Chipsmall
深圳市津利帝科技有限公司
Geelead Technology Limited
采购部

CLRC632

Standard multi-protocol reader solution

Rev. 3.7 — 27 February 2014
073937

Product data sheet
COMPANY PUBLIC

1. Introduction

This data sheet describes the functionality of the CLRC632 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 CLRC632 supports all variants of the MIFARE Mini, MIFARE 1K, MIFARE 4K and MIFARE Ultralight RF identification protocols. To aid readability throughout this data sheet, the MIFARE Mini, MIFARE 1K, MIFARE 4K and MIFARE Ultralight products and protocols have the generic name MIFARE.

2. General description

The CLRC632 is a highly integrated reader IC for contactless communication at 13.56 MHz. The CLRC632 reader IC provides:

- outstanding modulation and demodulation for passive contactless communication
- a wide range of methods and protocols
- a small, fully integrated package
- pin compatibility with the MFRC500, MFRC530, MFRC531 and SLRC400

All protocol layers of the ISO/IEC 14443 A and ISO/IEC 14443 B communication standards are supported provided:

- additional components, such as the oscillator, power supply, coil etc. are correctly applied.
- standardized protocols, such as ISO/IEC 14443-4 and/or ISO/IEC 14443 B anticollision are correctly implemented

The CLRC632 supports contactless communication using MIFARE higher baud rates (see Section 9.12 on page 40). The receiver module provides a robust and efficient demodulation/decoding circuitry implementation for compatible transponder signals (see Section 9.10 on page 34).

The digital module, manages the complete ISO/IEC 14443 standard framing and error detection (parity and CRC). In addition, it supports the fast MIFARE security algorithm for authenticating the MIFARE products (see Section 9.14 on page 42).

All layers of the I-CODE1 and ISO/IEC 15693 protocols are supported by the CLRC632. The receiver module provides a robust and efficient demodulation/decoding circuitry implementation for I-CODE1 and ISO/IEC 15693 compatible transponder signals. The digital module handles I-CODE1 and ISO/IEC 15693 framing and error detection (CRC).

The internal transmitter module (Section 9.9 on page 31) 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/label response
- Buffered output drivers enable antenna connection using the minimum of external components
- Proximity operating distance up to 100 mm
- Supports both ISO/IEC 14443 A and ISO/IEC 14443 B standards
- Supports MIFARE dual-interface card ICs and the MIFARE Mini, MIFARE 1K, MIFARE 4K protocols
- Contactless communication at MIFARE higher baud rates (up to 424 kBd)
- Supports both I-CODE1 and ISO/IEC 15693 protocols
- Crypto1 and secure non-volatile internal key memory
- Pin-compatible with the MFRC500, MFRC530, MFRC531 and the SLRC400
- Parallel microprocessor interface with internal address latch and IRQ line
- SPI compatibility
- Flexible interrupt handling
- Automatic detection of parallel microprocessor interface type
- 64-byte send and receive FIFO buffer
- Hard reset with low power function
- Software controlled 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		
	Name	Description	Version
CLRC63201T/0FE	SO32	plastic small outline package; 32 leads; body width 7.5 mm	SOT287-1

7. Block diagram

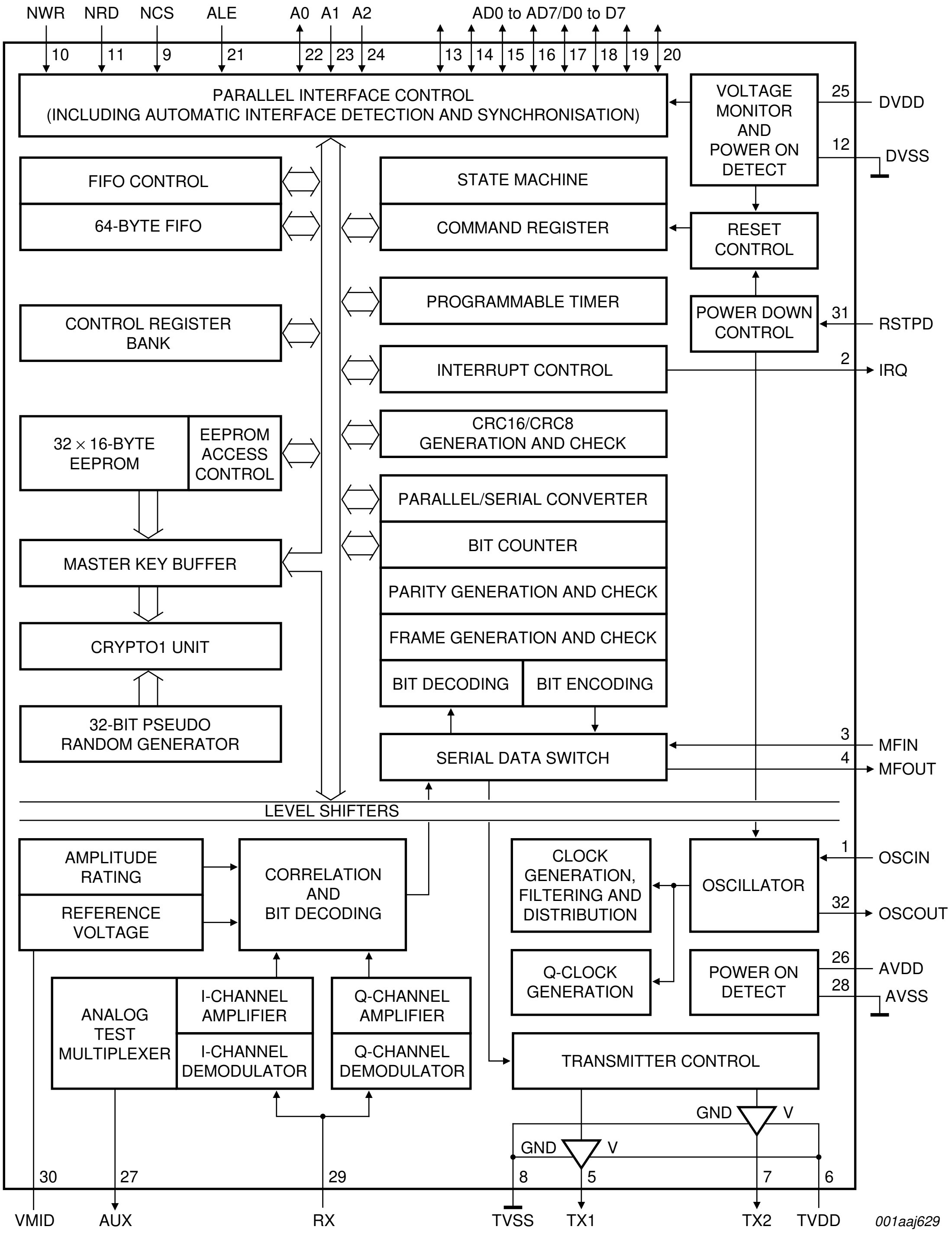


Fig 1. CLRC632 block diagram

8. Pinning information

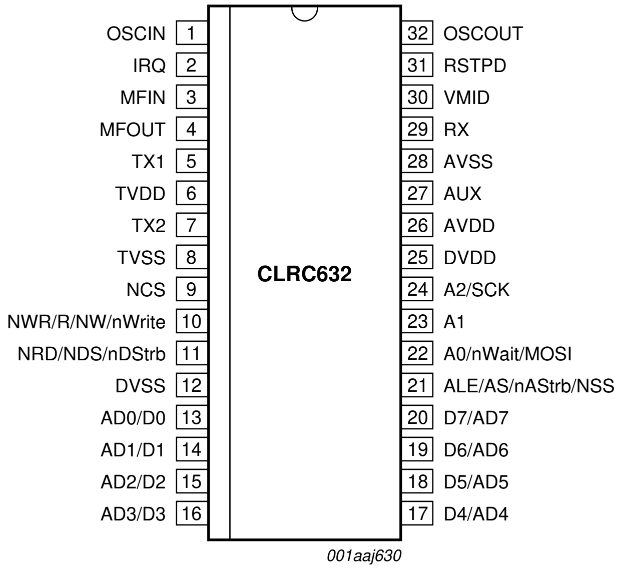


Fig 2. CLRC632 pin configuration

8.1 Pin description

Table 3. Pin description

Pin	Symbol	Type[1]	Description
1	OSCIN	I	oscillator/clock inputs: crystal oscillator input to the oscillator's inverting amplifier externally generated clock input; f_{osc} = 13.56 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[2]	MFOUT	O	interface outputs used as follows: MIFARE: generates serial data ISO/IEC 14443 A I-CODE: generates serial data based on I-CODE1 and ISO/IEC 15693
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 CLRC632's microprocessor interface
10[3]	NWR	I	not write input generates the strobe signal for writing data to the CLRC632 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

Table 3. Pin description *…continued*

Pin	Symbol	Type[1]	Description
11[3]	NRD	I	not read input generates the strobe signal for reading data from the CLRC632 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
12	DVSS	G	digital ground
13	D0	O	SPI master in, slave out output
13 to 20[3]	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[3]	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[3]	A0	I	address line 0 is the address register bit 0 input
	nWait	O	not wait output: LOW starts an access cycle 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[3]	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 **Remark:** It must be connected to a 100 nF block capacitor connected between pin VMID and ground
31	RSTPD	I	reset and power-down input: HIGH: the internal current sinks are switched off, the oscillator is inhibited and the input pads are disconnected 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 CLRC632 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 CLRC632 supports direct interfacing to various 8-bit microprocessors. Alternatively, the CLRC632 can be connected to a PC's Enhanced Parallel Port (EPP). Table 4 shows the parallel interface signals supported by the CLRC632.

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 CLRC632 resets parallel microprocessor interface mode and detects the microprocessor interface type.

The CLRC632 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 30).

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

CLRC632 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	nAStrb
A2	A2	LOW	A2	LOW	HIGH
A1	A1	HIGH	A1	HIGH	HIGH
A0	A0	HIGH	A0	LOW	nWait
NRD	NRD	NRD	NDS	NDS	nDStrb
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

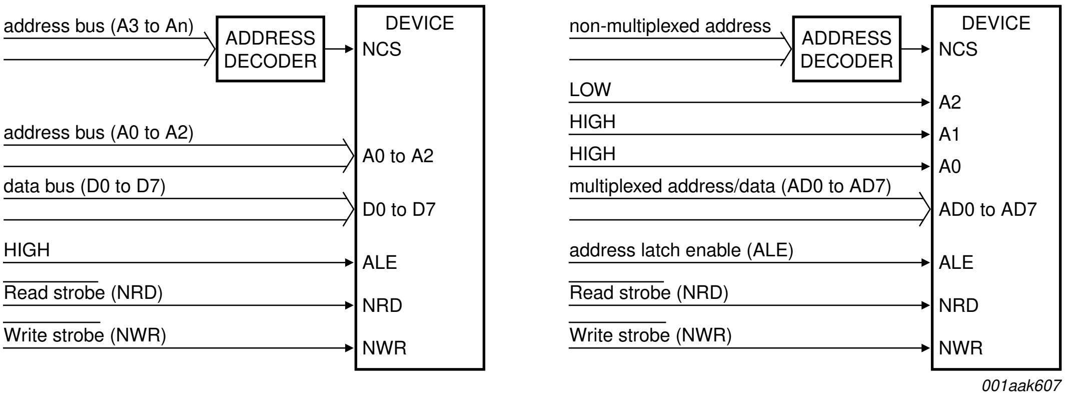


Fig 3. Connection to microprocessor: separate read and write strobes

Refer to Section 13.4.1 on page 102 for timing specification.

9.1.3.2 Common read and write strobe

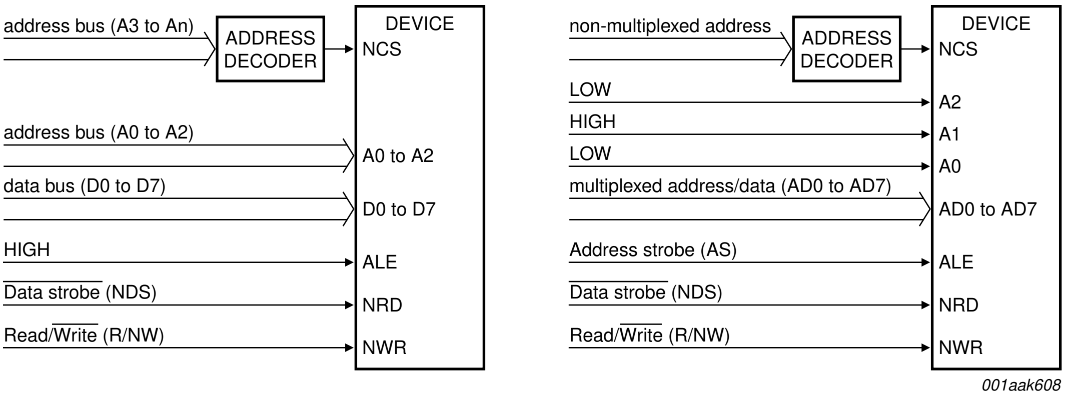


Fig 4. Connection to microprocessor: common read and write strobes

Refer to Section 13.4.2 on page 103 for timing specification.

9.1.3.3 Common read and write strobe: EPP with handshake

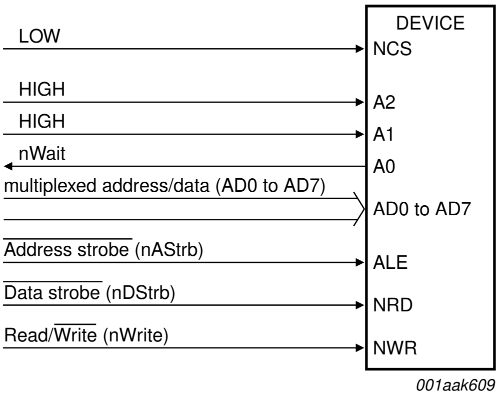


Fig 5. Connection to microprocessor: EPP common read/write strobes and handshake

Refer to Section 13.4.3 on page 104 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 CLRC632 does not support Read Address Cycle.

9.1.4 Serial Peripheral Interface

The CLRC632 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 CLRC632 to the master.

Table 6. SPI compatibility

CLRC632 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 CLRC632 using SPI.

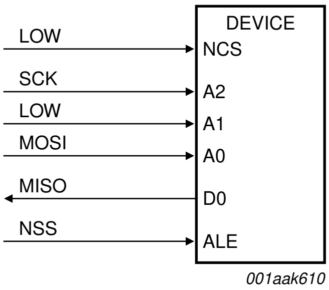


Fig 6. Connection to microprocessor: SPI

Remark: The SPI implementation for CLRC632 conforms to the SPI standard and ensures that the CLRC632 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 CLRC632 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)
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 CLRC632, 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)
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 106 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	Section 9.2.1 on page 13
1	1	10h to 1Fh	R/W	StartUp register initialization file	Section 9.2.2.1 on page 14
2	2	20h to 2Fh	R/W		
3	3	30h to 3Fh	R/W	register initialization file user data or second initialization	Section 9.2.2.3 “Register initialization file (read/write)” on page 16
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	Section 9.2.3 on page 18
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	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	RsMaxP	R	-	maximum source resistance for the p-channel driver transistor on pins TX1 and TX2 The source resistance of the p-channel driver transistors of pin TX1 and TX2 can be adjusted using the value GsCfgCW[5:0] in the CwConductance register (see Section 9.9.3 on page 32). The mean value of the maximum adjustable source resistance for pins TX1 and TX2 is stored as an integer value in Ω in this byte. Typical values for RsMaxP are between 60 Ω to 140 Ω. This value is denoted as maximum adjustable source resistance $R_{S(ref)maxP}$ and is measured by setting the CwConductance register's GsCfgCW[5:0] bits to 01h.
13 to 12	Internal	R	-	two 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 CLRC632 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 Table 13.

Table 13. Product type identification definition

Definition	Product type identification bytes				
Byte	0	1	2	3	4[1]
Value	30h	FFh	FFh	0Fh	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 30) using the StartUp register initialization file.

In addition, the CLRC632 registers can be initialized using values from the register initialization file when the LoadConfig command is executed (see Section 11.5.1 on page 95).

Remark: The following points apply to initialization:

- the Page register (addressed using 10h, 18h, 20h, 28h) is skipped and not initialized.
- make sure that all PreSetxx registers are not changed.
- make sure that all register bits that are reserved are set to logic 0.

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 14.

Table 14. 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 15. During each power-up and initialization phase, these values are written to the CLRC632’s registers.

Table 15. 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	ModConductance	defines the output conductance
14h	14h	19h	CoderControl	ISO/IEC 14443 A coding is set
15h	15h	13h	ModWidth	pulse width for Miller pulse coding is set to standard configuration
16h	16h	3Fh	ModWidthSOF	pulse width of Start Of Frame (SOF)
17h	17h	3Bh	TypeFraming	ISO/IEC 14443 A framing is set
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	TimeSlotPeriod	defines the time for the I-CODE1 time slots
26h	26h	00h	MFOUTSelect	pin MFOUT is set LOW
27h	27h	00h	PreSet27	-
28h	28h	00h	Page	free for user
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	-

Remark: The CLRC632 default configuration supports the MIFARE and ISO/IEC 14443 A communication scheme. Memory addresses 3 to 7 may be used for user-specific initialization files such as I-CODE1, ISO/IEC 15693 or ISO/IEC 14443 B.

9.2.2.3 Register initialization file (read/write)

The EEPROM memory content from block address 3 to 7 can initialize register sub addresses 10h to 2Fh when the LoadConfig command is executed (see Section 11.5.1 on page 95). This command requires the EEPROM starting byte address as a two byte argument for the initialization procedure.

The byte assignment is shown in Table 16.

Table 16. 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. The startup configuration could be adapted to the I-CODE1 StartUp configuration and stored in register block address 3 and 4, providing additional flexibility.

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.

9.2.2.4 Content of I-CODE1 and ISO/IEC 15693 StartUp register values

Table 17 gives an overview of the StartUp values for I-CODE1 and ISO/IEC 15693 communication.

Table 17. Content of I-CODE1 startup configuration

EEPROM byte address	Register address	Value	Symbol	Description
30h	10h	00h	Page	free for user
31h	11h	58h	TxControl	transmitter pins TX1 and TX2 switched off, bridge driver configuration, modulator driven from internal digital circuitry
32h	12h	3Fh	CwConductance	source resistance (R_S) of TX1 and TX2 to minimum
33h	13h	05h	ModGsCfgh	source resistance (R_S) of TX1 and TX2 at the time of modulation, to determine the modulation index
34h	14h	2Ch	CoderControl	selects the bit coding mode and the framing during transmission
35h	15h	3Fh	ModWidth	pulse width for code used (1 out of 256, NRZ or 1 out of 4) pulse coding is set to standard configuration
36h	16h	3Fh	ModWidthSOF	pulse width of SOF
37h	17h	00h	TypeBFraming	-
38h	18h	00h	Page	free for user
39h	19h	8Bh	RxControl1	amplifier gain is maximum
3Ah	1Ah	00h	DecoderControl	bit-collisions always evaluate to HIGH in the data bit stream
3Bh	1Bh	54h	BitPhase	BitPhase[7:0] is set to standard configuration
3Ch	1Ch	68h	RxThreshold:	MinLevel[3:0] and CollLevel[3:0] are set to maximum
3Dh	1Dh	00h	BPSKDemControl	-
3Eh	1Eh	41h	RxControl2	use Q-clock for the receiver, automatic receiver off is switched on, decoder is driven from internal analog circuitry
3Fh	1Fh	00h	ClockQControl	automatic Q-clock calibration is switched on
40h	20h	00h	Page	free for user
41h	21h	08h	RxWait	frame guard time is set to eight bit-clocks
42h	22h	0Ch	ChannelRedundancy	channel redundancy is set using I-CODE1
43h	23h	FEh	CRCPresetLSB	CRC preset value is set using I-CODE1
44h	24h	FFh	CRCPresetMSB	CRC preset value is set using I-CODE1
45h	25h	00h	TimeSlot Period	defines the time for the I-CODE1 time slots
46h	26h	00h	MFOUTSelect	pin MFOUT is set LOW
47h	27h	00h	PreSet27	-
48h	28h	00h	Page	free for user
49h	29h	3Eh	FIFOLevel	WaterLevel[5:0] FIFO buffer warning level is set to standard configuration
4Ah	2Ah	0Bh	TimerClock	TPreScaler[4:0] is set to standard configuration, timer unit restart function is switched off
4Bh	2Bh	02h	TimerControl	Timer is started at the end of transmission, stopped at the beginning of reception
4Ch	2Ch	00h	TimerReload	the timer unit preset value is set to standard configuration
4Dh	2Dh	02h	IRQPinConfig	pin IRQ is set to high-impedance
4Eh	2Eh	00h	PreSet2E	-
4Fh	2Fh	00h	PreSet2F	-

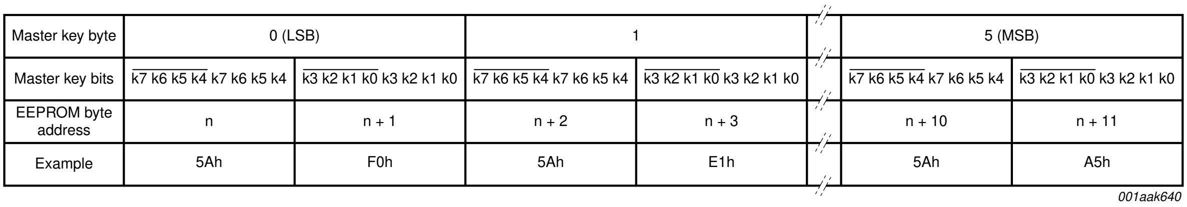
9.2.3 Crypto1 keys (write only)

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 11.7.1 on page 97) and LoadKey commands (see Section 11.7.2 on page 97).

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	$\overline{\text{k7 k6 k5 k4}}$ k7 k6 k5 k4	$\overline{\text{k3 k2 k1 k0}}$ k3 k2 k1 k0	$\overline{\text{k7 k6 k5 k4}}$ k7 k6 k5 k4	$\overline{\text{k3 k2 k1 k0}}$ k3 k2 k1 k0	$\overline{\text{k7 k6 k5 k4}}$ k7 k6 k5 k4	$\overline{\text{k3 k2 k1 k0}}$ 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 11.7.1 on page 97) to fail.

9.2.3.2 Storage of keys in the EEPROM

The CLRC632 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 CLRC632 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 CLRC632. 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 CLRC632 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 CLRC632 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 18 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 CLRC632 indicates interrupt events by setting the PrimaryStatus register bit IRq (see Section 10.5.1.4 "PrimaryStatus register" on page 51) 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 20) 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 on page 20) 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 CLRC632 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 any 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 because 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 CLRC632'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 SetIxx 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 SetIxx 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 29) 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 alphabetic order.

Table 22. Associated Interrupt request system registers and flags

Flags	Register name	Bit	Register address
HiAlertIEn	InterruptEn	1	06h
HiAlertIRq	InterruptRq	1	07h
IdleIEn	InterruptEn	2	06h
IdleIRq	InterruptRq	2	07h
IRq	PrimaryStatus	3	03h
IRQInv	IRQPinConfig	1	07h
IRQPushPull	IRQPinConfig	0	07h
LoAlertIEn	InterruptEn	0	06h
LoAlertIRq	InterruptRq	0	07h
RxIEn	InterruptEn	3	06h
RxIRq	InterruptRq	3	07h
SetIEn	InterruptEn	7	06h
SetIRq	InterruptRq	7	07h
TimerIEn	InterruptEn	5	06h
TimerIRq	InterruptRq	5	07h
TxIEn	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 receiving 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.

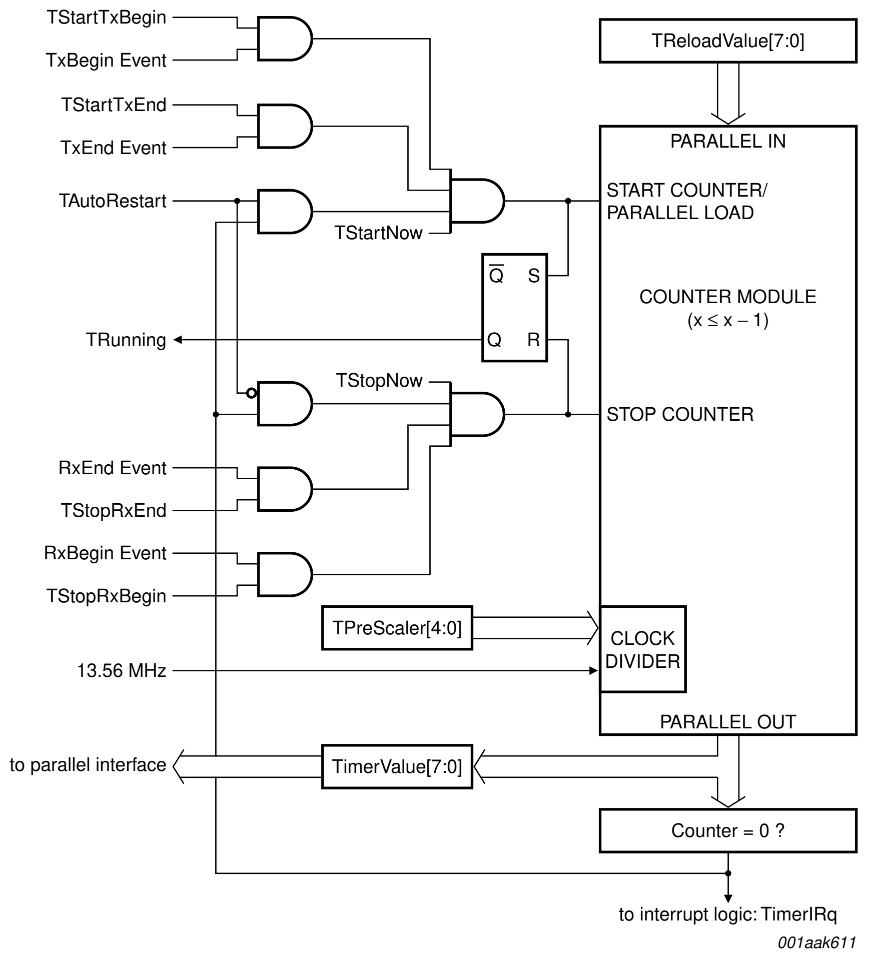


Fig 8. Timer module block diagram

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. Thus, the timer unit is re-triggered.

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:

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

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:

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

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 changes the status flag TRunning to logic 1. Conversely, configured stop events stop the timer and sets 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.