 80](#).

**Remark:** Since the counter values from 3 to 0 are needed to initialize the analog receiver circuitry, the minimum value for RxWait[7:0] is 3.

### 11.2.2.2 RF channel redundancy and framing

The decoder expects the SOF pattern at the beginning of each data stream. When the SOF is detected, it activates the serial-to-parallel converter and gathers the incoming data bits. Every completed byte is forwarded to the FIFO buffer.

If an EOF pattern is detected or the signal strength falls below the RxThreshold register MinLevel[3:0] bits setting, both the receiver and the decoder stop. Then the Idle command is entered and an appropriate response for the microprocessor is generated (interrupt request activated, status flags set).

When the ChannelRedundancy register bit RxCRCEn is set, a CRC block is expected. The CRC block can be one byte or two bytes depending on the ChannelRedundancy register CRC8 bit setting.

**Remark:** If the CRC block received is correct, it is not sent to the FIFO buffer. This is realized by shifting the incoming data bytes through an internal buffer of either one or two bytes (depending on the defined CRC). The CRC block remains in this internal buffer. Consequently, all data bytes in the FIFO buffer are delayed by one or two bytes. If the CRC fails, all received bytes are sent to the FIFO buffer including the faulty CRC.

If ParityEn is set in the ChannelRedundancy register, a parity bit is expected after each byte. If ParityOdd = logic 1, the expected parity is odd, otherwise even parity is expected.

### 11.2.2.3 Collision detection

If more than one card is within the RF field during the card selection phase, they both respond simultaneously. The MFRC530 supports the algorithm defined in ISO/IEC 14443 A to resolve card serial number data collisions by performing the anti-collision procedure. The basis for this procedure is the ability to detect bit-collisions.

Bit-collision detection is supported by the Manchester coding bit encoding scheme used in the MFRC530. If in the first and second half-bit of a subcarrier, modulation is detected, instead of forwarding a 1-bit or 0-bit, a bit-collision is indicated. The MFRC530 uses the RxThreshold register CollLevel[3:0] bits setting to distinguish between a 1-bit or 0-bit and a bit-collision. If the amplitude of the half-bit with smaller amplitude is larger than that defined by the CollLevel[3:0] bits, the MFRC530 flags a bit-collision using the error flag CollErr. If a bit-collision is detected in a parity bit, the ParityErr flag is set.

On a detected collision, the receiver continues receiving the incoming data stream. In the case of a bit-collision, the decoder sends logic 1 at the collision position.

**Remark:** As an exception, if bit ZeroAfterColl is set, all bits received after the first bit-collision are forced to zero, regardless whether a bit-collision or an unequivocal state has been detected. This feature makes it easier for the control software to perform the anti-collision procedure as defined in ISO/IEC 14443 A.

When the first bit collision in a frame is detected, the bit-collision position is stored in the CollPos register.

[Table 138](#) shows the collision positions.

Table 138. Return values for bit-collision positions

| Collision in bit   | CollPos register value (Decimal) |
|--|----------------------------------|
| SOF  | 0                                |
| Least Significant Bit (LSB) of the Least Significant Byte (LSByte) | 1                                |
| ...  | ...                              |
| Most Significant Bit (MSB) of the LSByte                           | 8                                |
| LSB of second byte   | 9                                |
| ...  | ...                              |
| MSB of second byte   | 16                               |
| LSB of third byte  | 17                               |
| ...  | ...                              |

Parity bits are not counted in the CollPos register because bit-collisions in parity bit occur after bit-collisions in the data bits. If a collision is detected in the SOF, a frame error is flagged and no data is sent to the FIFO buffer. In this case, the receiver continues to monitor the incoming signal. It generates the correct notifications to the microprocessor when the end of the faulty input stream is detected. This helps the microprocessor to determine when it is next allowed to send data to the card.

#### 11.2.2.4 Receiving bit oriented frames

The receiver can manage byte streams with incomplete bytes which result in bit-oriented frames. To support this, the following values may be used:

- BitFraming register's RxAlign[2:0] bits select a bit offset for the first incoming byte. For example, if RxAlign[2:0] = 3, the first 5 bits received are forwarded to the FIFO buffer. Further bits are packed into bytes and forwarded. After reception, RxAlign[2:0] is automatically cleared. If RxAlign[2:0] = logic 0, all incoming bits are packed into one byte.
- RxLastBits[2:0] returns the number of bits valid in the last received byte. For example, if RxLastBits[2:0] evaluates to 5 bits at the end of the received command, the 5 least significant bits are valid. If the last byte is complete, RxLastBits[2:0] evaluates to zero.

RxLastBits[2:0] is only valid if a frame error is not indicated by the FramingErr flag. If RxAlign[2:0] is not zero and ParityEn is active, the first parity bit is ignored and not checked.

The first byte containing a single bit from the card is not sent to the microprocessor but suppressed when RxAlign[2:0] is set to 7 (see [Section 10.5.2.4 on page 53](#)).

**Remark:** Collisions detected at CollPos register bit positions 6, 14, 22, 30 and 38 cannot be resolved using RxAlign[2:0]. They must be resolved using the control software.

### 11.2.2.5 Communication errors

The events which can set error flags are shown in [Table 139](#).

**Table 139. Communication error table**

| Cause  | Flag bit   |
|--|------------|
| Received data did not start with the SOF pattern                                     | FramingErr |
| CRC block is not equal to the expected value   | CRCErr     |
| Received data is shorter than the CRC block  | CRCErr     |
| The parity bit is not equal to the expected value (i.e. a bit-collision, not parity) | ParityErr  |
| A bit-collision is detected  | CollErr    |

### 11.2.3 Transceive command 1Eh

**Table 140. Transceive command 1Eh**

| Command    | Value | Action  | Arguments and data | Returned data |
|------------|-------|---|--------------------|---------------|
| Transceive | 1Eh   | transmits data from FIFO buffer to the card and then automatically activates the receiver | data stream        | data stream   |

The Transceive command first executes the Transmit command (see [Section 11.2.1 on page 74](#)) and then starts the Receive command (see [Section 11.2.2 on page 77](#)). All data transmitted is sent using the FIFO buffer and all data received is written to the FIFO buffer. The Transceive command can only be started by the microprocessor.

**Remark:** To adjust the timing relationship between transmitting and receiving, use the RxWait register. This register is used to define the time delay between the last bit transmitted and activation of the receiver. In addition, the BitPhase register determines the phase-shift between the transmitter and receiver clock.

### 11.2.4 Card communication states

The status of the transmitter and receiver state machine can be read from bits ModemState[2:0] in the PrimaryStatus register.

The assignment of ModemState[2:0] to the internal action is shown in [Table 141](#).

**Table 141. Meaning of ModemState**

| ModemState [2:0] | State      | Description   |
|------------------|------------|---|
| 000              | Idle       | transmitter and/or receiver are not operating                         |
| 001              | TxSOF      | transmitting the SOF pattern  |
| 010              | TxData     | transmitting data or redundancy check (CRC) bits from the FIFO buffer |
| 011              | TxEOF      | transmitting the EOF pattern  |
| 100              | GoToRx1    | intermediate state passed, when receiver starts                       |
|                  | GoToRx2    | intermediate state passed, when receiver finishes                     |
| 101              | PrepareRx  | waiting until the RxWait register time period expires                 |
| 110              | AwaitingRx | receiver activated; waiting for an input signal on pin RX             |
| 111              | Receiving  | receiving data  |



11.2.5 Card communication state diagram

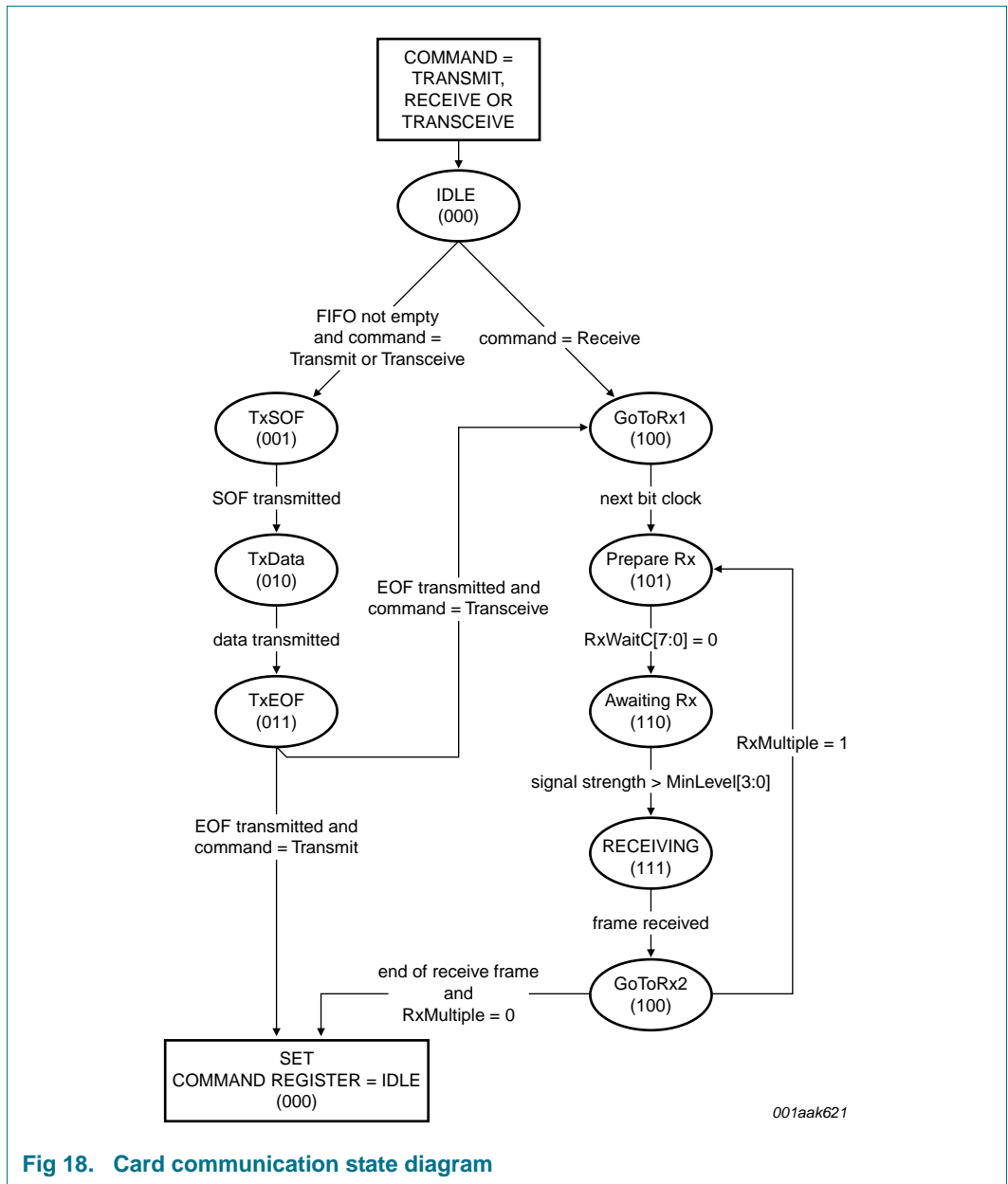


Fig 18. Card communication state diagram

## 11.3 EEPROM commands

### 11.3.1 WriteE2 command 01h

Table 142. WriteE2 command 01h

| Command | Value | Action   | FIFO               |               |
|---------|-------|--|--------------------|---------------|
|         |       |  | Arguments and data | Returned data |
| WriteE2 | 01h   | get data from FIFO buffer and write it to the EEPROM | start address LSB  | -             |
|         |       |  | start address MSB  | -             |
|         |       |  | data byte stream   | -             |

The WriteE2 command interprets the first two bytes in the FIFO buffer as the EEPROM start byte address. Any further bytes are interpreted as data bytes and are programmed into the EEPROM, starting from the given EEPROM start byte address. This command does not return any data.

The WriteE2 command can only be started by the microprocessor. It will not stop automatically but has to be stopped explicitly by the microprocessor by issuing the Idle command.

#### 11.3.1.1 Programming process

One byte up to 16 bytes can be programmed into the EEPROM during a single programming cycle. The time needed is approximately 5.8 ms.

The state machine copies all the prepared data bytes to the FIFO buffer and then to the EEPROM input buffer. The internal EEPROM input buffer is 16 bytes long which is equal to the block size of the EEPROM. A programming cycle is started if the last position of the EEPROM input buffer is written or if the last byte of the FIFO buffer has been read.

The E2Ready flag remains logic 0 when there are unprocessed bytes in the FIFO buffer or the EEPROM programming cycle is still in progress. When all the data from the FIFO buffer are programmed into the EEPROM, the E2Ready flag is set to logic 1. Together with the rising edge of E2Ready the TxIRq interrupt request flag shows logic 1. This can be used to generate an interrupt when programming of all data is finished.

Once E2Ready = logic 1, the WriteE2 command can be stopped by the microprocessor by sending the Idle command.

**Remark:** During the EEPROM programming indicated by E2Ready = logic 0, the WriteE2 command cannot be stopped using any other command.

11.3.1.2 Timing diagram

Figure 19 shows programming five bytes into the EEPROM.

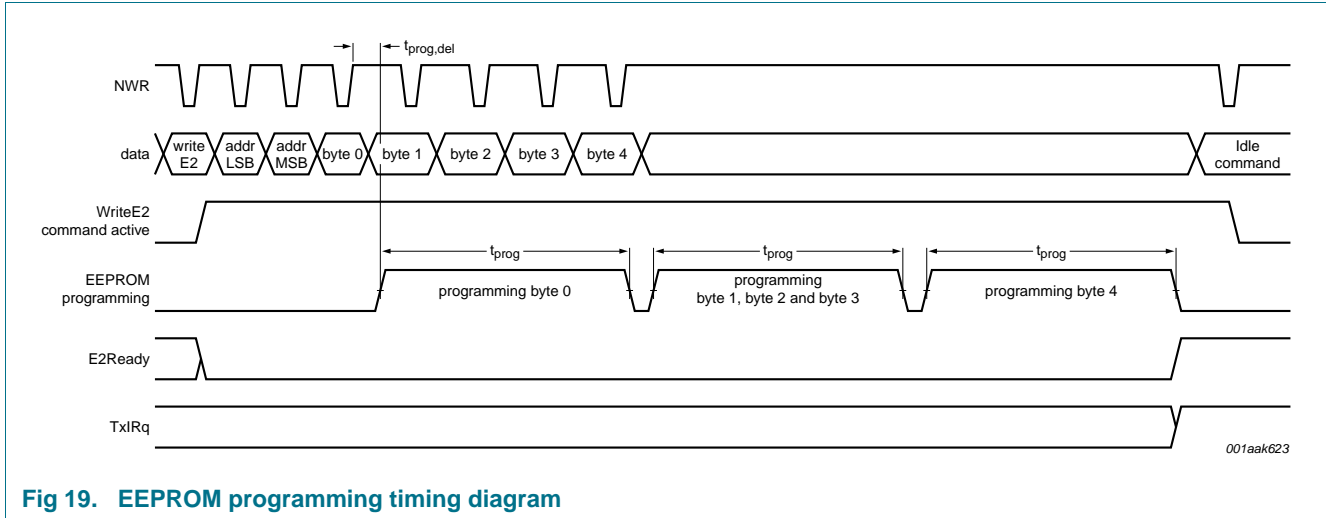


Fig 19. EEPROM programming timing diagram

Assuming that the MFRC530 finds and reads byte 0 before the microprocessor is able to write byte 1 ( $t_{prog,del} = 300\text{ ns}$ ). This causes the MFRC530 to start the programming cycle ( $t_{prog}$ ), which takes approximately 5.8 ms to complete. In the meantime, the microprocessor stores byte 1 to byte 4 in the FIFO buffer.

If the EEPROM start byte address is 16Ch then byte 0 is stored at that address. The MFRC530 copies the subsequent data bytes into the EEPROM input buffer. Whilst copying byte 3, it detects that this data byte has to be programmed at the EEPROM byte address 16Fh. As this is the end of the memory block, the MFRC530 automatically starts a programming cycle.

Next, byte 4 is programmed at the EEPROM byte address 170h. As this is the last data byte, the E2Ready and TxIRq flags are set indicating the end of the EEPROM programming activity.

Although all data has been programmed into the EEPROM, the MFRC530 stays in the WriteE2 command. Writing more data to the FIFO buffer would lead to another EEPROM programming cycle continuing from EEPROM byte address 171h. The command is stopped using the Idle command.

11.3.1.3 WriteE2 command error flags

Programming is restricted for EEPROM block 0 (EEPROM byte address 00h to 0Fh). If you program these addresses, the AccessErr flag is set and a programming cycle is not started.

Addresses above 1FFh are taken modulo 200h; see [Section 9.2 on page 12](#) for the EEPROM memory organization.

### 11.3.2 ReadE2 command 03h

Table 143. ReadE2 command 03h

| Command | Value | Action   | Arguments  | Returned data |
|---------|-------|--|--|---------------|
| ReadE2  | 03h   | reads EEPROM data and stores it in the FIFO buffer | start address LSB<br>start address MSB<br>number of data bytes | data bytes    |

The ReadE2 command interprets the first two bytes stored in the FIFO buffer as the EEPROM starting byte address. The next byte specifies the number of data bytes returned.

When all three argument bytes are available in the FIFO buffer, the specified number of data bytes is copied from the EEPROM into the FIFO buffer, starting from the given EEPROM starting byte address.

The ReadE2 command can only be triggered by the microprocessor and it automatically stops when all data has been copied.

#### 11.3.2.1 ReadE2 command error flags

Reading is restricted to EEPROM blocks 8h to 1Fh (key memory area). Reading from these addresses sets the flag AccessErr = logic 1.

Addresses above 1FFh are taken as modulo 200h; see [Section 9.2 on page 12](#) for the EEPROM memory organization.

## 11.4 Diverse commands

### 11.4.1 LoadConfig command 07h

Table 144. LoadConfig command 07h

| Command    | Value | Action   | Arguments and data                     | Returned data |
|------------|-------|--|--|---------------|
| LoadConfig | 07h   | reads data from EEPROM and initializes the registers | start address LSB<br>start address MSB | -<br>-        |

The LoadConfig command interprets the first two bytes found in the FIFO buffer as the EEPROM starting byte address. When the two argument bytes are available in the FIFO buffer, 32 bytes from the EEPROM are copied into the Control and other relevant registers, starting at the EEPROM starting byte address. The LoadConfig command can only be started by the microprocessor and it automatically stops when all relevant registers have been copied.

#### 11.4.1.1 Register assignment

The 32 bytes of EEPROM content are written to the MFRC530 registers 10h to register 2Fh; see [Section 9.2 on page 12](#) for the EEPROM memory organization.

**Remark:** The procedure for the register assignment is the same as it is for the StartUp initialization (see [Section 9.7.3 on page 27](#)). The difference is, the EEPROM starting byte address for the startup initialization is fixed to 10h (block 1, byte 0). However, it can be chosen with the LoadConfig command.

### 11.4.1.2 Relevant LoadConfig command error flags

Valid EEPROM starting byte addresses are between 10h and 60h.

Copying from block 8h to 1Fh (keys) is restricted. Reading from these addresses sets the flag AccessErr = logic 1.

Addresses above 1FFh are taken as modulo 200h; see [Section 9.2 on page 12](#) for the EEPROM memory organization.

### 11.4.2 CalcCRC command 12h

**Table 145. CalcCRC command 12h**

| Command | Value | Action                        | Arguments and data | Returned data |
|---------|-------|-------------------------------|--------------------|---------------|
| CalcCRC | 12h   | activates the CRC coprocessor | data byte stream   | -             |

The CalcCRC command takes all the data from the FIFO buffer as the input bytes for the CRC coprocessor. All data stored in the FIFO buffer before the command is started is processed.

This command does not return any data to the FIFO buffer but the content of the CRC can be read using the CRCResultLSB and CRCResultMSB registers.

The CalcCRC command can only be started by the microprocessor and it does not automatically stop. It must be stopped by the microprocessor sending the Idle command. If the FIFO buffer is empty, the CalcCRC command waits for further input before proceeding.

#### 11.4.2.1 CRC coprocessor settings

[Table 146](#) shows the parameters that can be configured for the CRC coprocessor.

**Table 146. CRC coprocessor parameters**

| Parameter           | Value                           | Bit          | Register          |
|---------------------|---------------------------------|--------------|-------------------|
| CRC register length | 8-bit or 16-bit CRC             | CRC8         | ChannelRedundancy |
| CRC algorithm       | ISO/IEC 14443 A or ISO/IEC 3309 | CRC3309      | ChannelRedundancy |
| CRC preset value    | any                             | CRCPresetLSB | CRCPresetLSB      |
|                     |                                 | CRCPresetMSB | CRCPresetMSB      |

The CRC polynomial for the 8-bit CRC is fixed to  $x^8 + x^4 + x^3 + x^2 + 1$ .

The CRC polynomial for the 16-bit CRC is fixed to  $x^{16} + x^{12} + x^5 + 1$ .

#### 11.4.2.2 CRC coprocessor status flags

The CRCReady status flag indicates that the CRC coprocessor has finished processing all the data bytes in the FIFO buffer. When the CRCReady flag is set to logic 1, an interrupt is requested which sets the TxIRq flag. This supports interrupt driven use of the CRC coprocessor.

When CRCReady and TxIRq flags are set to logic 1 the content of the CRCResultLSB and CRCResultMSB registers and the CRCErr flag are valid. The CRCResultLSB and CRCResultMSB registers hold the content of the CRC, the CRCErr flag indicates CRC validity for the processed data.

## 11.5 Error handling during command execution

If an error is detected during command execution, the PrimaryStatus register Err flag is set. The microprocessor can evaluate the status flags in the ErrorFlag register to get information about the cause of the error.

**Table 147. ErrorFlag register error flags overview**

| Error flag | Related commands             |
|------------|------------------------------|
| KeyErr     | LoadKeyE2, LoadKey           |
| AccessErr  | WriteE2, ReadE2, LoadConfig  |
| FIFOvlf    | no specific commands         |
| CRCErr     | Receive, Transceive, CalcCRC |
| FramingErr | Receive, Transceive          |
| ParityErr  | Receive, Transceive          |
| CollErr    | Receive, Transceive          |

## 11.6 MIFARE security commands

### 11.6.1 LoadKeyE2 command 0Bh

**Table 148. LoadKeyE2 command 0Bh**

| Command   | Value | Action   | Arguments and data                     | Returned data |
|-----------|-------|--|--|---------------|
| LoadKeyE2 | 0Bh   | reads a key from the EEPROM and puts it into the internal key buffer | start address LSB<br>start address MSB | -<br>-        |

The LoadKeyE2 command interprets the first two bytes found in the FIFO buffer as the EEPROM starting byte address. The EEPROM bytes starting from the given starting byte address are interpreted as the key when stored in the correct key format as described in [Section 9.2.3.1 “Key format” on page 15](#). When both argument bytes are available in the FIFO buffer, the command executes.

The LoadKeyE2 command can only be started by the microprocessor and it automatically stops after copying the key from the EEPROM to the key buffer.

#### 11.6.1.1 Relevant LoadKeyE2 command error flags

If the key format is incorrect (see [Section 9.2.3.1 “Key format” on page 15](#)) an undefined value is copied into the key buffer and the KeyErr flag is set.

### 11.6.2 LoadKey command 19h

**Table 149. LoadKey command 19h**

| Command | Value | Action   | Arguments and data  | Returned data         |
|---------|-------|--|---|-----------------------|
| LoadKey | 19h   | reads a key from the FIFO buffer and puts it into the key buffer | byte 0 (LSB)<br>byte 1<br>...<br>byte 10<br>byte 11 (MSB) | -<br>-<br>-<br>-<br>- |

The LoadKey command interprets the first twelve bytes it finds in the FIFO buffer as the key when stored in the correct key format as described in [Section 9.2.3.1 “Key format” on page 15](#). When the twelve argument bytes are available in the FIFO buffer they are checked and, if valid, are copied into the key buffer.

The LoadKey command can only be started by the microprocessor and it automatically stops after copying the key from the FIFO buffer to the key buffer.

#### 11.6.2.1 Relevant LoadKey command error flags

All bytes requested are copied from the FIFO buffer to the key buffer. If the key format is not correct (see [Section 9.2.3.1 “Key format” on page 15](#)) an undefined value is copied into the key buffer and the KeyErr flag is set.

#### 11.6.3 Authent1 command 0Ch

**Table 150. Authent1 command 0Ch**

| Command  | Value | Action   | Arguments and data  | Returned data              |
|----------|-------|--|---|----------------------------|
| Authent1 | 0Ch   | performs the first part of the Crypto1 card authentication | card Authent1 command<br>card block address<br>card serial number LSB<br>card serial number byte1<br>card serial number byte2<br>card serial number MSB | -<br>-<br>-<br>-<br>-<br>- |

The Authent1 command is a special Transceive command; it sends six argument bytes to the card. The card's response is not sent to the microprocessor, it is used instead to authenticate the card to the MFRC530 and vice versa.

The Authent1 command can be triggered only by the microprocessor. The sequence of states for this command are the same as those for the Transceive command; see [Section 11.2.3 on page 80](#).

#### 11.6.4 Authent2 command 14h

**Table 151. Authent2 command 14h**

| Command  | Value | Action  | Arguments and data | Returned data |
|----------|-------|---|--------------------|---------------|
| Authent2 | 14h   | performs the second part of the card authentication using the Crypto1 algorithm | -                  | -             |

The Authent2 command is a special Transceive command. It does not need an argument byte, however all the data needed to be sent to the card is assembled by the MFRC530. The card response is not sent to the microprocessor but is used to authenticate the card to the MFRC530 and vice versa.

The Authent2 command can only be started by the microprocessor. The sequence of states for this command are the same as those for the Transceive command; see [Section 11.2.3 on page 80](#).

### 11.6.4.1 Authent2 command effects

If the Authent2 command is successful, the authenticity of card and the MFRC530 are proved. This automatically sets the Crypto1On control bit. When bit Crypto1On = logic 1, all further card communication is encrypted using the Crypto1 security algorithm. If the Authent2 command fails, bit Crypto1On is cleared (Crypto1On = logic 0).

**Remark:** The Crypto1On flag can only be set by a successfully executed Authent2 command and not by the microprocessor. The microprocessor can clear bit Crypto1On to continue with unencrypted (plain) card communication.

**Remark:** The Authent2 command must be executed immediately after a successful Authent1 command; see [Section 11.6.3 “Authent1 command 0Ch”](#). In addition, the keys stored in the key buffer and those on the card must match.

## 12. Limiting values

**Table 152. Limiting values**

In accordance with the Absolute Maximum Rating System (IEC 60134).

| Symbol                | Parameter                      | Conditions                 | Min  | Max                    | Unit |
|-----------------------|--------------------------------|----------------------------|------|------------------------|------|
| T <sub>amb</sub>      | ambient temperature            |                            | -40  | +150                   | °C   |
| T <sub>stg</sub>      | storage temperature            |                            | -40  | +150                   | °C   |
| V <sub>DDD</sub>      | digital supply voltage         |                            | -0.5 | +6                     | V    |
| V <sub>DDA</sub>      | analog supply voltage          |                            | -0.5 | +6                     | V    |
| V <sub>DD(TVDD)</sub> | TVDD supply voltage            |                            | -0.5 | +6                     | V    |
| V <sub>i</sub>        | input voltage (absolute value) | on any digital pin to DVSS | -0.5 | V <sub>DDD</sub> + 0.5 | V    |
|                       |                                | on pin RX to AVSS          | -0.5 | V <sub>DDA</sub> + 0.5 | V    |

## 13. Characteristics

### 13.1 Operating condition range

**Table 153. Operating condition range**

| Symbol                | Parameter                       | Conditions                                | Min | Typ | Max  | Unit |
|-----------------------|---------------------------------|---|-----|-----|------|------|
| T <sub>amb</sub>      | ambient temperature             | -   | -25 | +25 | +85  | °C   |
| V <sub>DDD</sub>      | digital supply voltage          | DVSS = AVSS = TVSS = 0 V                  | 4.5 | 5.0 | 5.5  | V    |
| V <sub>DDA</sub>      | analog supply voltage           | DVSS = AVSS = TVSS = 0 V                  | 4.5 | 5.0 | 5.5  | V    |
| V <sub>DD(TVDD)</sub> | TVDD supply voltage             | DVSS = AVSS = TVSS = 0 V                  | 3.0 | 5.0 | 5.5  | V    |
| V <sub>ESD</sub>      | electrostatic discharge voltage | Human Body Model (HBM); 1.5 kΩ,<br>100 pF | -   | -   | 1000 | V    |
|                       |                                 | Machine Model (MM); 0.75 μH,<br>200 pF    | -   | -   | 100  | V    |



## 13.2 Current consumption

Table 154. Current consumption

| Symbol                | Parameter              | Conditions   | Min | Typ | Max  | Unit |
|-----------------------|------------------------|--|-----|-----|------|------|
| I <sub>DDD</sub>      | digital supply current | Idle command   | -   | 8   | 11   | mA   |
|                       |                        | Standby mode   | -   | 3   | 5    | mA   |
|                       |                        | Soft power-down mode   | -   | 800 | 1000 | μA   |
|                       |                        | Hard power-down mode   | -   | 1   | 10   | μA   |
| I <sub>DDA</sub>      | analog supply current  | Idle command; receiver on                                      | -   | 25  | 40   | mA   |
|                       |                        | Idle command; receiver off                                     | -   | 12  | 15   | mA   |
|                       |                        | Standby mode   | -   | 10  | 13   | mA   |
|                       |                        | Soft power-down mode   | -   | 1   | 10   | μA   |
|                       |                        | Hard power-down mode   | -   | 1   | 10   | μA   |
| I <sub>DD(TVDD)</sub> | TVDD supply current    | continuous wave  | -   | -   | 150  | mA   |
|                       |                        | pins TX1 and TX2 unconnected;<br>TX1RFEn and TX2RFEn = logic 1 | -   | 5.5 | 7    | mA   |
|                       |                        | pins TX1 and TX2 unconnected;<br>TX1RFEn and TX2RFEn = logic 0 | -   | 65  | 130  | μA   |

## 13.3 Pin characteristics

### 13.3.1 Input pin characteristics

Pins D0 to D7, A0 and A1 have TTL input characteristics and behave as defined in [Table 155](#).

Table 155. Standard input pin characteristics

| Symbol          | Parameter             | Conditions                     | Min                  | Typ | Max                  | Unit |
|-----------------|-----------------------|--------------------------------|----------------------|-----|----------------------|------|
| I <sub>LI</sub> | input leakage current |                                | -1.0                 | -   | +1.0                 | μA   |
| V <sub>th</sub> | threshold voltage     | CMOS: V <sub>DDD</sub> < 3.6 V | 0.35V <sub>DDD</sub> | -   | 0.65V <sub>DDD</sub> | V    |
|                 |                       | TTL: 4.5 < V <sub>DDD</sub>    | 0.8                  | -   | 2.0                  | V    |

The digital input pins NCS, NWR, NRD, ALE, A2 and MFIN have Schmitt trigger characteristics, and behave as defined in [Table 156](#).

Table 156. Schmitt trigger input pin characteristics

| Symbol          | Parameter             | Conditions                      | Min                  | Typ | Max                  | Unit |
|-----------------|-----------------------|---------------------------------|----------------------|-----|----------------------|------|
| I <sub>LI</sub> | input leakage current |                                 | -1.0                 | -   | +1.0                 | μA   |
| V <sub>th</sub> | threshold voltage     | positive-going threshold;       | 1.4                  | -   | 2.0                  | V    |
|                 |                       | TTL = 4.5 < V <sub>DDD</sub>    |                      |     |                      |      |
|                 |                       | CMOS = V <sub>DDD</sub> < 3.6 V | 0.65V <sub>DDD</sub> | -   | 0.75V <sub>DDD</sub> | V    |
|                 |                       | negative-going threshold;       | 0.8                  | -   | 1.3                  | V    |
|                 |                       | TTL = 4.5 < V <sub>DDD</sub>    |                      |     |                      |      |
|                 |                       | CMOS = V <sub>DDD</sub> < 3.6 V | 0.25V <sub>DDD</sub> | -   | 0.4V <sub>DDD</sub>  | V    |

Pin RSTPD has Schmitt trigger CMOS characteristics. In addition, it is internally filtered by a RC low-pass filter which causes a propagation delay on the reset signal.

**Table 157. RSTPD input pin characteristics**

| Symbol   | Parameter             | Conditions   | Min          | Typ | Max          | Unit          |
|----------|-----------------------|--|--------------|-----|--------------|---------------|
| $I_{LI}$ | input leakage current |  | -1.0         | -   | +1.0         | $\mu\text{A}$ |
| $V_{th}$ | threshold voltage     | positive-going threshold;<br>CMOS = $V_{DD}$ < 3.6 V | $0.65V_{DD}$ | -   | $0.75V_{DD}$ | V             |
|          |                       | negative-going threshold;<br>CMOS = $V_{DD}$ < 3.6 V | $0.25V_{DD}$ | -   | $0.4V_{DD}$  | V             |
| $t_{PD}$ | propagation delay     |  | -            | -   | 20           | $\mu\text{s}$ |

The analog input pin RX has the input capacitance and input voltage range shown in [Table 158](#).

**Table 158. RX input capacitance and input voltage range**

| Symbol       | Parameter             | Conditions  | Min | Typ | Max | Unit |
|--------------|-----------------------|---|-----|-----|-----|------|
| $C_i$        | input capacitance     |   | -   | -   | 15  | pF   |
| $V_{i(dyn)}$ | dynamic input voltage | $V_{DDA} = 5\text{ V}$ ; $T_{amb} = 25\text{ }^\circ\text{C}$ | 1.1 | -   | 4.4 | V    |

### 13.3.2 Digital output pin characteristics

Pins D0 to D7, MFOUT and IRQ have CMOS output characteristics and behave as defined in [Table 159](#).

**Table 159. Digital output pin characteristics**

| Symbol   | Parameter                 | Conditions                                       | Min | Typ | Max | Unit |
|----------|---------------------------|--|-----|-----|-----|------|
| $V_{OH}$ | HIGH-level output voltage | $V_{DD} = 5\text{ V}$ ; $I_{OH} = -1\text{ mA}$  | 2.4 | 4.9 | -   | V    |
|          |                           | $V_{DD} = 5\text{ V}$ ; $I_{OH} = -10\text{ mA}$ | 2.4 | 4.2 | -   | V    |
| $V_{OL}$ | LOW-level output voltage  | $V_{DD} = 5\text{ V}$ ; $I_{OL} = 1\text{ mA}$   | -   | 25  | 400 | mV   |
|          |                           | $V_{DD} = 5\text{ V}$ ; $I_{OL} = 10\text{ mA}$  | -   | 250 | 400 | mV   |
| $I_O$    | output current            | source or sink; $V_{DD} = 5\text{ V}$            | -   | -   | 10  | mA   |

**Remark:** Pin IRQ can be configured as open collector which causes the  $V_{OH}$  values to be no longer applicable.

### 13.3.3 Antenna driver output pin characteristics

The source conductance of the antenna driver pins TX1 and TX2 for driving the HIGH-level can be configured using the CwConductance register's GsCfgCW[5:0] bits, while their source conductance for driving the LOW-level is constant.

The antenna driver default configuration output characteristics are specified in [Table 160](#).

**Table 160. Antenna driver output pin characteristics**

| Symbol          | Parameter                 | Conditions  | Min | Typ  | Max | Unit |
|-----------------|---------------------------|---|-----|------|-----|------|
| V <sub>OH</sub> | HIGH-level output voltage | V <sub>DD(TVDD)</sub> = 5.0 V; I <sub>OL</sub> = 20 mA  | -   | 4.97 | -   | V    |
|                 |                           | V <sub>DD(TVDD)</sub> = 5.0 V; I <sub>OL</sub> = 100 mA | -   | 4.85 | -   | V    |
| V <sub>OL</sub> | LOW-level output voltage  | V <sub>DD(TVDD)</sub> = 5.0 V; I <sub>OL</sub> = 20 mA  | -   | 30   | -   | mV   |
|                 |                           | V <sub>DD(TVDD)</sub> = 5.0 V; I <sub>OL</sub> = 100 mA | -   | 150  | -   | mV   |
| I <sub>O</sub>  | output current            | transmitter; continuous wave; peak-to-peak              | -   | -    | 200 | mA   |

## 13.4 AC electrical characteristics

### 13.4.1 Separate read/write strobe bus timing

**Table 161. Timing specification for separate read/write strobe**

| Symbol              | Parameter                                   | Conditions                      | Min | Typ | Max | Unit |
|---------------------|---|---------------------------------|-----|-----|-----|------|
| t <sub>LHLL</sub>   | ALE HIGH time                               |                                 | 20  | -   | -   | ns   |
| t <sub>AVLL</sub>   | address valid to ALE LOW time               |                                 | 15  | -   | -   | ns   |
| t <sub>LLAX</sub>   | address hold after ALE LOW time             |                                 | 8   | -   | -   | ns   |
| t <sub>LLRWL</sub>  | ALE LOW to read/write LOW time              | ALE LOW to NRD or NWR LOW       | 15  | -   | -   | ns   |
| t <sub>SLRWL</sub>  | chip select LOW to read/write LOW time      | NCS LOW to NRD or NWR LOW       | 0   | -   | -   | ns   |
| t <sub>RWHS</sub>   | read/write HIGH to chip select HIGH time    | NRD or NWR HIGH to NCS HIGH     | 0   | -   | -   | ns   |
| t <sub>RLDV</sub>   | read LOW to data input valid time           | NRD LOW to data valid           | -   | -   | 65  | ns   |
| t <sub>RHDZ</sub>   | read HIGH to data input high impedance time | NRD HIGH to data high-impedance | -   | -   | 20  | ns   |
| t <sub>WLQV</sub>   | write LOW to data output valid time         | NWR LOW to data valid           | -   | -   | 35  | ns   |
| t <sub>WHDX</sub>   | data output hold after write HIGH time      | data hold time after NWR HIGH   | 8   | -   | -   | ns   |
| t <sub>RWLRWH</sub> | read/write LOW time                         | NRD or NWR                      | 65  | -   | -   | ns   |
| t <sub>AVRWL</sub>  | address valid to read/write LOW time        | NRD or NWR LOW (set-up time)    | 30  | -   | -   | ns   |
| t <sub>WHAX</sub>   | address hold after write HIGH time          | NWR HIGH (hold time)            | 8   | -   | -   | ns   |
| t <sub>RWHRWL</sub> | read/write HIGH time                        |                                 | 150 | -   | -   | ns   |

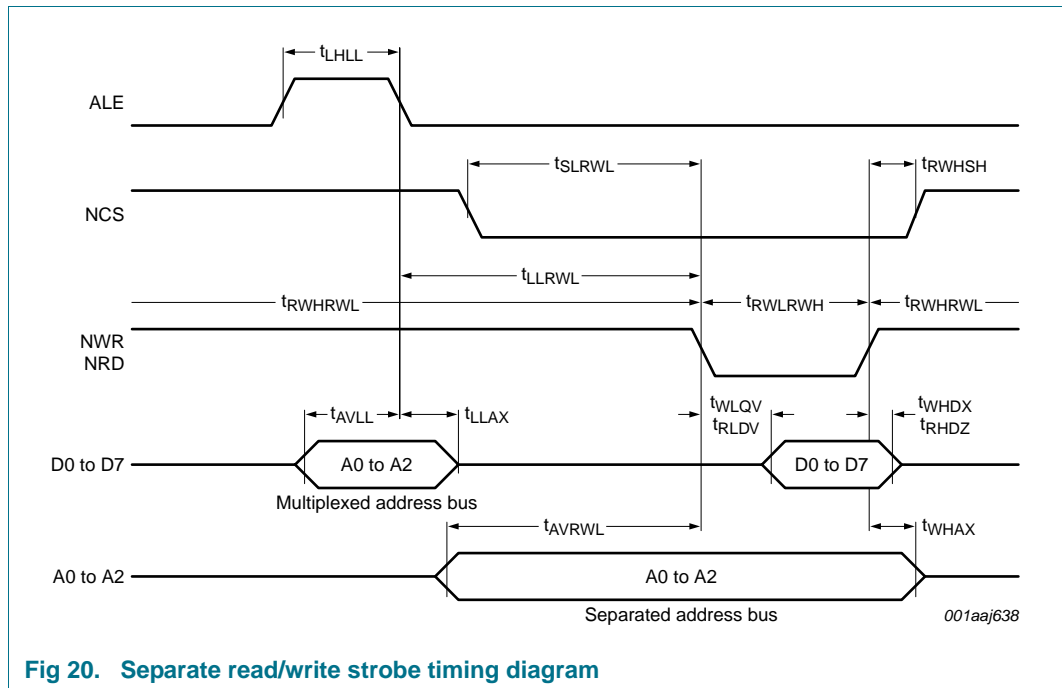


Fig 20. Separate read/write strobe timing diagram

**Remark:** The signal ALE is not relevant for separate address/data bus and the multiplexed addresses on the data bus do not care. The multiplexed address and data bus address lines (A0 to A2) must be connected as described in [Section 9.1.3 on page 8](#).

### 13.4.2 Common read/write strobe bus timing

Table 162. Common read/write strobe timing specification

| Symbol              | Parameter  | Conditions                       | Min | Typ | Max | Unit |
|---------------------|--|----------------------------------|-----|-----|-----|------|
| t <sub>LHLL</sub>   | ALE HIGH time                                      |                                  | 20  | -   | -   | ns   |
| t <sub>AVLL</sub>   | address valid to ALE LOW time                      |                                  | 15  | -   | -   | ns   |
| t <sub>LLAX</sub>   | address hold after ALE LOW time                    |                                  | 8   | -   | -   | ns   |
| t <sub>LLDSL</sub>  | ALE LOW to data strobe LOW time                    | NWR or NRD LOW                   | 15  | -   | -   | ns   |
| t <sub>SLDSL</sub>  | chip select LOW to data strobe LOW time            | NCS LOW to NDS LOW               | 0   | -   | -   | ns   |
| t <sub>DSHSH</sub>  | data strobe HIGH to chip select HIGH time          |                                  | 0   | -   | -   | ns   |
| t <sub>DSLQV</sub>  | data strobe LOW to data input valid time           |                                  | -   | -   | 65  | ns   |
| t <sub>DSHDZ</sub>  | data strobe HIGH to data input high impedance time |                                  | -   | -   | 20  | ns   |
| t <sub>DSLQV</sub>  | data strobe LOW to data output valid time          | NDS/NCS LOW                      | -   | -   | 35  | ns   |
| t <sub>DSHQX</sub>  | data output hold after data strobe HIGH time       | NDS HIGH (write cycle hold time) | 8   | -   | -   | ns   |
| t <sub>DSHRWX</sub> | RW hold after data strobe HIGH time                | after NDS HIGH                   | 8   | -   | -   | ns   |
| t <sub>DSLDSH</sub> | data strobe LOW time                               | NDS/NCS                          | 65  | -   | -   | ns   |

Table 162. Common read/write strobe timing specification ...continued

| Symbol       | Parameter                             | Conditions                     | Min | Typ | Max | Unit |
|--------------|---------------------------------------|--------------------------------|-----|-----|-----|------|
| $t_{AVDSL}$  | address valid to data strobe LOW time |                                | 30  | -   | -   | ns   |
| $t_{RHAX}$   | address hold after read HIGH time     |                                | 8   | -   | -   | ns   |
| $t_{DSHDSL}$ | data strobe HIGH time                 | period between write sequences | 150 | -   | -   | ns   |
| $t_{WLDSL}$  | write LOW to data strobe LOW time     | R/NW valid to NDS LOW          | 8   | -   | -   | ns   |

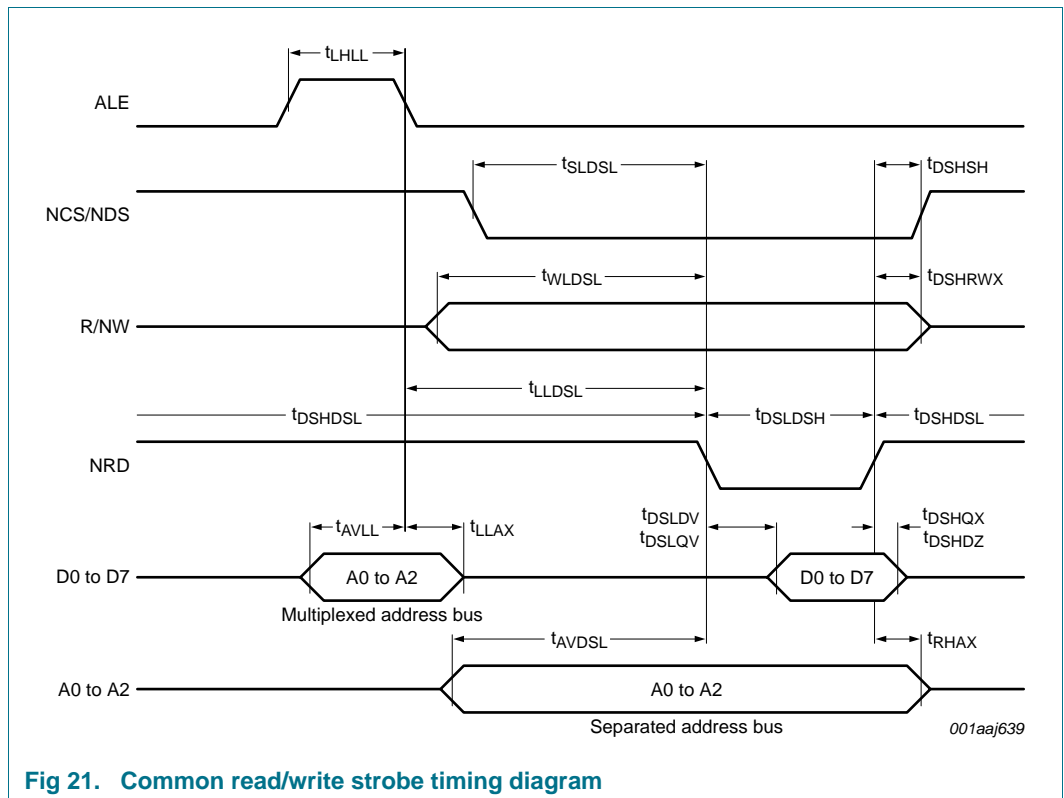


Fig 21. Common read/write strobe timing diagram

13.4.3 EPP bus timing

Table 163. Common read/write strobe timing specification for EPP

| Symbol       | Parameter                                    | Conditions                          | Min | Typ | Max | Unit |
|--------------|--|-------------------------------------|-----|-----|-----|------|
| $t_{ASLASH}$ | address strobe LOW time                      | nAStsb                              | 20  | -   | -   | ns   |
| $t_{AVASH}$  | address valid to address strobe HIGH time    | multiplexed address bus set-up time | 15  | -   | -   | ns   |
| $t_{ASHAV}$  | address valid after address strobe HIGH time | multiplexed address bus hold time   | 8   | -   | -   | ns   |
| $t_{SLDSL}$  | chip select LOW to data strobe LOW time      | NCS LOW to nDStsb LOW               | 0   | -   | -   | ns   |
| $t_{DSHSH}$  | data strobe HIGH to chip select HIGH time    | nDStsb HIGH to NCS HIGH             | 0   | -   | -   | ns   |
| $t_{DSLQV}$  | data strobe LOW to data input valid time     | read cycle                          | -   | -   | 65  | ns   |

Table 163. Common read/write strobe timing specification for EPP ...continued

| Symbol          | Parameter  | Conditions                 | Min | Typ | Max | Unit |
|-----------------|--|----------------------------|-----|-----|-----|------|
| $t_{DSHDZ}$     | data strobe HIGH to data input high impedance time | read cycle                 | -   | -   | 20  | ns   |
| $t_{DSLQV}$     | data strobe LOW to data output valid time          | nDStrb LOW                 | -   | -   | 35  | ns   |
| $t_{DSHQX}$     | data output hold after data strobe HIGH time       | NCS HIGH                   | 8   | -   | -   | ns   |
| $t_{DSHWX}$     | write hold after data strobe HIGH time             | nWrite                     | 8   | -   | -   | ns   |
| $t_{DSLDSH}$    | data strobe LOW time                               | nDStrb                     | 65  | -   | -   | ns   |
| $t_{WLDSL}$     | write LOW to data strobe LOW time                  | nWrite valid to nDStrb LOW | 8   | -   | -   | ns   |
| $t_{DSL-WAITH}$ | data strobe LOW to WAIT HIGH time                  | nDStrb LOW to nWrite HIGH  | -   | -   | 75  | ns   |
| $t_{DSH-WAITL}$ | data strobe HIGH to WAIT LOW time                  | nDStrb HIGH to nWrite LOW  | -   | -   | 75  | ns   |

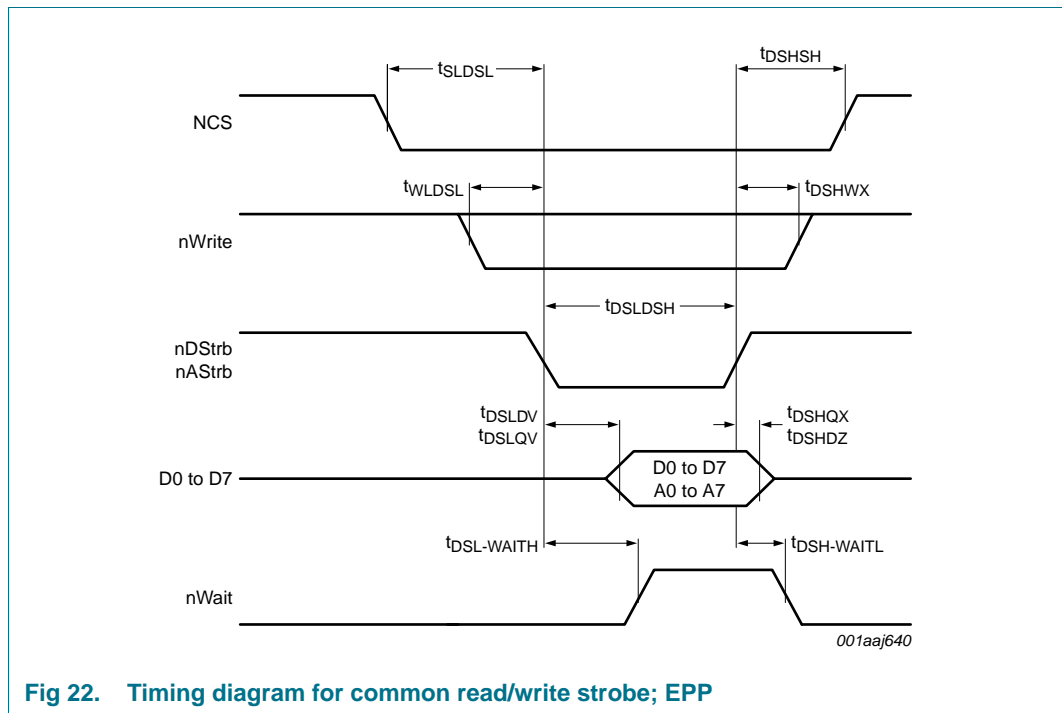


Fig 22. Timing diagram for common read/write strobe; EPP

**Remark:** [Figure 22](#) does not distinguish between the address write cycle and a data write cycle. The timings for the address write and data write cycle are different. In EPP mode, the address lines (A0 to A2) must be connected as described in [Section 9.1.3 on page 8](#).

13.4.4 SPI timing

Table 164. SPI timing specification

| Symbol          | Parameter                                     | Conditions | Min | Typ | Max | Unit |
|-----------------|---|------------|-----|-----|-----|------|
| $t_{SCKL}$      | SCK LOW time                                  |            | 100 | -   | -   | ns   |
| $t_{SCKH}$      | SCK HIGH time                                 |            | 100 | -   | -   | ns   |
| $t_{DSHQX}$     | data output hold after data strobe HIGH time  |            | 20  | -   | -   | ns   |
| $t_{DQXCH}$     | data input/output changing to clock HIGH time |            | 20  | -   | -   | ns   |
| $t_{h(SCKL-Q)}$ | SCK LOW to data output hold time              |            | -   | -   | 15  | ns   |
| $t_{CLSH}$      | clock LOW to chip select HIGH time            |            | 20  | -   | -   | ns   |

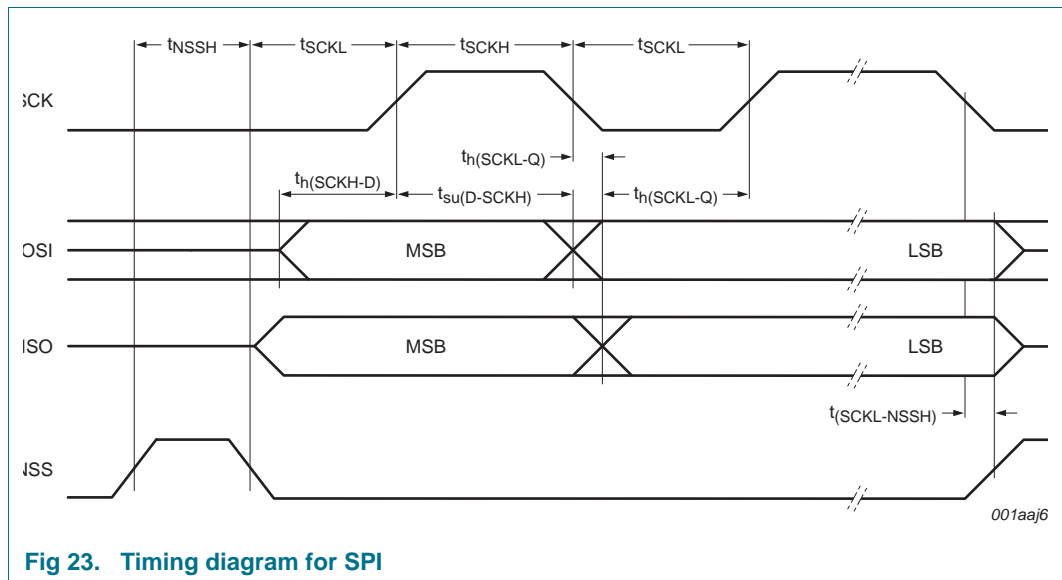


Fig 23. Timing diagram for SPI

**Remark:** To send more bytes in one data stream the NSS signal must be LOW during the send process. To send more than one data stream the NSS signal must be HIGH between each data stream.

13.4.5 Clock frequency

The clock input is pin OSCIN.

Table 165. Clock frequency

| Symbol         | Parameter        | Conditions                  | Min | Typ   | Max | Unit |
|----------------|------------------|-----------------------------|-----|-------|-----|------|
| $f_{clk}$      | clock frequency  | checked by the clock filter | -   | 13.56 | -   | MHz  |
| $\delta_{clk}$ | clock duty cycle |                             | 40  | 50    | 60  | %    |
| $t_{jit}$      | jitter time      | of clock edges              | -   | -     | 10  | ps   |

The clock applied to the MFRC530 acts as a time constant for the synchronous system’s encoder and decoder. The stability of the clock frequency is an important factor for ensuring proper performance. To obtain highest performance, clock jitter must be as small as possible. This is best achieved using the internal oscillator buffer and the

recommended circuitry; see [Section 9.8 on page 27](#).

## 14. EEPROM characteristics

The EEPROM size is  $32 \times 16 \times 8 = 4096$  bit.

**Table 166. EEPROM characteristics**

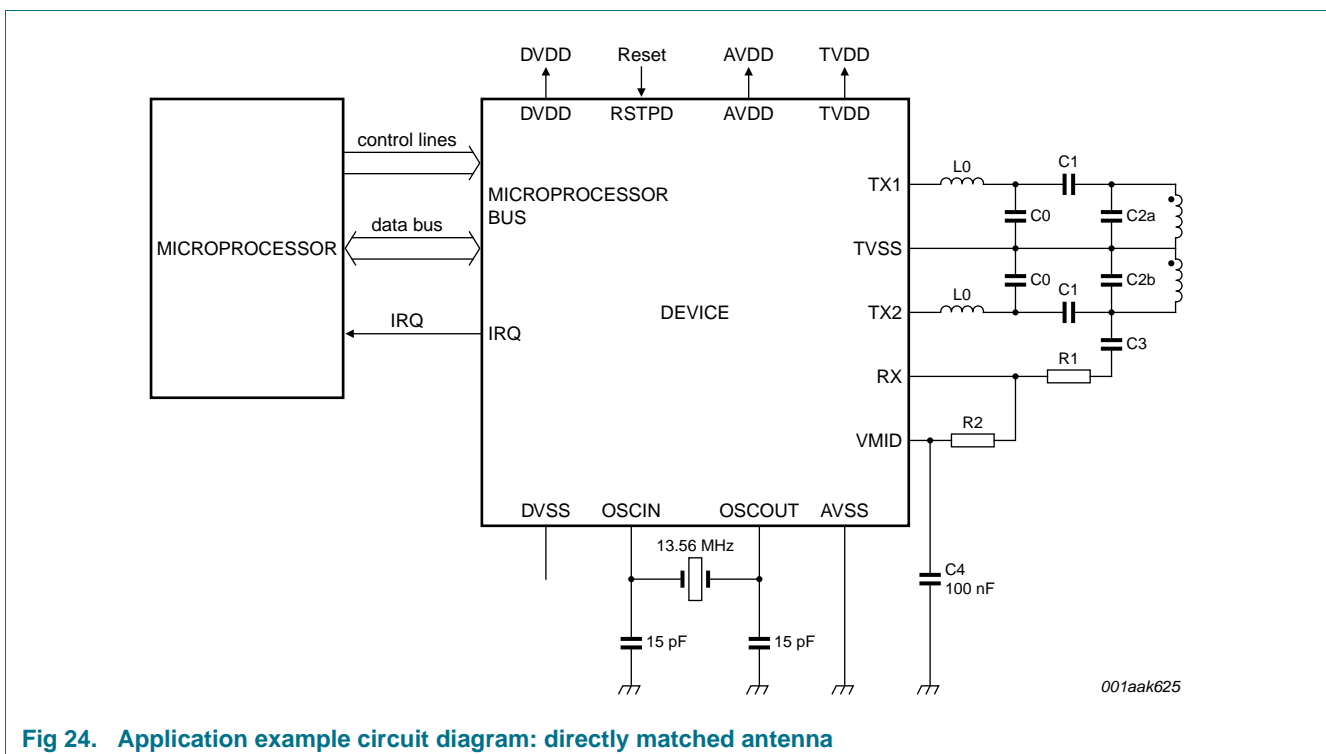
| Symbol                   | Parameter                | Conditions                                      | Min    | Typ | Max | Unit |
|--------------------------|--------------------------|---|--------|-----|-----|------|
| $N_{\text{endu(W\_ER)}}$ | write or erase endurance | erase/write cycles                              | 100000 | -   | -   | Hz   |
| $t_{\text{ret}}$         | retention time           | $T_{\text{amb}} \leq 55 \text{ }^\circ\text{C}$ | 10     | -   | -   | year |
| $t_{\text{er}}$          | erase time               |   | -      | -   | 2.9 | ms   |
| $t_{\text{a(W)}}$        | write access time        |   | -      | -   | 2.9 | ms   |

## 15. Application information

### 15.1 Typical application

#### 15.1.1 Circuit diagram

[Figure 24](#) shows a typical application where the antenna is directly matched to the MFRC530



**Fig 24. Application example circuit diagram: directly matched antenna**

#### 15.1.2 Circuit description

The matching circuit consists of an EMC low-pass filter (L0 and C0), matching circuitry (C1 and C2), a receiver circuit (R1, R2, C3 and C4) and the antenna itself.



Refer to the following application notes for more detailed information about designing and tuning an antenna.

- *MICORE reader IC family; Directly Matched Antenna Design* [Ref. 1](#)
- *MIFARE (14443 A) 13.56 MHz RFID Proximity Antennas* [Ref. 2](#).

#### 15.1.2.1 EMC low-pass filter

The MIFARE system operates at a frequency of 13.56 MHz. This frequency is generated by a quartz oscillator to clock the MFRC530. It is also the basis for driving the antenna using the 13.56 MHz energy carrier. This not only causes power emissions at 13.56 MHz, it also emits power at higher harmonics. International EMC regulations define the amplitude of the emitted power over a broad frequency range. To meet these regulations, appropriate filtering of the output signal is required.

A multilayer board is recommended to implement a low-pass filter as shown in [Figure 24](#). The low-pass filter consists of the components L0 and C0. The recommended values are given in Application notes *MICORE reader IC family; Directly Matched Antenna Design* [Ref. 1](#) and *MIFARE (14443 A) 13.56 MHz RFID Proximity Antennas* [Ref. 2](#).

**Remark:** To achieve best performance, all components must be at least equal in quality to those recommended.

**Remark:** The layout has a major influence on the overall performance of the filter.

#### 15.1.2.2 Antenna matching

Due to the impedance transformation of the low-pass filter, the antenna coil has to be matched to a given impedance. The matching elements C1 and C2 can be estimated and have to be fine tuned depending on the design of the antenna coil.

The correct impedance matching is important to ensure optimum performance. The overall quality factor has to be considered to guarantee a proper ISO/IEC 14443 A communication scheme. Environmental influences have to be considered and common EMC design rules.

Refer to Application notes *MICORE reader IC family; Directly Matched Antenna Design* [Ref. 1](#) and *MIFARE (14443 A) 13.56 MHz RFID Proximity Antennas* [Ref. 2](#) for details.

**Remark:** Do not exceed the current limits ( $I_{DD(TVDD)}$ ), otherwise the chip might be destroyed.

**Remark:** The overall 13.56 MHz RFID proximity antenna design in combination with the MFRC530 IC does not require any specialist RF knowledge. However, all relevant parameters have to be considered to guarantee optimum performance and international EMC compliance.

#### 15.1.2.3 Receiver circuit

The internal receiver of the MFRC530 makes use of both subcarrier load modulation side-bands. No external filtering is required.

It is recommended to use the internally generated VMID potential as the input potential for pin RX. This VMID DC voltage level has to be coupled to pin RX using resistor (R2). To provide a stable DC reference voltage, a capacitor (C4) must be connected between VMID and ground.

The AC voltage divider of R1 + C3 and R2 has to be designed taking in to account the AC voltage limits on pin RX. Depending on the antenna coil design and the impedance, matching the voltage at the antenna coil will differ. Therefore the recommended way to design the receiver circuit is to use the given values for R1, R2, and C3; refer to Application note; *MIFARE (14443 A) 13.56 MHz RFID Proximity Antennas* [Ref. 2](#). The voltage on pin RX can be altered by varying R1 within the given limits.

**Remark:** R2 is AC connected to ground using C4.

#### 15.1.2.4 Antenna coil

The precise calculation of the antenna coil's inductance is not practicable but the inductance can be estimated using [Equation 10](#). We recommend designing an antenna that is either circular or rectangular.

$$L_1[nH] = 2 \cdot I_1[cm] \cdot \left( \ln \left\langle \frac{I_1}{D_1} \right\rangle - K \right) N_1^{1.8} \quad (10)$$

- $I_1$  = length of one turn of the conductor loop
- $D_1$  = diameter of the wire or width of the PCB conductor, respectively
- $K$  = antenna shape factor ( $K = 1.07$  for circular antennas and  $K = 1.47$  for square antennas)
- $N_1$  = number of turns
- $\ln$  = natural logarithm function

The values of the antenna inductance, resistance, and capacitance at 13.56 MHz depend on various parameters such as:

- antenna construction (type of PCB)
- thickness of conductor
- distance between the windings
- shielding layer
- metal or ferrite in the near environment

Therefore a measurement of these parameters under real life conditions or at least a rough measurement and a tuning procedure is highly recommended to guarantee adequate performance. Refer to Application notes *MICORE reader IC family; Directly Matched Antenna Design* [Ref. 1](#) and *MIFARE (14443 A) 13.56 MHz RFID Proximity Antennas* [Ref. 2](#) for details.

## 15.2 Test signals

The MFRC530 allows different kinds of signal measurements. These measurements can be used to check the internally generated and received signals using the serial signal switch as described in [Section 9.11 on page 34](#).

In addition, the MFRC530 enables users to select between:

- internal analog signals for measurement on pin AUX
- internal digital signals for observation on pin MFOUT (based on register selections)

These measurements can be helpful during the design-in phase to optimize the receiver's behavior, or for test purposes.

### 15.2.1 Measurements using the serial signal switch

Using the serial signal switch on pin MFOUT, data is observed that is sent to the card or received from the card. [Table 167](#) gives an overview of the different signals available.

**Table 167. Signal routed to pin MFOUT**

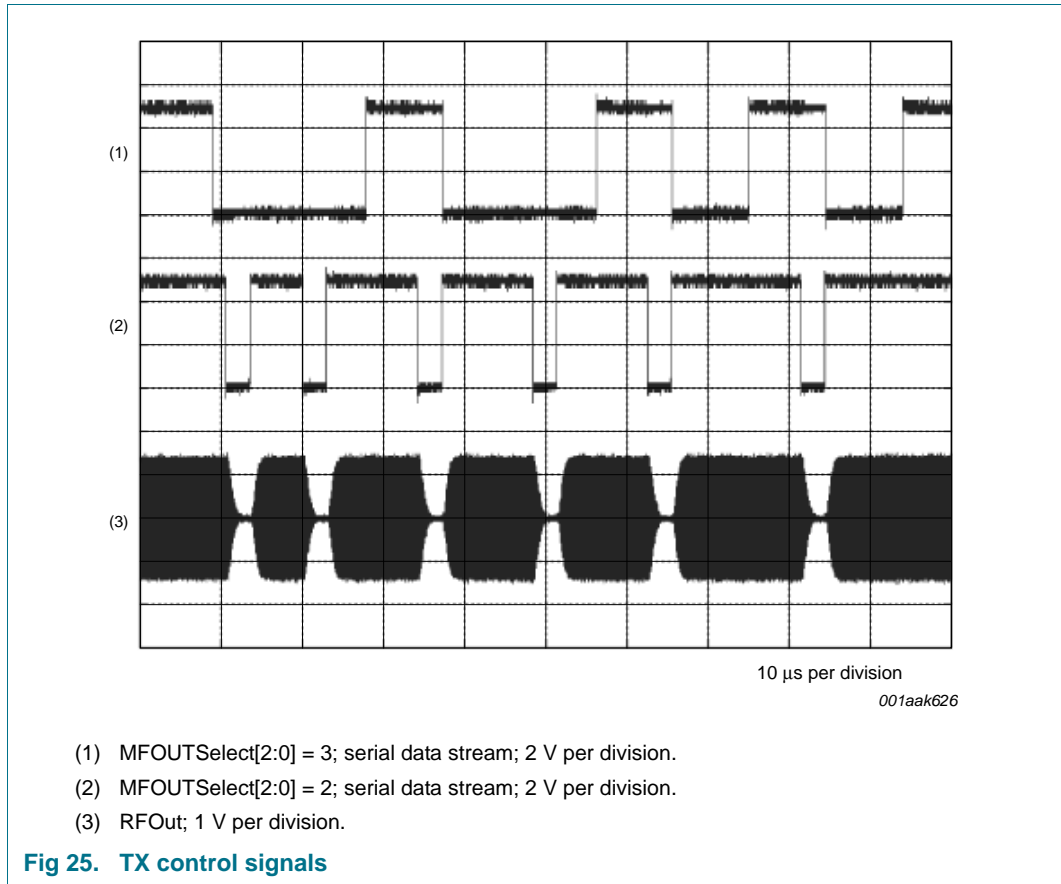
| SignalToMFOUT | MFOUTSelect | Signal routed to pin MFOUT |
|---------------|-------------|----------------------------|
| 0             | 0           | LOW                        |
| 0             | 1           | HIGH                       |
| 0             | 2           | envelope                   |
| 0             | 3           | transmit NRZ               |
| 0             | 4           | Manchester with subcarrier |
| 0             | 5           | Manchester                 |
| 0             | 6           | reserved                   |
| 0             | 7           | reserved                   |
| 1             | X           | digital test signal        |

#### 15.2.1.1 TX control

[Figure 25](#) shows as an example of an ISO/IEC 14443 A communication.

The signal is measured on pin MFOUT using the serial signal switch to control the data sent to the card. Setting the flag MFOUTSelect[2:0] = 3 sends the data to the card coded as NRZ. Setting MFOUTSelect[2:0] = 2 shows the data as a Miller coded signal.

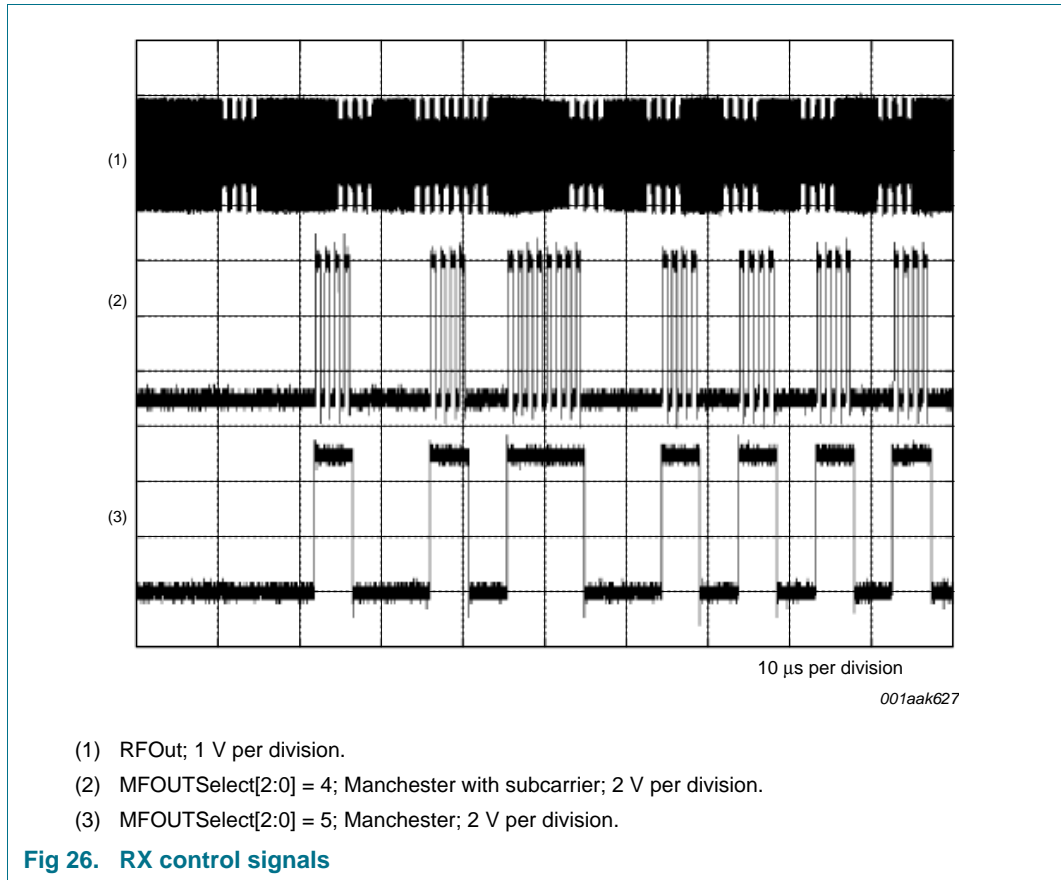
The RFOut signal is measured directly on the antenna and gives the RF signal pulse shape. Refer to Application note *Directly matched Antenna - Excel calculation* ([Ref. 3](#)) for detail information on the RF signal pulse.



**15.2.1.2 RX control**

[Figure 26](#) shows an example of ISO/IEC 14443 A communication which represents the beginning of a card's answer to a request signal.

The RF signal shows the RF voltage measured directly on the antenna so that the card's load modulation is visible. Setting MFOUTSelect[2:0] = 4 shows the Manchester decoded signal with subcarrier. Setting MFOUTSelect[2:0] = 5 shows the Manchester decoded signal.



**15.2.2 Analog test signals**

The analog test signals can be routed to pin AUX by selecting them using the TestAnaSelect register TestAnaOutSel[4:0] bits.

**Table 168. Analog test signal selection**

| Value | Signal Name | Description  |
|-------|-------------|--|
| 0     | VMID        | voltage at internal node VMID  |
| 1     | Vbandgap    | internal reference voltage generated by the bandgap                          |
| 2     | VRxFollI    | output signal from the demodulator using the I-clock                         |
| 3     | VRxFollQ    | output signal from the demodulator using the Q-clock                         |
| 4     | VRxAmpI     | I-channel subcarrier signal amplified and filtered                           |
| 5     | VRxAmpQ     | Q-channel subcarrier signal amplified and filtered                           |
| 6     | VCorrNI     | output signal of N-channel correlator fed by the I-channel subcarrier signal |
| 7     | VCorrNQ     | output signal of N-channel correlator fed by the Q-channel subcarrier signal |
| 8     | VCorrDI     | output signal of D-channel correlator fed by the I-channel subcarrier signal |
| 9     | VCorrDQ     | output signal of D-channel correlator fed by the Q-channel subcarrier signal |
| A     | VEvalL      | evaluation signal from the left half-bit                                     |

Table 168. Analog test signal selection ...continued

| Value | Signal Name | Description                               |
|-------|-------------|---|
| B     | VEvalR      | evaluation signal from the right half-bit |
| C     | VTemp       | temperature voltage derived from band gap |
| D     | reserved    | reserved for future use                   |
| E     | reserved    | reserved for future use                   |
| F     | reserved    | reserved for future use                   |

### 15.2.3 Digital test signals

Digital test signals can be routed to pin MFOUT by setting bit SignalToMFOUT = logic 1. A digital test signal is selected using the TestDigiSelect register TestDigiSignalSel[6:0] bits. The signals selected by the TestDigiSignalSel[6:0] bits are shown in [Table 169](#).

Table 169. Digital test signal selection

| TestDigiSignalSel[6:0] | Signal name    | Description   |
|------------------------|----------------|---|
| F4h                    | s_data         | data received from the card   |
| E4h                    | s_valid        | when logic 1 is returned the s_data and s_coll signals are valid  |
| D4h                    | s_coll         | when logic 1 is returned a collision has been detected in the current bit   |
| C4h                    | s_clock        | internal serial clock:<br>during transmission, this is the encoder clock<br>during reception this is the receiver clock |
| B5h                    | rd_sync        | internal synchronized read signal which is derived from the parallel microprocessor interface                           |
| A5h                    | wr_sync        | internal synchronized write signal which is derived from the parallel microprocessor interface                          |
| 96h                    | int_clock      | internal 13.56 MHz clock  |
| 83h                    | BPSK_out       | BPSK output signal  |
| E2h                    | BPSK_sig       | BPSK signal's amplitude detected  |
| 00h                    | no test signal | output as defined by the MFOUTSelect register MFOUTSelect[2:0] bits routed to pin MFOUT                                 |

If test signals are not used, the TestDigiSelect register address value must be 00h.

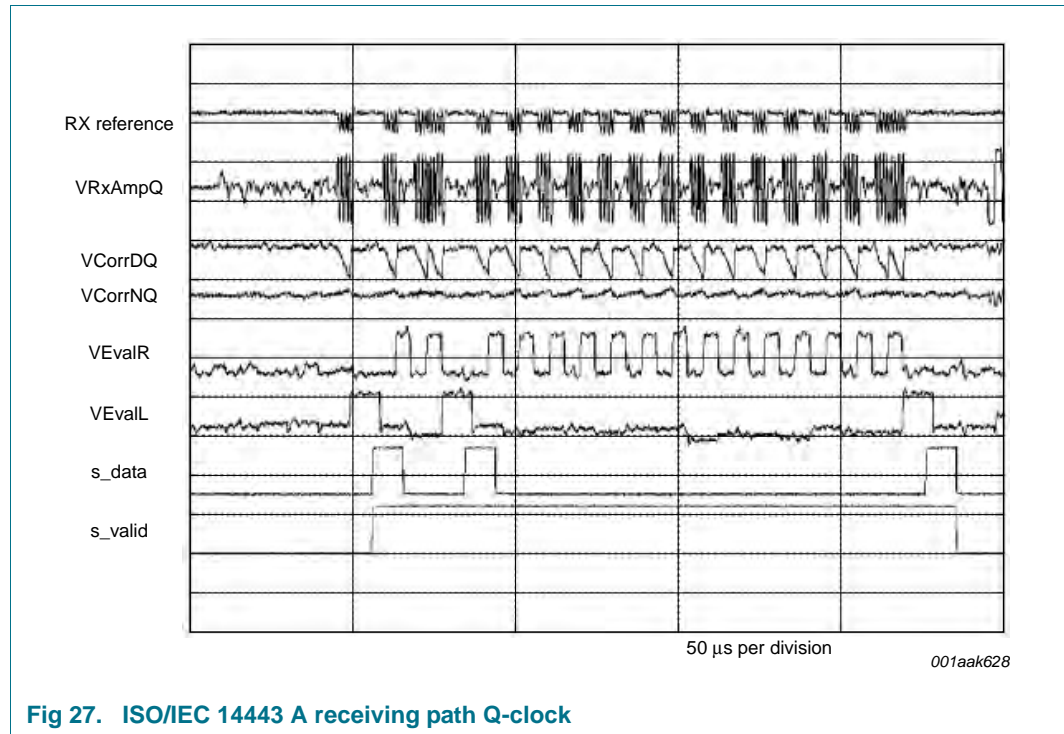
**Remark:** All other values for TestDigiSignalSel[6:0] are for production test purposes only.

### 15.2.4 Analog and digital test signal Examples

[Figure 27](#) shows a MIFARE card's answer to a request command using the Q-clock receiving path. RX reference is given to show the Manchester modulated signal on pin RX.

The signal is demodulated and amplified in the receiver circuitry. Signal VRXampQ is the amplified side-band signal using the Q-clock for demodulation. The signals VCorrDQ and VCorrNQ were generated in the correlation circuitry. They are processed further in the evaluation and digitizer circuitry.

Signals VEvalR and VEvalL show the evaluation of the signal's right and left half-bit. Finally, the digital test signal s\_data shows the received data. This is then sent to the internal digital circuit and s\_valid which indicates the received data stream is valid.



**Fig 27. ISO/IEC 14443 A receiving path Q-clock**

16. Package outline

SO32: plastic small outline package; 32 leads; body width 7.5 mm

SOT287-1

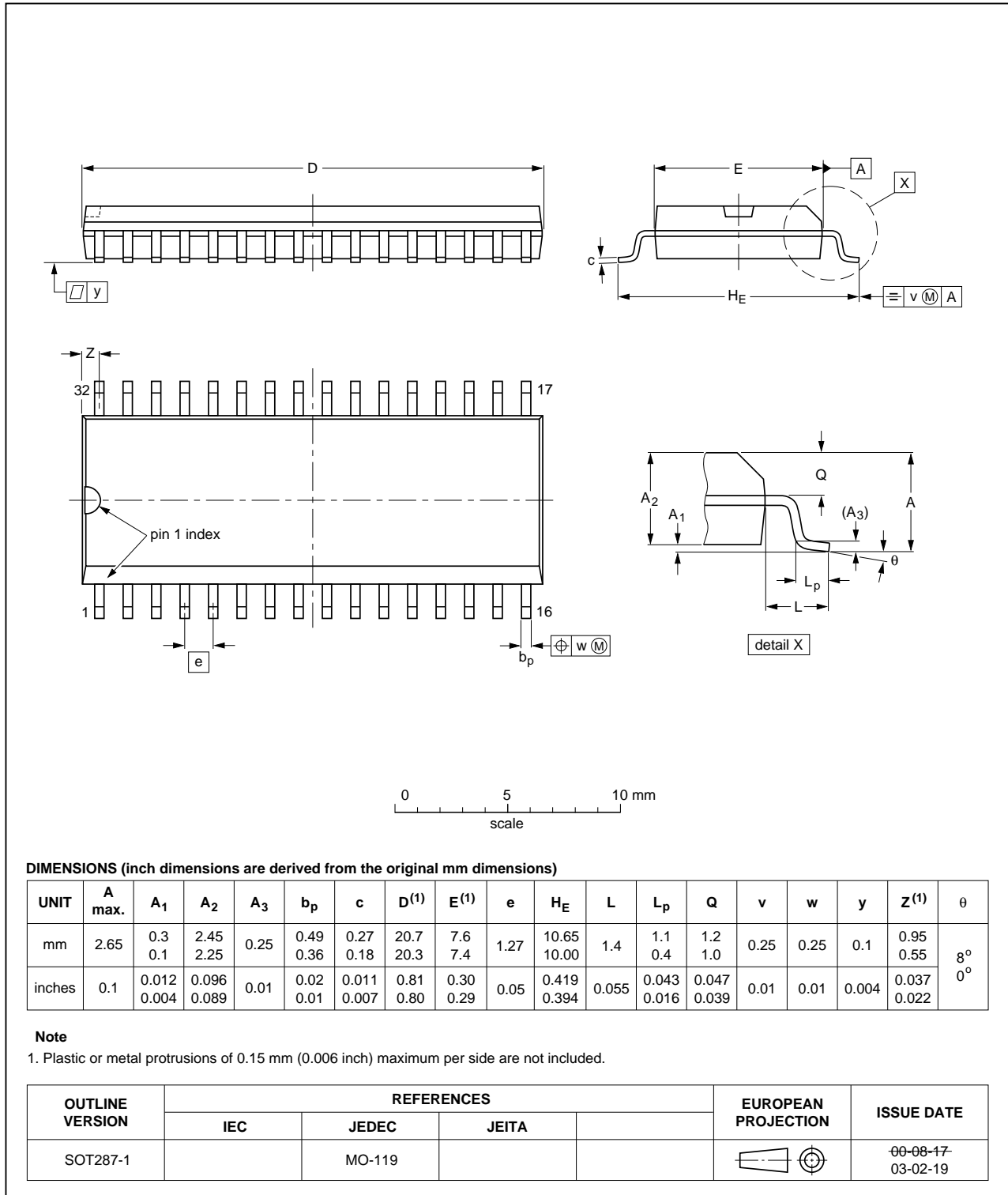


Fig 28. Package outline SOT287-1



## 17. Abbreviations

Table 170. Abbreviations and acronyms

| Acronym | Description                             |
|---------|---|
| ASK     | Amplitude-Shift Keying                  |
| BPSK    | Binary Phase-Shift Keying               |
| CMOS    | Complementary Metal-Oxide Semiconductor |
| CRC     | Cyclic Redundancy Check                 |
| EOF     | End Of Frame                            |
| EPP     | Enhanced Parallel Port                  |
| ETU     | Elementary Time Unit                    |
| FIFO    | First In, First Out                     |
| HBM     | Human Body Model                        |
| LSB     | Least Significant Bit                   |
| MM      | Machine Model                           |
| MSB     | Most Significant Bit                    |
| NRZ     | None Return to Zero                     |
| POR     | Power-On Reset                          |
| PCD     | Proximity Coupling Device               |
| PICC    | Proximity Integrated Circuit Card       |
| SOF     | Start Of Frame                          |
| SPI     | Serial Peripheral Interface             |

## 18. References

- [1] **Application note** — MICORE reader IC family; Directly Matched Antenna Design.
- [2] **Application note** — MIFARE (14443 A) 13.56 MHz RFID Proximity Antennas.
- [3] **Application note** — Directly matched Antenna - Excel calculation.
- [4] **ISO standard** — ISO/IEC 14443 Identification cards - Contactless integrated circuit(s) cards - Proximity cards, part 1-4.
- [5] **Application note** — MIFARE Implementation of Higher Baud rates.

## 19. Revision history

Table 171. Revision history

| Document ID    | Release date  | Data sheet status      | Change notice | Supersedes |
|----------------|---|------------------------|---------------|------------|
| MFRC530 v. 3.4 | 20140212  | Product data sheet     | -             | MFRC530_33 |
| Modifications: | <ul style="list-style-type: none"> <li>• Descriptive title changed</li> <li>• <a href="#">Section 2 “General description”</a>: updated</li> </ul>   |                        |               |            |
| MFRC530_33     | 20100706  | Product data sheet     | -             | 057432     |
| Modifications: | <ul style="list-style-type: none"> <li>• The format of this data sheet has been redesigned to comply with the new identity guidelines of NXP Semiconductors</li> <li>• Legal texts have been adapted to the new company name where appropriate</li> <li>• This version supersedes all previous revisions.</li> <li>• The symbols for electrical characteristics and their parameters have been updated to meet the NXP Semiconductors' guidelines</li> <li>• A number of inconsistencies in pin, register and bit names have been eliminated from the data sheet</li> <li>• All drawings have been updated</li> <li>• <a href="#">Section 5 “Quick reference data” on page 3</a>: section added</li> <li>• <a href="#">Section 15.1.2.4 “Antenna coil” on page 98</a>: added missing formula and updated the last clause</li> <li>• <a href="#">Section 16 “Package outline” on page 104</a>: updated</li> <li>• <a href="#">Section 18 “References” on page 105</a>: added section and updated the references in the document</li> </ul> |                        |               |            |
| 057432         | 20051210  | Product data sheet     | -             | 057431     |
| 057431         | 20040523  | Product data sheet     | -             | 057430     |
| 057430         | 20021102  | Product data sheet     | -             | 057420     |
| 057420         | 20011218  | Preliminary data sheet | -             | 057410     |
| 057410         | 20010812  | Objective data sheet   | -             | -          |

## 20. Legal information

### 20.1 Data sheet status

| Document status <sup>[1][2]</sup> | Product status <sup>[3]</sup> | Definition  |
|-----------------------------------|-------------------------------|---|
| Objective [short] data sheet      | Development                   | This document contains data from the objective specification for product development. |
| Preliminary [short] data sheet    | Qualification                 | This document contains data from the preliminary specification.                       |
| Product [short] data sheet        | Production                    | This document contains the product specification.                                     |

[1] Please consult the most recently issued document before initiating or completing a design.

[2] The term 'short data sheet' is explained in section "Definitions".

[3] The product status of device(s) described in this document may have changed since this document was published and may differ in case of multiple devices. The latest product status information is available on the Internet at URL <http://www.nxp.com>.

### 20.2 Definitions

**Draft** — The document is a draft version only. The content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included herein and shall have no liability for the consequences of use of such information.

**Short data sheet** — A short data sheet is an extract from a full data sheet with the same product type number(s) and title. A short data sheet is intended for quick reference only and should not be relied upon to contain detailed and full information. For detailed and full information see the relevant full data sheet, which is available on request via the local NXP Semiconductors sales office. In case of any inconsistency or conflict with the short data sheet, the full data sheet shall prevail.

**Product specification** — The information and data provided in a Product data sheet shall define the specification of the product as agreed between NXP Semiconductors and its customer, unless NXP Semiconductors and customer have explicitly agreed otherwise in writing. In no event however, shall an agreement be valid in which the NXP Semiconductors product is deemed to offer functions and qualities beyond those described in the Product data sheet.

### 20.3 Disclaimers

**Limited warranty and liability** — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the *Terms and conditions of commercial sale* of NXP Semiconductors.

**Right to make changes** — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

**Suitability for use** — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors and its suppliers accept no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

**Applications** — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

**Limiting values** — Stress above one or more limiting values (as defined in the Absolute Maximum Ratings System of IEC 60134) will cause permanent damage to the device. Limiting values are stress ratings only and (proper) operation of the device at these or any other conditions above those given in the Recommended operating conditions section (if present) or the Characteristics sections of this document is not warranted. Constant or repeated exposure to limiting values will permanently and irreversibly affect the quality and reliability of the device.

**Terms and conditions of commercial sale** — NXP Semiconductors products are sold subject to the general terms and conditions of commercial sale, as published at <http://www.nxp.com/profile/terms>, unless otherwise agreed in a valid written individual agreement. In case an individual agreement is concluded only the terms and conditions of the respective agreement shall apply. NXP Semiconductors hereby expressly objects to applying the customer's general terms and conditions with regard to the purchase of NXP Semiconductors products by customer.

**No offer to sell or license** — Nothing in this document may be interpreted or construed as an offer to sell products that is open for acceptance or the grant, conveyance or implication of any license under any copyrights, patents or other industrial or intellectual property rights.

**Export control** — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

**Quick reference data** — The Quick reference data is an extract of the product data given in the Limiting values and Characteristics sections of this document, and as such is not complete, exhaustive or legally binding.

**Non-automotive qualified products** — Unless this data sheet expressly states that this specific NXP Semiconductors product is automotive qualified, the product is not suitable for automotive use. It is neither qualified nor tested in accordance with automotive testing or application requirements. NXP Semiconductors accepts no liability for inclusion and/or use of non-automotive qualified products in automotive equipment or applications.

In the event that customer uses the product for design-in and use in automotive applications to automotive specifications and standards, customer (a) shall use the product without NXP Semiconductors' warranty of the product for such automotive applications, use and specifications, and (b)

whenever customer uses the product for automotive applications beyond NXP Semiconductors' specifications such use shall be solely at customer's own risk, and (c) customer fully indemnifies NXP Semiconductors for any liability, damages or failed product claims resulting from customer design and use of the product for automotive applications beyond NXP Semiconductors' standard warranty and NXP Semiconductors' product specifications.

**Translations** — A non-English (translated) version of a document is for reference only. The English version shall prevail in case of any discrepancy between the translated and English versions.

## 20.4 Trademarks

Notice: All referenced brands, product names, service names and trademarks are the property of their respective owners.

**MIFARE** — is a trademark of NXP B.V.

## 21. Contact information

For more information, please visit: <http://www.nxp.com>

For sales office addresses, please send an email to: [salesaddresses@nxp.com](mailto:salesaddresses@nxp.com)

## 22. Tables

|  |    |   |    |
|--|----|---|----|
| Table 1. Quick reference data . . . . .  | 3  | Table 39. Page register (address: 00h, 08h, 10h, 18h, 20h, 28h, 30h, 38h) reset value: 1000 0000b, 80h bit allocation . . . . . | 46 |
| Table 2. Ordering information . . . . .  | 3  | Table 40. Page register bit descriptions . . . . .  | 46 |
| Table 3. Pin description . . . . .   | 5  | Table 41. Command register (address: 01h) reset value: x000 0000b, x0h bit allocation . . . . .                                 | 46 |
| Table 4. Supported microprocessor and EPP interface signals . . . . .  | 7  | Table 42. Command register bit descriptions . . . . .   | 46 |
| Table 5. Connection scheme for detecting the parallel interface type . . . . .                               | 8  | Table 43. FIFOData register (address: 02h) reset value: xxxx xxxxb, xxh bit allocation . . . . .                                | 47 |
| Table 6. SPI compatibility . . . . .   | 10 | Table 44. FIFOData register bit descriptions . . . . .  | 47 |
| Table 7. SPI read data . . . . .   | 10 | Table 45. PrimaryStatus register (address: 03h) reset value: 0000 0101b, 05h bit allocation . . . . .                           | 47 |
| Table 8. SPI read address . . . . .  | 11 | Table 46. PrimaryStatus register bit descriptions . . . . .   | 47 |
| Table 9. SPI write data . . . . .  | 11 | Table 47. FIFOLength register (address: 04h) reset value: 0000 0000b, 00h bit allocation . . . . .                              | 48 |
| Table 10. SPI write address . . . . .  | 11 | Table 48. FIFOLength bit descriptions . . . . .   | 48 |
| Table 11. EEPROM memory organization diagram . . . . .   | 12 | Table 49. SecondaryStatus register (address: 05h) reset value: 01100 000b, 60h bit allocation . . . . .                         | 49 |
| Table 12. Product information field byte allocation . . . . .  | 13 | Table 50. SecondaryStatus register bit descriptions . . . . .   | 49 |
| Table 13. Product information field byte description . . . . .   | 13 | Table 51. InterruptEn register (address: 06h) reset value: 0000 0000b, 00h bit allocation . . . . .                             | 49 |
| Table 14. Product type identification definition . . . . .   | 13 | Table 52. InterruptEn register bit descriptions . . . . .   | 49 |
| Table 15. Byte assignment for register initialization at start-up . . . . .                                  | 14 | Table 53. InterruptRq register (address: 07h) reset value: 0000 0000b, 00h bit allocation . . . . .                             | 50 |
| Table 16. Shipment content of StartUp configuration file . . . . .   | 14 | Table 54. InterruptRq register bit descriptions . . . . .   | 50 |
| Table 17. Byte assignment for register initialization at startup . . . . .                                   | 15 | Table 55. Control register (address: 09h) reset value: 0000 0000b, 00h bit allocation . . . . .                                 | 51 |
| Table 18. FIFO buffer access . . . . .   | 17 | Table 56. Control register bit descriptions . . . . .   | 51 |
| Table 19. Associated FIFO buffer registers and flags . . . . .   | 18 | Table 57. ErrorFlag register (address: 0Ah) reset value: 0100 0000b, 40h bit allocation . . . . .                               | 52 |
| Table 20. Interrupt sources . . . . .  | 19 | Table 58. ErrorFlag register bit descriptions . . . . .   | 52 |
| Table 21. Interrupt control registers . . . . .  | 19 | Table 59. CollPos register (address: 0Bh) reset value: 0000 0000b, 00h bit allocation . . . . .                                 | 53 |
| Table 22. Associated Interrupt request system registers and flags . . . . .                                  | 20 | Table 60. CollPos register bit descriptions . . . . .   | 53 |
| Table 23. Associated timer unit registers and flags . . . . .  | 24 | Table 61. TimerValue register (address: 0Ch) reset value: xxxx xxxxb, xxh bit allocation . . . . .                              | 53 |
| Table 24. Signal on pins during Hard power-down . . . . .  | 25 | Table 62. TimerValue register bit descriptions . . . . .  | 53 |
| Table 25. Pin TX1 configurations . . . . .   | 28 | Table 63. CRCResultLSB register (address: 0Dh) reset value: xxxx xxxxb, xxh bit allocation . . . . .                            | 53 |
| Table 26. Pin TX2 configurations . . . . .   | 28 | Table 64. CRCResultLSB register bit descriptions . . . . .  | 53 |
| Table 27. TX1 and TX2 source resistance of n-channel driver transistor against GsCfgCW or GsCfgMod . . . . . | 29 | Table 65. CRCResultMSB register (address: 0Eh) reset value: xxxx xxxxb, xxh bit allocation . . . . .                            | 54 |
| Table 28. Gain factors for the internal amplifier . . . . .  | 33 | Table 66. CRCResultMSB register bit descriptions . . . . .  | 54 |
| Table 29. DecoderSource[1:0] values . . . . .  | 35 | Table 67. BitFraming register (address: 0Fh) reset value: 0000 0000b, 00h bit allocation . . . . .                              | 54 |
| Table 30. ModulatorSource[1:0] values . . . . .  | 36 | Table 68. BitFraming register bit descriptions . . . . .  | 54 |
| Table 31. MFOUTSelect[2:0] values . . . . .  | 36 | Table 69. TxControl register (address: 11h) reset value: 0101 1000b, 58h bit allocation . . . . .                               | 55 |
| Table 32. Register settings to enable use of the analog circuitry . . . . .                                  | 36 |   |    |
| Table 33. MIFARE higher baud rates . . . . .   | 37 |   |    |
| Table 34. Dedicated address bus: assembling the register address . . . . .                                   | 39 |   |    |
| Table 35. Multiplexed address bus: assembling the register address . . . . .                                 | 39 |   |    |
| Table 36. Behavior and designation of register bits . . . . .  | 40 |   |    |
| Table 37. MFRC530 register overview . . . . .  | 41 |   |    |
| Table 38. MFRC530 register flags overview . . . . .  | 43 |   |    |

continued >>

|   |    |  |    |
|---|----|--|----|
| Table 70. TxControl register bit descriptions . . . . .   | 55 | Table 104. CRCPresetMSB bit descriptions . . . . .   | 64 |
| Table 71. CwConductance register (address: 12h) reset value: 0011 1111b, 3Fh bit allocation . . . . .     | 56 | Table 105. PreSet25 register (address: 25h) reset value: 0000 0000b, 00h bit allocation . . . . .                                | 64 |
| Table 72. CwConductance register bit descriptions . . . . .   | 56 | Table 106. PreSet25 register bit descriptions . . . . .  | 65 |
| Table 73. PreSet13 register (address: 13h) reset value: 0011 1111b, 3Fh bit allocation . . . . .          | 56 | Table 107. MFOUTSelect register (address: 26h) reset value: 0000 0000b, 00h bit allocation . . . . .                             | 65 |
| Table 74. PreSet13 register bit descriptions . . . . .  | 56 | Table 108. MFOUTSelect register bit descriptions . . . . .   | 65 |
| Table 75. CoderControl register (address: 14h) reset value: 0001 1001b, 19h bit allocation . . . . .      | 57 | Table 109. PreSet27 (address: 27h) reset value: xxxx xxxxb, xxh bit allocation . . . . .   | 65 |
| Table 76. CoderControl register bit descriptions . . . . .  | 57 | Table 110. PreSet27 register bit descriptions . . . . .  | 65 |
| Table 77. ModWidth register (address: 15h) reset value: 0001 0011b, 13h bit allocation . . . . .          | 57 | Table 111. FIFOLevel register (address: 29h) reset value: 0000 1000b, 08h bit allocation . . . . .                               | 66 |
| Table 78. ModWidth register bit descriptions . . . . .  | 57 | Table 112. FIFOLevel register bit descriptions . . . . .   | 66 |
| Table 79. PreSet16 register (address: 16h) reset value: 0000 0000b, 00h bit allocation . . . . .          | 58 | Table 113. TimerClock register (address: 2Ah) reset value: 0000 0111b, 07h bit allocation . . . . .                              | 66 |
| Table 80. PreSet16 register bit descriptions . . . . .  | 58 | Table 114. TimerClock register bit descriptions . . . . .  | 66 |
| Table 81. PreSet17 register (address: 17h) reset value: 0000 0000b, 00h bit allocation . . . . .          | 58 | Table 115. TimerControl register (address: 2Bh) reset value: 0000 0110b, 06h bit allocation . . . . .                            | 67 |
| Table 82. PreSet17 register bit descriptions . . . . .  | 58 | Table 116. TimerControl register bit descriptions . . . . .  | 67 |
| Table 83. RxControl1 register (address: 19h) reset value: 0111 0011b, 73h bit allocation . . . . .        | 59 | Table 117. TimerReload register (address: 2Ch) reset value: 0000 1010b, 0Ah bit allocation . . . . .                             | 67 |
| Table 84. RxControl1 register bit descriptions . . . . .  | 59 | Table 118. TimerReload register bit descriptions . . . . .   | 67 |
| Table 85. DecoderControl register (address: 1Ah) reset value: 0000 1000b, 08h bit allocation . . . . .    | 60 | Table 119. IRQPinConfig register (address: 2Dh) reset value: 0000 0010b, 02h bit allocation . . . . .                            | 68 |
| Table 86. DecoderControl register bit descriptions . . . . .  | 60 | Table 120. IRQPinConfig register bit descriptions . . . . .  | 68 |
| Table 87. BitPhase register (address: 1Bh) reset value: 1010 1101b, ADh bit allocation . . . . .          | 60 | Table 121. PreSet2E register (address: 2Eh) reset value: xxxx xxxxb, xxh bit allocation . . . . .                                | 68 |
| Table 88. BitPhase register bit descriptions . . . . .  | 60 | Table 122. PreSet2F register (address: 2Fh) reset value: xxxx xxxxb, xxh bit allocation . . . . .                                | 68 |
| Table 89. RxThreshold register (address: 1Ch) reset value: 1111 1111b, FFh bit allocation . . . . .       | 61 | Table 123. Reserved registers (address: 31h, 32h, 33h, 34h, 35h, 36h, 37h) reset value: xxxx xxxxb, xxh bit allocation . . . . . | 68 |
| Table 90. RxThreshold register bit descriptions . . . . .   | 61 | Table 124. Reserved register (address: 39h) reset value: xxxx xxxxb, xxh bit allocation . . . . .                                | 69 |
| Table 91. BPSKDemControl register (address: 1Dh) reset value: 0001 1110b, 1Eh bit allocation . . . . .    | 61 | Table 125. TestAnaSelect register (address: 3Ah) reset value: 0000 0000b, 00h bit allocation . . . . .                           | 69 |
| Table 92. BPSKDemControl register bit descriptions . . . . .  | 61 | Table 126. TestAnaSelect bit descriptions . . . . .  | 69 |
| Table 93. RxControl2 register (address: 1Eh) reset value: 0100 0001b, 41h bit allocation . . . . .        | 62 | Table 127. Reserved register (address: 3Bh) reset value: xxxx xxxxb, xxh bit allocation . . . . .                                | 70 |
| Table 94. RxControl2 register bit descriptions . . . . .  | 62 | Table 128. Reserved register (address: 3Ch) reset value: xxxx xxxxb, xxh bit allocation . . . . .                                | 70 |
| Table 95. ClockQControl register (address: 1Fh) reset value: 000x xxxxb, xxh bit allocation . . . . .     | 62 | Table 129. TestDigiSelect register (address: 3Dh) reset value: xxxx xxxxb, xxh bit allocation . . . . .                          | 70 |
| Table 96. ClockQControl register bit descriptions . . . . .   | 62 | Table 130. TestDigiSelect register bit descriptions . . . . .  | 70 |
| Table 97. RxWait register (address: 21h) reset value: 0000 0101b, 06h bit allocation . . . . .            | 63 | Table 131. Reserved register (address: 3Eh, 3Fh) reset value: xxxx xxxxb, xxh bit allocation . . . . .                           | 71 |
| Table 98. RxWait register bit descriptions . . . . .  | 63 | Table 132. MFRC530 commands overview . . . . .   | 71 |
| Table 99. ChannelRedundancy register (address: 22h) reset value: 0000 0011b, 03h bit allocation . . . . . | 63 | Table 133. StartUp command 3Fh . . . . .   | 73 |
| Table 100. ChannelRedundancy bit descriptions . . . . .   | 63 | Table 134. Idle command 00h . . . . .  | 73 |
| Table 101. CRCPresetLSB register (address: 23h) reset value: 0101 0011b, 63h bit allocation . . . . .     | 64 | Table 135. Transmit command 1Ah . . . . .  | 74 |
| Table 102. CRCPresetLSB register bit descriptions . . . . .   | 64 | Table 136. Transmission of frames of more than   |    |
| Table 103. CRCPresetMSB register (address: 24h) reset value: 0101 0011b, 63h bit allocation . . . . .     | 64 |  |    |

continued >>

64 bytes .....77

Table 137. Receive command 16h .....77

Table 138. Return values for bit-collision positions .....79

Table 139. Communication error table .....80

Table 140. Transceive command 1Eh .....80

Table 141. Meaning of ModemState .....80

Table 142. WriteE2 command 01h .....82

Table 143. ReadE2 command 03h .....84

Table 144. LoadConfig command 07h .....84

Table 145. CalcCRC command 12h .....85

Table 146. CRC coprocessor parameters .....85

Table 147. ErrorFlag register error flags overview .....86

Table 148. LoadKeyE2 command 0Bh .....86

Table 149. LoadKey command 19h .....86

Table 150. Authent1 command 0Ch .....87

Table 151. Authent2 command 14h .....87

Table 152. Limiting values .....88

Table 153. Operating condition range .....88

Table 154. Current consumption .....89

Table 155. Standard input pin characteristics .....89

Table 156. Schmitt trigger input pin characteristics .....89

Table 157. RSTPD input pin characteristics .....90

Table 158. RX input capacitance and input voltage range 90

Table 159. Digital output pin characteristics .....90

Table 160. Antenna driver output pin characteristics .....91

Table 161. Timing specification for separate read/write  
strobe .....91

Table 162. Common read/write strobe timing  
specification .....92

Table 163. Common read/write strobe timing specification  
for EPP .....93

Table 164. SPI timing specification .....95

Table 165. Clock frequency .....95

Table 166. EEPROM characteristics .....96

Table 167. Signal routed to pin MFOUT .....99

Table 168. Analog test signal selection .....101

Table 169. Digital test signal selection .....102

Table 170. Abbreviations and acronyms .....105

Table 171. Revision history .....106

## 23. Figures

|         |   |     |
|---------|---|-----|
| Fig 1.  | MFRC530 block diagram . . . . .   | 4   |
| Fig 2.  | MFRC530 pin configuration . . . . .   | 5   |
| Fig 3.  | Connection to microprocessor: separate read and write strobes . . . . .             | 8   |
| Fig 4.  | Connection to microprocessor: common read and write strobes . . . . .               | 9   |
| Fig 5.  | Connection to microprocessor: EPP common read/write strobes and handshake . . . . . | 9   |
| Fig 6.  | Connection to microprocessor: SPI . . . . .   | 10  |
| Fig 7.  | Key storage format . . . . .  | 16  |
| Fig 8.  | Timer module block diagram . . . . .  | 22  |
| Fig 9.  | The StartUp procedure. . . . .  | 26  |
| Fig 10. | Quartz clock connection . . . . .   | 27  |
| Fig 11. | Receiver circuit block diagram . . . . .  | 31  |
| Fig 12. | Automatic Q-clock calibration . . . . .   | 32  |
| Fig 13. | Serial signal switch block diagram . . . . .  | 35  |
| Fig 14. | Crypto1 key handling block diagram . . . . .  | 38  |
| Fig 15. | Transmitting bit oriented frames . . . . .  | 75  |
| Fig 16. | Timing for transmitting byte oriented frames . . . . .                              | 76  |
| Fig 17. | Timing for transmitting bit oriented frames. . . . .                                | 76  |
| Fig 18. | Card communication state diagram . . . . .  | 81  |
| Fig 19. | EEPROM programming timing diagram. . . . .  | 83  |
| Fig 20. | Separate read/write strobe timing diagram . . . . .                                 | 92  |
| Fig 21. | Common read/write strobe timing diagram . . . . .                                   | 93  |
| Fig 22. | Timing diagram for common read/write strobe; EPP . . . . .                          | 94  |
| Fig 23. | Timing diagram for SPI . . . . .  | 95  |
| Fig 24. | Application example circuit diagram: directly matched antenna . . . . .             | 96  |
| Fig 25. | TX control signals . . . . .  | 100 |
| Fig 26. | RX control signals . . . . .  | 101 |
| Fig 27. | ISO/IEC 14443 A receiving path Q-clock. . . . .                                     | 103 |
| Fig 28. | Package outline SOT287-1 . . . . .  | 104 |



## 24. Contents

|          |   |          |          |  |    |
|----------|---|----------|----------|--|----|
| <b>1</b> | <b>Introduction</b> . . . . .                                     | <b>1</b> | 9.5      | Timer unit . . . . .   | 21 |
| <b>2</b> | <b>General description</b> . . . . .                              | <b>1</b> | 9.5.1    | Timer unit implementation . . . . .                              | 21 |
| <b>3</b> | <b>Features and benefits</b> . . . . .                            | <b>2</b> | 9.5.1.1  | Timer unit block diagram . . . . .                               | 21 |
| 3.1      | General . . . . .   | 2        | 9.5.1.2  | Controlling the timer unit . . . . .                             | 22 |
| <b>4</b> | <b>Applications</b> . . . . .                                     | <b>2</b> | 9.5.1.3  | Timer unit clock and period . . . . .                            | 23 |
| <b>5</b> | <b>Quick reference data</b> . . . . .                             | <b>3</b> | 9.5.1.4  | Timer unit status . . . . .                                      | 23 |
| <b>6</b> | <b>Ordering information</b> . . . . .                             | <b>3</b> | 9.5.2    | Using the timer unit functions . . . . .                         | 24 |
| <b>7</b> | <b>Block diagram</b> . . . . .                                    | <b>4</b> | 9.5.2.1  | Time-out and WatchDog counters . . . . .                         | 24 |
| <b>8</b> | <b>Pinning information</b> . . . . .                              | <b>5</b> | 9.5.2.2  | Stopwatch . . . . .  | 24 |
| 8.1      | Pin description . . . . .   | 5        | 9.5.2.3  | Programmable one shot timer and periodic<br>trigger . . . . .    | 24 |
| <b>9</b> | <b>Functional description</b> . . . . .                           | <b>7</b> | 9.5.3    | Timer unit registers . . . . .                                   | 24 |
| 9.1      | Digital interface . . . . .                                       | 7        | 9.6      | Power reduction modes . . . . .                                  | 25 |
| 9.1.1    | Overview of supported microprocessor<br>interfaces . . . . .      | 7        | 9.6.1    | Hard power-down . . . . .  | 25 |
| 9.1.2    | Automatic microprocessor interface detection . . . . .            | 7        | 9.6.2    | Soft power-down mode . . . . .                                   | 25 |
| 9.1.3    | Connection to different microprocessor types . . . . .            | 8        | 9.6.3    | Standby mode . . . . .   | 26 |
| 9.1.3.1  | Separate read and write strobe . . . . .                          | 8        | 9.6.4    | Automatic receiver power-down . . . . .                          | 26 |
| 9.1.3.2  | Common read and write strobe . . . . .                            | 9        | 9.7      | StartUp phase . . . . .  | 26 |
| 9.1.3.3  | Common read and write strobe: EPP with<br>handshake . . . . .     | 9        | 9.7.1    | Hard power-down phase . . . . .                                  | 26 |
| 9.1.4    | Serial Peripheral Interface . . . . .                             | 9        | 9.7.2    | Reset phase . . . . .  | 26 |
| 9.1.4.1  | SPI read data . . . . .   | 10       | 9.7.3    | Initialization phase . . . . .                                   | 27 |
| 9.1.4.2  | SPI write data . . . . .  | 11       | 9.7.4    | Initializing the parallel interface type . . . . .               | 27 |
| 9.2      | Memory organization of the EEPROM . . . . .                       | 12       | 9.8      | Oscillator circuit . . . . .                                     | 27 |
| 9.2.1    | Product information field (read only) . . . . .                   | 13       | 9.9      | Transmitter pins TX1 and TX2 . . . . .                           | 28 |
| 9.2.2    | Register initialization files (read/write) . . . . .              | 13       | 9.9.1    | Configuring pins TX1 and TX2 . . . . .                           | 28 |
| 9.2.2.1  | StartUp register initialization file (read/write) . . . . .       | 13       | 9.9.2    | Antenna operating distance versus power<br>consumption . . . . . | 28 |
| 9.2.2.2  | Factory default StartUp register initialization<br>file . . . . . | 14       | 9.9.3    | Antenna driver output source resistance . . . . .                | 29 |
| 9.2.2.3  | Register initialization file (read/write) . . . . .               | 15       | 9.9.3.1  | Source resistance table . . . . .                                | 29 |
| 9.2.3    | Crypto1 keys (write only) . . . . .                               | 15       | 9.9.3.2  | Calculating the relative source resistance . . . . .             | 30 |
| 9.2.3.1  | Key format . . . . .  | 15       | 9.9.3.3  | Calculating the effective source resistance . . . . .            | 30 |
| 9.2.3.2  | Storage of keys in the EEPROM . . . . .                           | 16       | 9.9.4    | Pulse width . . . . .  | 30 |
| 9.3      | FIFO buffer . . . . .   | 16       | 9.10     | Receiver circuit . . . . .                                       | 31 |
| 9.3.1    | Accessing the FIFO buffer . . . . .                               | 16       | 9.10.1   | Receiver circuit block diagram . . . . .                         | 31 |
| 9.3.1.1  | Access rules . . . . .  | 16       | 9.10.2   | Receiver operation . . . . .                                     | 31 |
| 9.3.2    | Controlling the FIFO buffer . . . . .                             | 17       | 9.10.2.1 | Automatic Q-clock calibration . . . . .                          | 32 |
| 9.3.3    | FIFO buffer status information . . . . .                          | 17       | 9.10.2.2 | Amplifier . . . . .  | 33 |
| 9.3.4    | FIFO buffer registers and flags . . . . .                         | 18       | 9.10.2.3 | Correlation circuitry . . . . .                                  | 33 |
| 9.4      | Interrupt request system . . . . .                                | 18       | 9.10.2.4 | Evaluation and digitizer circuitry . . . . .                     | 33 |
| 9.4.1    | Interrupt sources overview . . . . .                              | 18       | 9.11     | Serial signal switch . . . . .                                   | 34 |
| 9.4.2    | Interrupt request handling . . . . .                              | 19       | 9.11.1   | Serial signal switch block diagram . . . . .                     | 34 |
| 9.4.2.1  | Controlling interrupts and getting their status . . . . .         | 19       | 9.11.2   | Serial signal switch registers . . . . .                         | 35 |
| 9.4.2.2  | Accessing the interrupt registers . . . . .                       | 19       | 9.11.2.1 | Active antenna concept . . . . .                                 | 36 |
| 9.4.3    | Configuration of pin IRQ . . . . .                                | 20       | 9.11.2.2 | Driving both RF parts . . . . .                                  | 37 |
| 9.4.4    | Register overview interrupt request system . . . . .              | 20       | 9.12     | MIFARE higher baud rates . . . . .                               | 37 |
|          |   |          | 9.13     | MIFARE authentication and Crypto1 . . . . .                      | 37 |
|          |   |          | 9.13.1   | Crypto1 key handling . . . . .                                   | 38 |
|          |   |          | 9.13.2   | Authentication procedure . . . . .                               | 38 |

continued >>

|           |  |           |           |   |
|-----------|--|-----------|-----------|---|
| <b>10</b> | <b>MFRC530 registers</b>                 | <b>39</b> |           |   |
| 10.1      | Register addressing modes                | 39        | 10.5.5.6  | PreSet25 register                                       |
| 10.1.1    | Page registers                           | 39        | 10.5.5.7  | MFOUSelect register                                     |
| 10.1.2    | Dedicated address bus                    | 39        | 10.5.5.8  | PreSet27 register                                       |
| 10.1.3    | Multiplexed address bus                  | 39        | 10.5.6    | Page 5: FIFO, timer and IRQ pin configuration           |
| 10.2      | Register bit behavior                    | 40        | 10.5.6.1  | Page register   |
| 10.3      | Register overview                        | 41        | 10.5.6.2  | FIFOLevel register                                      |
| 10.4      | MFRC530 register flags overview          | 43        | 10.5.6.3  | TimerClock register                                     |
| 10.5      | Register descriptions                    | 46        | 10.5.6.4  | TimerControl register                                   |
| 10.5.1    | Page 0: Command and status               | 46        | 10.5.6.5  | TimerReload register                                    |
| 10.5.1.1  | Page register                            | 46        | 10.5.6.6  | IRQPinConfig register                                   |
| 10.5.1.2  | Command register                         | 46        | 10.5.6.7  | PreSet2E register                                       |
| 10.5.1.3  | FIFOData register                        | 47        | 10.5.6.8  | PreSet2F register                                       |
| 10.5.1.4  | PrimaryStatus register                   | 47        | 10.5.7    | Page 6: reserved  |
| 10.5.1.5  | FIFOLength register                      | 48        | 10.5.7.1  | Page register   |
| 10.5.1.6  | SecondaryStatus register                 | 49        | 10.5.7.2  | Reserved registers 31h, 32h, 33h, 34h, 35h, 36h and 37h |
| 10.5.1.7  | InterruptEn register                     | 49        | 10.5.8    | Page 7: Test control                                    |
| 10.5.1.8  | InterruptRq register                     | 50        | 10.5.8.1  | Page register   |
| 10.5.2    | Page 1: Control and status               | 51        | 10.5.8.2  | Reserved register 39h                                   |
| 10.5.2.1  | Page register                            | 51        | 10.5.8.3  | TestAnaSelect register                                  |
| 10.5.2.2  | Control register                         | 51        | 10.5.8.4  | Reserved register 3Bh                                   |
| 10.5.2.3  | ErrorFlag register                       | 52        | 10.5.8.5  | Reserved register 3Ch                                   |
| 10.5.2.4  | CollPos register                         | 53        | 10.5.8.6  | TestDigiSelect register                                 |
| 10.5.2.5  | TimerValue register                      | 53        | 10.5.8.7  | Reserved registers 3Eh, 3Fh                             |
| 10.5.2.6  | CRCResultLSB register                    | 53        | <b>11</b> | <b>MFRC530 command set</b>                              |
| 10.5.2.7  | CRCResultMSB register                    | 54        | 11.1      | MFRC530 command overview                                |
| 10.5.2.8  | BitFraming register                      | 54        | 11.1.1    | Basic states  |
| 10.5.3    | Page 2: Transmitter and control          | 55        | 11.1.2    | StartUp command 3Fh                                     |
| 10.5.3.1  | Page register                            | 55        | 11.1.3    | Idle command 00h  |
| 10.5.3.2  | TxControl register                       | 55        | 11.2      | Commands for card communication                         |
| 10.5.3.3  | CwConductance register                   | 56        | 11.2.1    | Transmit command 1Ah                                    |
| 10.5.3.4  | PreSet13 register                        | 56        | 11.2.1.1  | Using the Transmit command                              |
| 10.5.3.5  | CoderControl register                    | 57        | 11.2.1.2  | RF channel redundancy and framing                       |
| 10.5.3.6  | ModWidth register                        | 57        | 11.2.1.3  | Transmission of bit oriented frames                     |
| 10.5.3.7  | PreSet16 register                        | 58        | 11.2.1.4  | Transmission of frames with more than 64 bytes          |
| 10.5.3.8  | PreSet17 register                        | 58        | 11.2.2    | Receive command 16h                                     |
| 10.5.4    | Page 3: Receiver and decoder control     | 59        | 11.2.2.1  | Using the Receive command                               |
| 10.5.4.1  | Page register                            | 59        | 11.2.2.2  | RF channel redundancy and framing                       |
| 10.5.4.2  | RxControl1 register                      | 59        | 11.2.2.3  | Collision detection                                     |
| 10.5.4.3  | DecoderControl register                  | 60        | 11.2.2.4  | Receiving bit oriented frames                           |
| 10.5.4.4  | BitPhase register                        | 60        | 11.2.2.5  | Communication errors                                    |
| 10.5.4.5  | RxThreshold register                     | 61        | 11.2.3    | Transceive command 1Eh                                  |
| 10.5.4.6  | BPSKDemControl                           | 61        | 11.2.4    | Card communication states                               |
| 10.5.4.7  | RxControl2 register                      | 62        | 11.2.5    | Card communication state diagram                        |
| 10.5.4.8  | ClockQControl register                   | 62        | 11.3      | EEPROM commands   |
| 10.5.5    | Page 4: RF Timing and channel redundancy | 63        | 11.3.1    | WriteE2 command 01h                                     |
| 10.5.5.1  | Page register                            | 63        | 11.3.1.1  | Programming process                                     |
| 10.5.5.2  | RxWait register                          | 63        | 11.3.1.2  | Timing diagram  |
| 10.5.5.3  | ChannelRedundancy register               | 63        | 11.3.1.3  | WriteE2 command error flags                             |
| 10.5.5.4  | CRCPresetLSB register                    | 64        |           |   |
| 10.5.5.5  | CRCPresetMSB register                    | 64        |           |   |

continued >>

|           |   |            |           |                                      |            |
|-----------|---|------------|-----------|--------------------------------------|------------|
| 11.3.2    | ReadE2 command 03h . . . . .                      | 84         | <b>17</b> | <b>Abbreviations . . . . .</b>       | <b>105</b> |
| 11.3.2.1  | ReadE2 command error flags . . . . .              | 84         | <b>18</b> | <b>References . . . . .</b>          | <b>105</b> |
| 11.4      | Diverse commands . . . . .                        | 84         | <b>19</b> | <b>Revision history . . . . .</b>    | <b>106</b> |
| 11.4.1    | LoadConfig command 07h . . . . .                  | 84         | <b>20</b> | <b>Legal information . . . . .</b>   | <b>107</b> |
| 11.4.1.1  | Register assignment . . . . .                     | 84         | 20.1      | Data sheet status . . . . .          | 107        |
| 11.4.1.2  | Relevant LoadConfig command error flags . .       | 85         | 20.2      | Definitions . . . . .                | 107        |
| 11.4.2    | CalcCRC command 12h . . . . .                     | 85         | 20.3      | Disclaimers . . . . .                | 107        |
| 11.4.2.1  | CRC coprocessor settings . . . . .                | 85         | 20.4      | Trademarks . . . . .                 | 108        |
| 11.4.2.2  | CRC coprocessor status flags . . . . .            | 85         | <b>21</b> | <b>Contact information . . . . .</b> | <b>108</b> |
| 11.5      | Error handling during command execution . .       | 86         | <b>22</b> | <b>Tables . . . . .</b>              | <b>109</b> |
| 11.6      | MIFARE security commands . . . . .                | 86         | <b>23</b> | <b>Figures . . . . .</b>             | <b>112</b> |
| 11.6.1    | LoadKeyE2 command 0Bh . . . . .                   | 86         | <b>24</b> | <b>Contents . . . . .</b>            | <b>113</b> |
| 11.6.1.1  | Relevant LoadKeyE2 command error flags . .        | 86         |           |                                      |            |
| 11.6.2    | LoadKey command 19h . . . . .                     | 86         |           |                                      |            |
| 11.6.2.1  | Relevant LoadKey command error flags . . . .      | 87         |           |                                      |            |
| 11.6.3    | Authent1 command 0Ch . . . . .                    | 87         |           |                                      |            |
| 11.6.4    | Authent2 command 14h . . . . .                    | 87         |           |                                      |            |
| 11.6.4.1  | Authent2 command effects . . . . .                | 88         |           |                                      |            |
| <b>12</b> | <b>Limiting values . . . . .</b>                  | <b>88</b>  |           |                                      |            |
| <b>13</b> | <b>Characteristics . . . . .</b>                  | <b>88</b>  |           |                                      |            |
| 13.1      | Operating condition range . . . . .               | 88         |           |                                      |            |
| 13.2      | Current consumption . . . . .                     | 89         |           |                                      |            |
| 13.3      | Pin characteristics . . . . .                     | 89         |           |                                      |            |
| 13.3.1    | Input pin characteristics . . . . .               | 89         |           |                                      |            |
| 13.3.2    | Digital output pin characteristics . . . . .      | 90         |           |                                      |            |
| 13.3.3    | Antenna driver output pin characteristics . . . . | 91         |           |                                      |            |
| 13.4      | AC electrical characteristics . . . . .           | 91         |           |                                      |            |
| 13.4.1    | Separate read/write strobe bus timing . . . . .   | 91         |           |                                      |            |
| 13.4.2    | Common read/write strobe bus timing . . . . .     | 92         |           |                                      |            |
| 13.4.3    | EPP bus timing . . . . .                          | 93         |           |                                      |            |
| 13.4.4    | SPI timing . . . . .                              | 95         |           |                                      |            |
| 13.4.5    | Clock frequency . . . . .                         | 95         |           |                                      |            |
| <b>14</b> | <b>EEPROM characteristics . . . . .</b>           | <b>96</b>  |           |                                      |            |
| <b>15</b> | <b>Application information . . . . .</b>          | <b>96</b>  |           |                                      |            |
| 15.1      | Typical application . . . . .                     | 96         |           |                                      |            |
| 15.1.1    | Circuit diagram . . . . .                         | 96         |           |                                      |            |
| 15.1.2    | Circuit description . . . . .                     | 97         |           |                                      |            |
| 15.1.2.1  | EMC low-pass filter . . . . .                     | 97         |           |                                      |            |
| 15.1.2.2  | Antenna matching . . . . .                        | 97         |           |                                      |            |
| 15.1.2.3  | Receiver circuit . . . . .                        | 97         |           |                                      |            |
| 15.1.2.4  | Antenna coil . . . . .                            | 98         |           |                                      |            |
| 15.2      | Test signals . . . . .                            | 98         |           |                                      |            |
| 15.2.1    | Measurements using the serial signal switch .     | 99         |           |                                      |            |
| 15.2.1.1  | TX control . . . . .                              | 99         |           |                                      |            |
| 15.2.1.2  | RX control . . . . .                              | 100        |           |                                      |            |
| 15.2.2    | Analog test signals . . . . .                     | 101        |           |                                      |            |
| 15.2.3    | Digital test signals . . . . .                    | 102        |           |                                      |            |
| 15.2.4    | Analog and digital test signal Examples . . . .   | 102        |           |                                      |            |
| <b>16</b> | <b>Package outline . . . . .</b>                  | <b>104</b> |           |                                      |            |

Please be aware that important notices concerning this document and the product(s) described herein, have been included in section 'Legal information'.

© NXP B.V. 2014.

All rights reserved.

For more information, please visit: <http://www.nxp.com>

For sales office addresses, please send an email to: [salesaddresses@nxp.com](mailto:salesaddresses@nxp.com)

Date of release: 12 February 2014  
057434