SRIX4K

13.56MHz Short Range Contactless Memory Chip with 4096 bit EEPROM, Anti-Collision and Anti-Clone Functions

FEATURES SUMMARY

- ISO 14443 - 2 Type B Air Interface Compliant
- ISO 14443 - 3 Type B Frame Format Compliant
- 13.56MHz Carrier Frequency
- 847kHz Sub-carrier Frequency
- 106 kbit/second Data Transfer
- France Telecom Proprietary Anti-Clone Function
- 8 bit Chip_ID based anticollision system
- 2 Count-Down Binary Counters with automated anti-tearing protection
- 64-bit Unique Identifier
- 4096-bit EEPROM with Write Protect Feature
- READ BLOCK and WRITE BLOCK (32 Bits)
- Internal Tuning Capacitor
- 1million ERASE/WRITE Cycles
- 40-Year Data Retention
- Self-Timed Programming Cycle
- 5ms Typical Programming Time

Figure 1. Delivery Forms

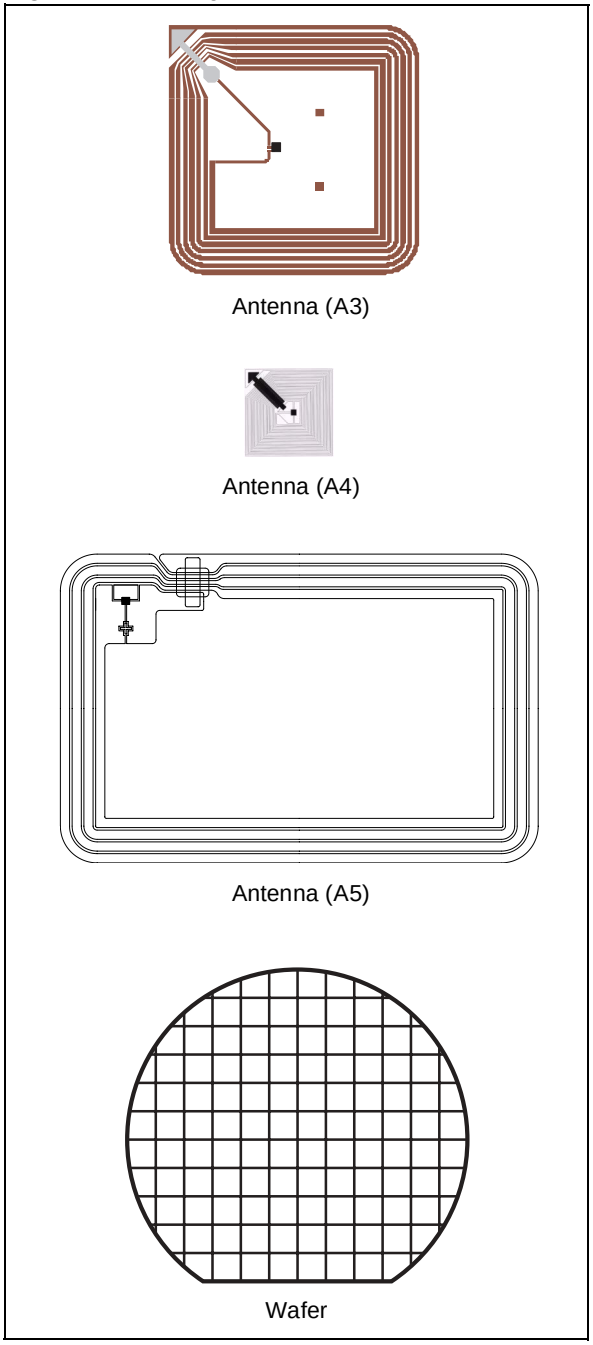


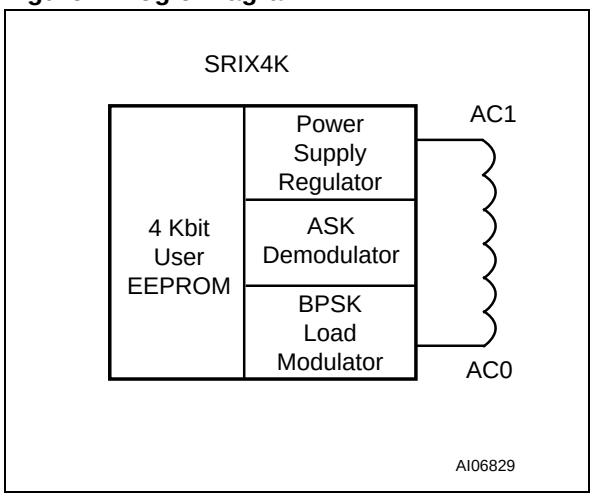
TABLE OF CONTENTS

SUMMARY DESCRIPTION

The SRIX4K is a contactless memory, powered by an externally transmitted radio wave. It contains a 4096-bit user EEPROM fabricated with STMicroelectronics CMOS technology. The memory is organized as 128 blocks of 32 bits. The SRIX4K is accessed via the 13.56MHz carrier. Incoming data are demodulated and decoded from the received Amplitude Shift Keying (ASK) modulation signal and outgoing data are generated by load variation using Bit Phase Shift Keying (BPSK) coding of a 847kHz sub-carrier. The received ASK wave is 10% modulated. The Data transfer rate between the SRIX4K and the reader is 106kbit/s in both reception and emission modes.

The SRIX4K follows the ISO 14443 part 2 type B recommendation for the radio-frequency power and signal interface.

Figure 2. Logic Diagram



The SRIX4K is specifically designed for short range applications that need secure and re-usable products. The SRIX4K includes an anti-collision mechanism that allows it to detect and select tags present at the same time within range of the reader. The anti-collision is based on a probabilistic scanning method using slot markers. The SRIX4K provides an anti-clone function which allows its authentication. Using the STMicroelectronics single chip coupler, CRX14, it is easy to design a reader with the authentication capability and to build a system with a high level of security.

Table 1. Signal Names

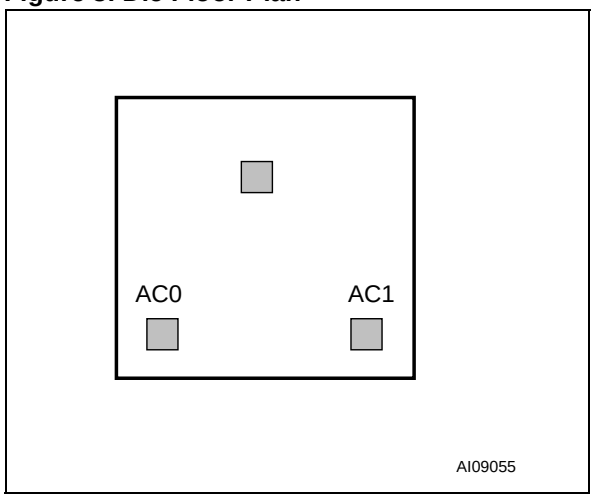
AC1	Antenna Coil
AC0	Antenna Coil

The SRIX4K contactless EEPROM can be randomly read and written in block mode (each block containing 32 bits). The instruction set includes the following ten commands:

- READ_BLOCK
- WRITE_BLOCK
- INITIATE
- PCALL16
- SLOT_MARKER
- SELECT
- COMPLETION
- RESET_TO_INVENTORY
- AUTHENTICATE
- GET_UID

The SRIX4K memory is organized in three areas, as described in Figure 13.. The first area is a resettable OTP (one time programmable) area in which bits can only be switched from 1 to 0. Using a special command, it is possible to erase all bits of this area to 1. The second area provides two 32-bit binary counters which can only be decremented from FFFFFFFFh to 00000000h, and gives a capacity of 4,294,967,296 units per counter. The last area is the EEPROM memory. It is accessible by block of 32 bits and includes an auto-erase cycle during each WRITE_BLOCK command.

Figure 3. Die Floor Plan



SIGNAL DESCRIPTION

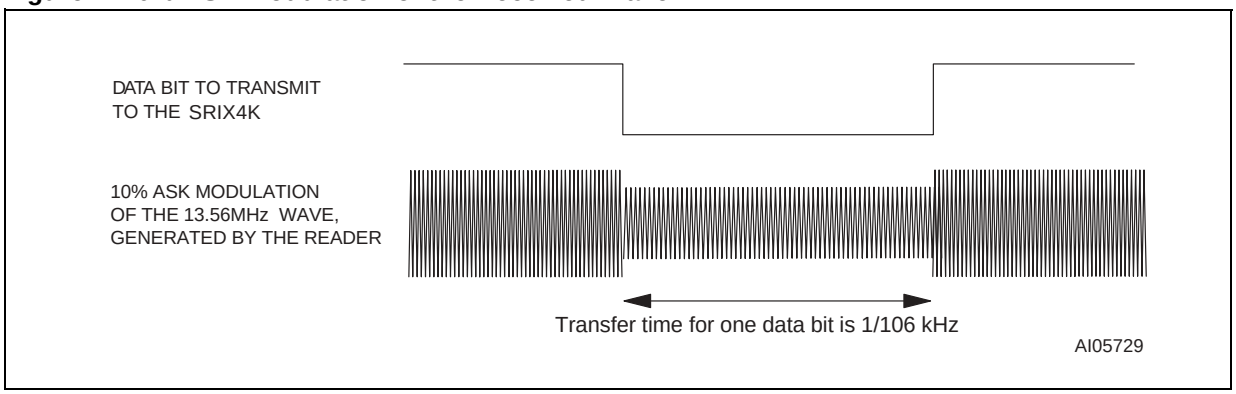
AC1, AC0. The pads for the Antenna Coil. AC1 and AC0 must be directly bonded to the antenna.

DATA TRANSFER

Input Data Transfer from the Reader to the SRIX4K (Request Frame)

The reader must generate a 13.56MHz sinusoidal carrier frequency at its antenna, with enough energy to “remote-power” the memory. The energy received at the SRIX4K’s antenna is transformed into a Supply Voltage by a regulator, and into data bits by the ASK demodulator. For the SRIX4K to decode correctly the information it receives, the reader must 10% amplitude-modulate the 13.56MHz wave before sending it to the SRIX4K. This is represented in Figure 4.. The data transfer rate is 106 kbits/s.

Figure 4. 10% ASK Modulation of the Received Wave



Character Transmission Format for Request Frame. The SRIX4K transmits and receives data bytes as 10-bit characters, with the least significant bit (b_0) transmitted first, as shown in Figure 5.. Each bit duration, an ETU (Elementary Time Unit), is equal to 9.44µs (1/106kHz).

These characters, framed by a Start Of Frame (SOF) and an End Of Frame (EOF), are put together to form a Command Frame as shown in Figure 11.. A frame includes an SOF, commands, addresses, data, a CRC and an EOF as defined in the ISO14443-3 type B Standard. If an error is detected during data transfer, the SRIX4K does not execute the command, but it does not generate an error frame.

Figure 5. SRIX4K Request Frame Character Format

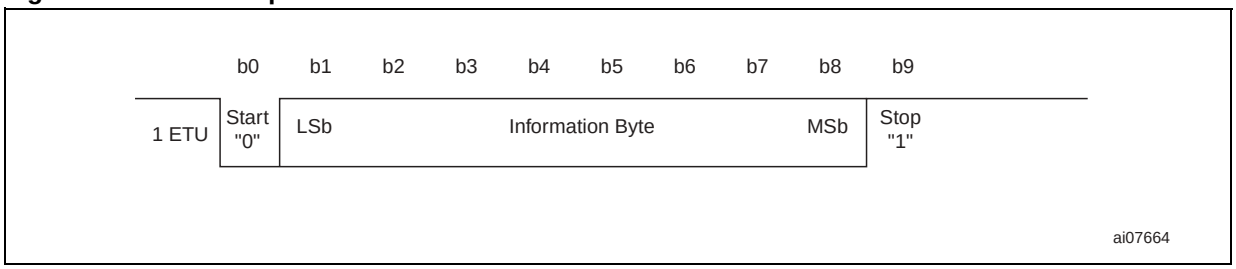


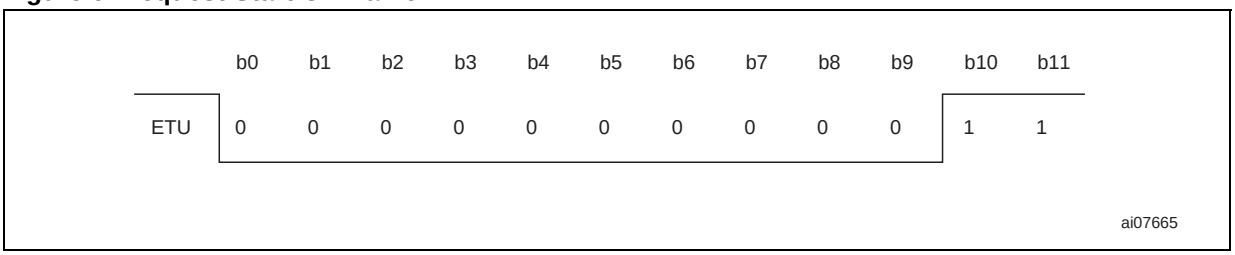
Table 2. Bit Description

Bit	Description	Value
b_0	Start bit used to synchronize the transmission	$b_0 = 0$
b_1 to b_8	Information Byte (command, address or data)	The information byte is sent with the least significant bit first
b_9	Stop bit used to indicate the end of a character	$b_9 = 1$

Request Start Of Frame. The SOF described in Figure 6. is composed of:

- one falling edge,
- followed by 10 ETUs at logic-0,
- followed by a single rising edge,
- followed by at least 2 ETUs (and at most 3) at logic-1.

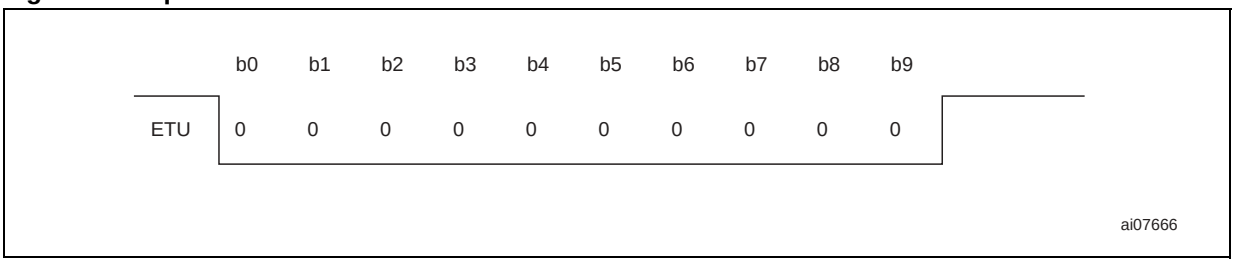
Figure 6. Request Start Of Frame



Request End Of Frame. The EOF shown in Figure 7. is composed of:

- one falling edge,
- followed by 10 ETUs at logic-0,
- followed by a single rising edge.

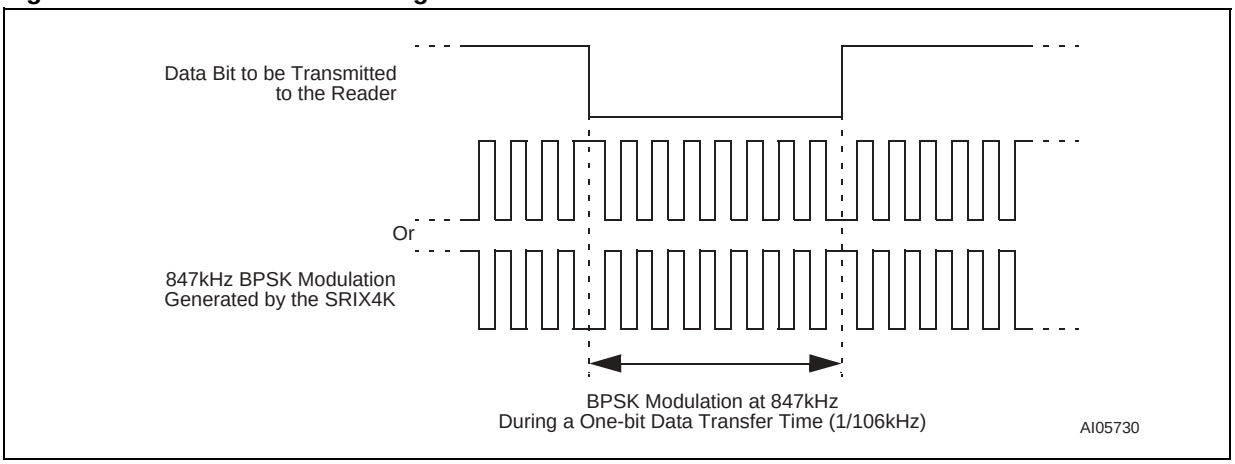
Figure 7. Request End Of Frame



Output Data Transfer from the SRIX4K to the Reader (Answer Frame)

The data bits issued by the SRIX4K use retro-modulation. Retro-modulation is obtained by modifying the SRIX4K current consumption at the antenna (load modulation). The load modulation causes a variation at the reader antenna by inductive coupling. With appropriate detector circuitry, the reader is able to pick up information from the SRIX4K. To improve load-modulation detection, data is transmitted using a BPSK encoded, 847kHz sub-carrier frequency f_S as shown in Figure 8., and as specified in the ISO14443-2 type B Standard.

Figure 8. Wave Transmitted using BPSK Sub-carrier Modulation

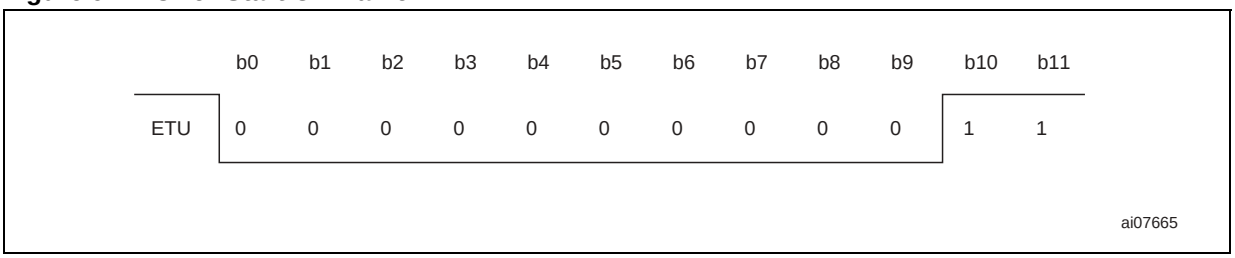


Character Transmission Format for Answer Frame. The character format is the same as for input data transfer (Figure 5.). The transmitted frames are made up of an SOF, data, a CRC and an EOF (Figure 11.). As with an input data transfer, if an error occurs, the reader does not issue an error code to the SRIX4K, but it should be able to detect it and manage the situation. The data transfer rate is 106 kbits/second.

Answer Start Of Frame. The SOF described in Figure 9. is composed of:

– followed by 10 ETUs at logic-0
– followed by 2 ETUs at logic-1

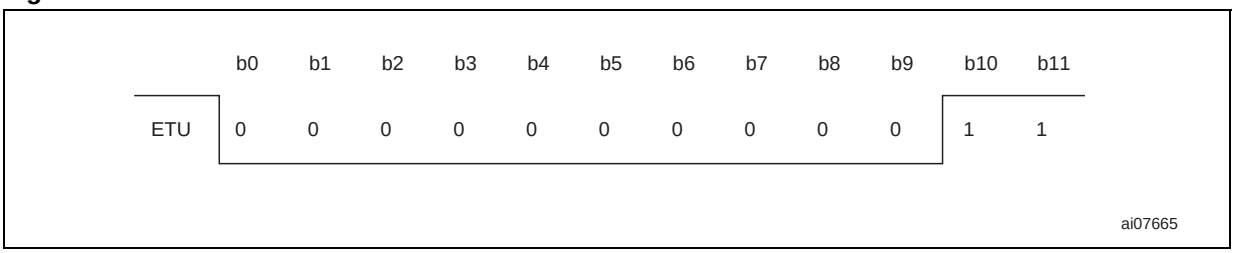
Figure 9. Answer Start Of Frame



Answer End Of Frame. The EOF shown in Figure 10. is composed of:

– followed by 10 ETUs at logic-0,
– followed by 2 ETUs at logic-1.

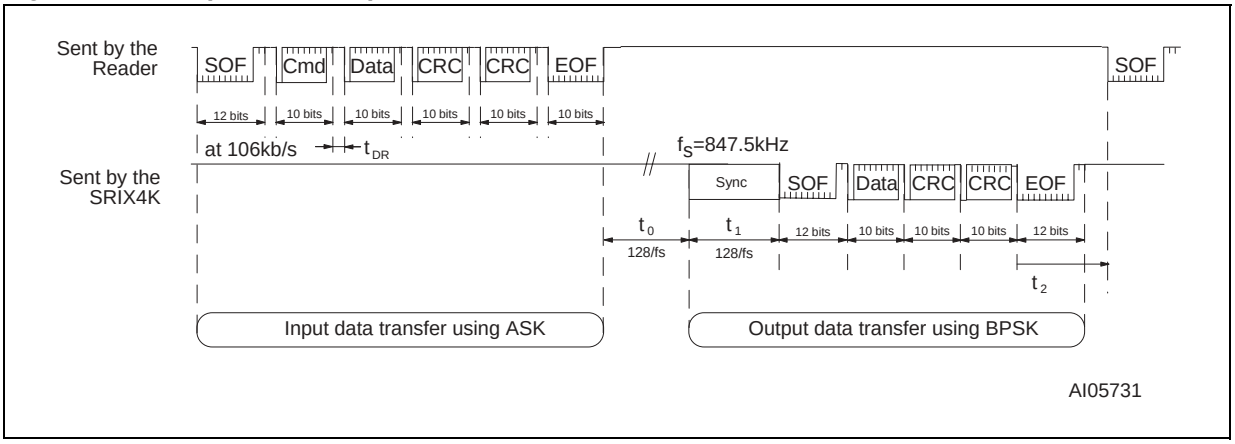
Figure 10. Answer End Of Frame



Transmission Frame

Between the Request data transfer and the Answer data transfer, all ASK and BPSK modulations are suspended for a minimum time of $t_0 = 128/f_S$. This delay allows the reader to switch from Transmission to Reception mode. It is repeated after each frame. After t_0, the 13.56MHz carrier frequency is modulated by the SRIX4K at 847kHz for a period of $t_1 = 128/f_S$ to allow the reader to synchronize. After t_1, the first phase transition generated by the SRIX4K forms the start bit ('0') of the Answer SOF. After the falling edge of the Answer EOF, the reader waits a minimum time, t_2, before sending a new Request Frame to the SRIX4K.

Figure 11. Example of a Complete Transmission Frame



CRC

The 16-bit CRC used by the SRIX4K is generated in compliance with the ISO14443 type B recommendation. For further information, please see APPENDIX A.. The initial register contents are all 1s: FFFFh.

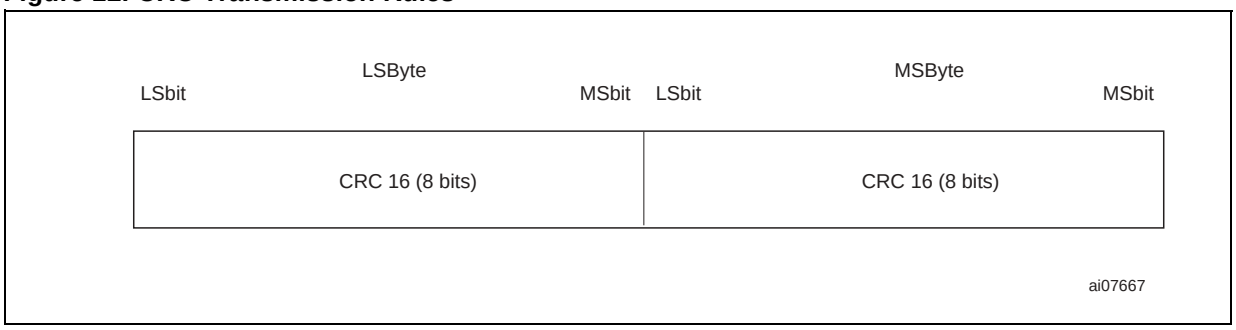
The two-byte CRC is present in every Request and in every Answer Frame, before the EOF. The CRC is calculated on all the bytes between SOF (not included) and the CRC field.

Upon reception of a Request from a reader, the SRIX4K verifies that the CRC value is valid. If it is invalid, the SRIX4K discards the frame and does not answer the reader.

Upon reception of an Answer from the SRIX4K, the reader should verify the validity of the CRC. In case of error, the actions to be taken are the reader designer's responsibility.

The CRC is transmitted with the Least Significant Byte first and each byte is transmitted with the least significant bit first.

Figure 12. CRC Transmission Rules



MEMORY MAPPING

The SRIX4K is organized as 128 blocks of 32 bits as shown in Figure 13.. All blocks are accessible by the READ_BLOCK command. Depending on the write access, they can be updated by the WRITE_BLOCK command. A WRITE_BLOCK updates all the 32 bits of the block.

Figure 13. SRIX4K Memory Mapping

Block Addr	Msb b_{31}	b_{24} b_{23}	32 bits Block b_{16} b_{15}	b_8 b_7	Lsb b_0	Description
0	32 bits Boolean Area					Resettable OTP bits
1	32 bits Boolean Area					
2	32 bits Boolean Area					
3	32 bits Boolean Area					
4	32 bits Boolean Area					
5	32 bits binary counter					Count down Counter
6	32 bits binary counter					
7	User Area					Lockable EEPROM
8	User Area					
9	User Area					
10	User Area					
11	User Area					
12	User Area					
13	User Area					
14	User Area					
15	User Area					
16	User Area					EEPROM
...	User Area					
127	User Area					

255	OTP_Lock_Reg	ST Reserved	Fixed Chip_ID (Option)	System OTP bits

UID0	64 bits UID Area	ROM
UID1		

Resettable OTP Area

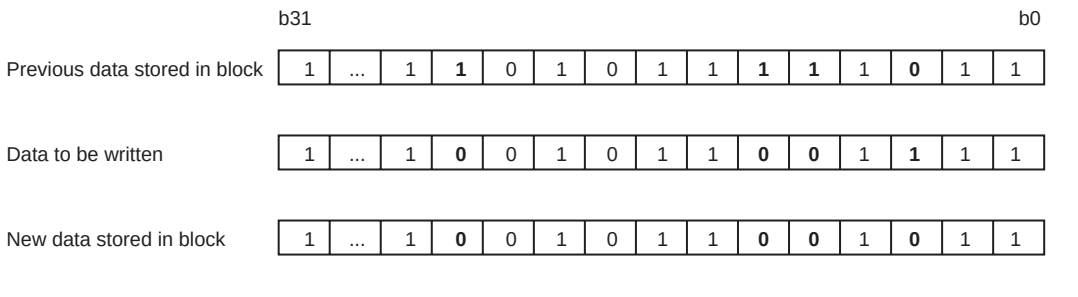
In this area contains five individual 32-bit Boolean Words (see Figure 14. for a map of the area). A WRITE_BLOCK command will not erase the previous contents of the block as the Write cycle is not preceded by an Auto Erase cycle. This feature can be used to reset selected bits from 1 to 0. All bits previously at 0 remain unchanged. When the 32 bits of a block are all at 0, the block is empty, and cannot be updated any more. See Figure 15. and Figure 16. for examples of the result of the WRITE_BLOCK command in the resettable OTP area.

Figure 14. Resettable OTP Area (addresses 0 to 4)

Block Address	MSb b31	b24 b23	32-bit Block b16 b15	b8 b7	LSb b0	Description
0	32-bit Boolean Area					Resettable OTP Bit
1	32-bit Boolean Area					
2	32-bit Boolean Area					
3	32-bit Boolean Area					
4	32-bit Boolean Area					

ai07657

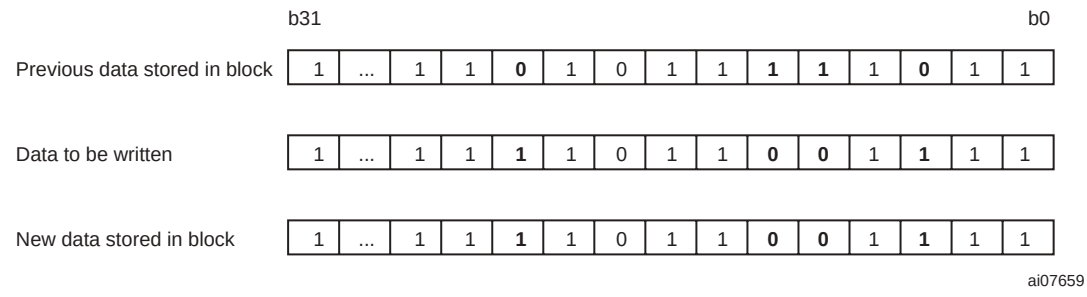
Figure 15. WRITE_BLOCK Update in Standard Mode (Binary Format)



ai07658

The five 32-bit blocks making up the Resettable OTP area can be erased in one go by adding an Auto Erase cycle to the WRITE_BLOCK command. An Auto Erase cycle is added each time the SRIX4K detects a Reload command. The Reload command is implemented through a specific update of the 32-bit binary counter located at block address 6 (see "32-bit Binary Counters" for details).

Figure 16. WRITE_BLOCK Update in Reload Mode (Binary Format)



ai07659

32-bit Binary Counters

The two 32-bit binary counters located at block addresses 5 and 6, respectively, are used to count down from 2^{32} (4096 million) to 0. The SRIX4K uses dedicated logic that only allows the update of a counter if the new value is lower than the previous one. This feature allows the application to count down by steps of 1 or more. The initial value in the counter is FFFFFFFFh. When the value displayed is 00000000h, the counter is empty and cannot be reloaded. The counter is updated by issuing the WRITE_BLOCK command to block address 5 or 6, depending on which counter is to be updated. The WRITE_BLOCK command writes the new 32-bit value to the counter block address. Figure 18. shows examples of how the counters operate.

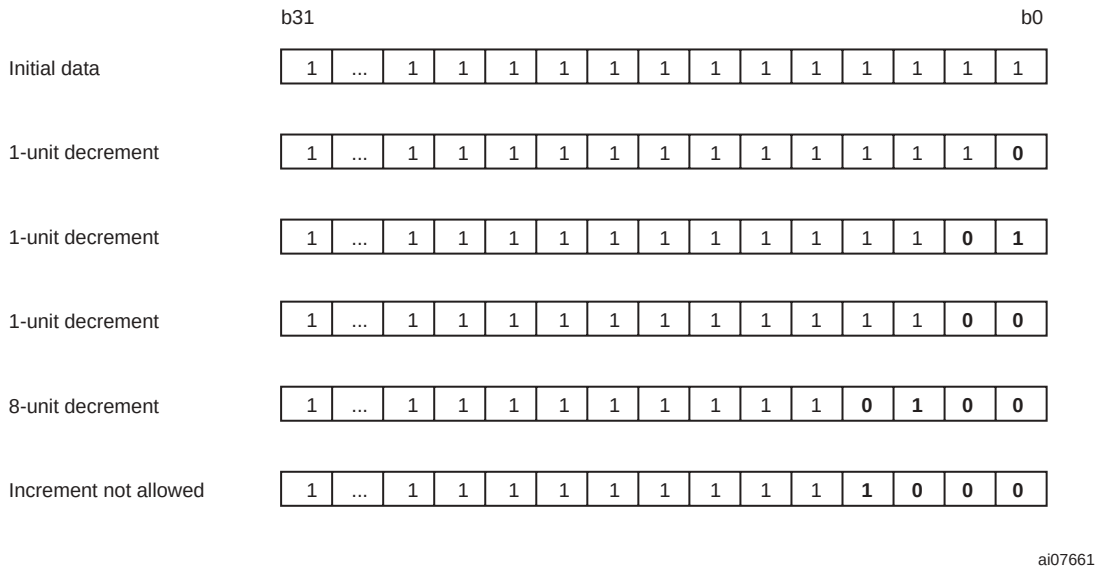
The counter programming cycles are protected by automated anti-tearing logic. This function allows the counter value to be protected in case of power down within the programming cycle. In case of power down, the counter value is not updated and the previous value continues to be stored.

Figure 17. Binary Counter (addresses 5 to 6)

Block Address	MSb b31	b24 b23	32-bit Block b16 b15	b8 b7	LSb b0	Description
5	32-bit Binary Counter					Count down Counter
6	32-bit Binary Counter					

ai07660

Figure 18. Count Down Example (Binary Format)



ai07661

The counter with block address 6 controls the Reload command used to reset the resettable OTP area (addresses 0 to 4). Bits b_{31} to b_{21} act as an 11-bit Reload counter; whenever one of these 11 bits is updated, the SRIX4K detects the change and adds an Erase cycle to the WRITE_BLOCK command for locations 0 to 4 (see the "Resettable OTP Area" paragraph). The Erase cycle remains active until a POWER-OFF or a SELECT command is issued. The SRIX4K's resettable OTP area can be reloaded up to 2,047 times (2^{11}-1).

EEPROM Area

The 121 blocks between addresses 7 and 127 are EEPROM blocks of 32 bits each (484 Bytes in total). (See Figure 19. for a map of the area.) These blocks can be accessed using the READ_BLOCK and WRITE_BLOCK commands. The WRITE_BLOCK command for the EEPROM area always includes an Auto-Erase cycle prior to the Write cycle.

Blocks 7 to 15 can be Write-protected. Write access is controlled by the 8 bits of the OTP_Lock_Reg located at block address 255 (see "OTP_Lock_Reg" for details). Once protected, these blocks (7 to 15) cannot be unprotected.

Figure 19. EEPROM (Addresses 7 to 127)

Block Address	MSb b31 b24 b23 32-bit Block b16 b15 b8 b7 LSb b0	Description
7	User Area	Lockable EEPROM
8	User Area	
9	User Area	
10	User Area	
11	User Area	
12	User Area	
13	User Area	
14	User Area	
15	User Area	
16	User Area	EEPROM
...	User Area	
127	User Area	

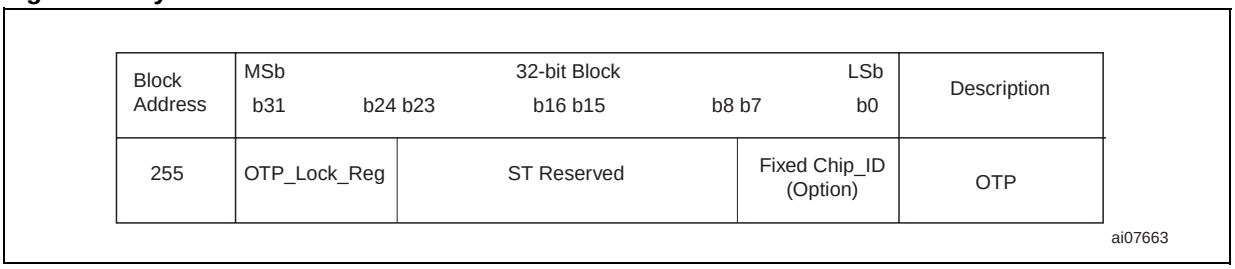
Ai07661B

System Area

This area is used to modify the settings of the SRIX4K. It contains 3 registers: OTP_Lock_Reg, Fixed Chip_ID and ST Reserved. See Figure 20. for a map of this area.

A WRITE_BLOCK command in this area will not erase the previous contents. Selected bits can thus be set from 1 to 0. All bits previously at 0 remain unchanged. Once all the 32 bits of a block are at 0, the block is empty and cannot be updated any more.

Figure 20. System Area



OTP_Lock_Reg. The 8 bits, b_{31} to b_{24}, of the System Area (block address 255) are used as OTP_Lock_Reg bits in the SRIX4K. They control the Write access to the 9 EEPROM blocks with addresses 7 to 15 as follows:

- When b_{24} is at 0, blocks 7 and 8 are Write-protected
- When b_{25} is at 0, block 9 is Write-protected
- When b_{26} is at 0, block 10 is Write-protected
- When b_{27} is at 0, block 11 is Write-protected
- When b_{28} is at 0, block 12 is Write-protected
- When b_{29} is at 0, block 13 is Write-protected
- When b_{30} is at 0, block 14 is Write-protected
- When b_{31} is at 0, block 15 is Write-protected.

The OTP_Lock_Reg bits cannot be erased. Once Write-protected, EEPROM blocks behave like ROM blocks and cannot be unprotected.

Fixed Chip_ID (Option). The SRIX4K is provided with an anti-collision feature based on a random 8-bit Chip_ID. Prior to selecting an SRIX4K, an anti-collision sequence has to be run to search for the Chip_ID of the SRIX4K. This is a very flexible feature, however the searching loop requires time to run.

For some applications, much time could be saved by knowing the value of the SRIX4K Chip_ID beforehand, so that the SRIX4K can be identified and selected directly without having to run an anti-collision sequence. This is why the SRIX4K was designed with an optional mask setting used to program a fixed 8-bit Chip_ID to bits b_7 to b_0 of the system area. When the fixed Chip_ID option is used, the random Chip_ID function is disabled.

SRIX4K OPERATION

All commands, data and CRC are transmitted to the SRIX4K as 10-bit characters using ASK modulation. The start bit of the 10 bits, b_0, is sent first. The command frame received by the SRIX4K at the antenna is demodulated by the 10% ASK demodulator, and decoded by the internal logic. Prior to any operation, the SRIX4K must have been selected by a SELECT command. Each frame transmitted to the SRIX4K must start with a Start Of Frame, followed by one or more data characters, two CRC Bytes and the final End Of Frame. When an invalid frame is decoded by the SRIX4K (wrong command or CRC error), the memory does not return any error code.

When a valid frame is received, the SRIX4K may have to return data to the reader. In this case, data is returned using BPSK encoding, in the form of 10-bit characters framed by an SOF and an EOF. The transfer is ended by the SRIX4K sending the 2 CRC Bytes and the EOF.

SRIX4K STATES

The SRIX4K can be switched into different states. Depending on the current state of the SRIX4K, its logic will only answer to specific commands. These states are mainly used during the anti-collision sequence, to identify and to access the SRIX4K in a very short time. The SRIX4K provides 6 different states, as described in the following paragraphs and in Figure 21..

POWER-OFF State

The SRIX4K is in POWER-OFF state when the electromagnetic field around the tag is not strong enough. In this state, the SRIX4K does not respond to any command.

READY State

When the electromagnetic field is strong enough, the SRIX4K enters the READY state. After Power-up, the Chip_ID is initialized with a random value. The whole logic is reset and remains in this state until an INITIATE() command is issued. Any other command will be ignored by the SRIX4K.

INVENTORY State

The SRIX4K switches from the READY to the INVENTORY state after an INITIATE() command has been issued. In INVENTORY state, the SRIX4K will respond to any anti-collision commands: INITIATE(), PCALL16() and SLOT_MARKER(), and then remain in the INVENTORY state. It will switch to the SELECTED state after a SELECT(Chip_ID) command is issued, if the Chip_ID in the command matches its own. If not, it will remain in INVENTORY state.

SELECTED State

In SELECTED state, the SRIX4K is active and responds to all READ_BLOCK(), WRITE_BLOCK(), AUTHENTICATE() and GET_UID() commands. When an SRIX4K has entered the SELECTED state, it no longer responds to anti-collision commands. So that the reader can access another tag, the SRIX4K can be switched to the DESELECTED state by sending a SELECT(Chip_ID2) with a Chip_ID that does not match its own, or it can be placed in DEACTIVATED state by issuing a COMPLETION() command. Only one SRIX4K can be in SELECTED state at a time.

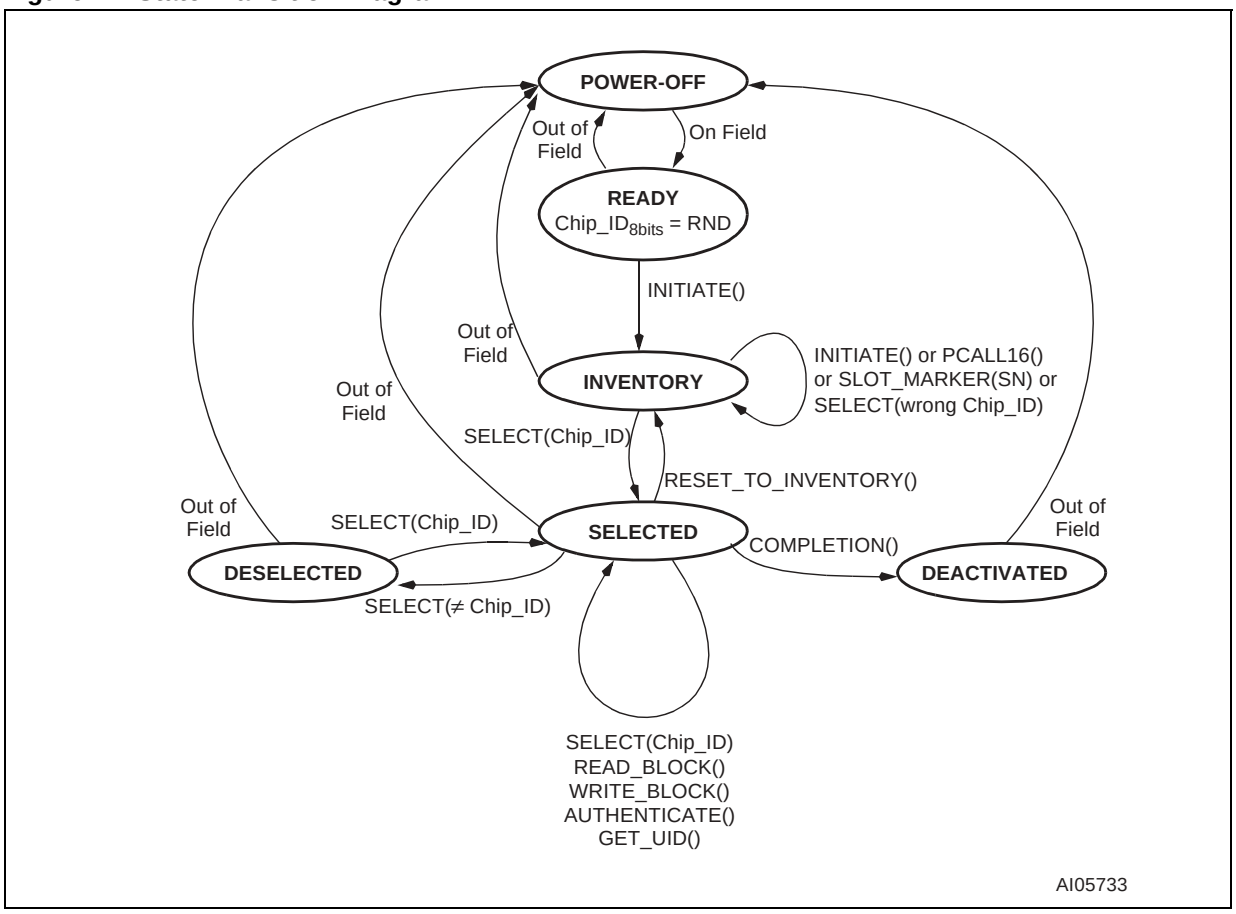
DESELECTED State

Once the SRIX4K is in DESELECTED state, only a SELECT(Chip_ID) command with a Chip_ID matching its own can switch it back to SELECTED state. All other commands are ignored.

DEACTIVATED State

When in this state, the SRIX4K can only be turned off. All commands are ignored.

Figure 21. State Transition Diagram



ANTI-COLLISION

The SRIX4K provides an anti-collision mechanism that searches for the Chip_ID of each device that is present in the reader field range. When known, the Chip_ID is used to select an SRIX4K individually, and access its memory. The anti-collision sequence is managed by the reader through a set of commands described in the "SRIX4K OPERATION" section:

- INITIATE()
- PCALL16()
- SLOT_MARKER().

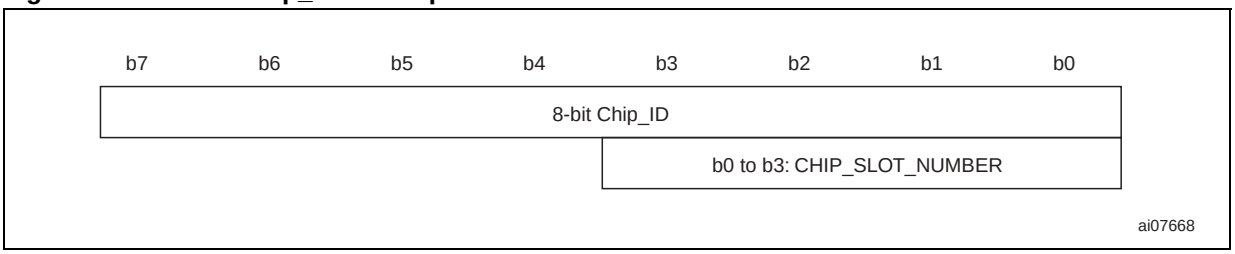
The reader is the master of the communication with one or more SRIX4K device(s). It initiates the tag communication activity by issuing an INITIATE(), PCALL16() or SLOT_MARKER() command to prompt the SRIX4K to answer. During the anti-collision sequence, it might happen that two or more SRIX4K devices respond simultaneously, so causing a collision. The command set allows the reader to handle the sequence, to separate SRIX4K transmissions into different time slots. Once the anti-collision sequence has completed, SRIX4K communication is fully under the control of the reader, allowing only one SRIX4K to transmit at a time.

The Anti-collision scheme is based on the definition of time slots during which the SRIX4K devices are invited to answer with minimum identification data: the Chip_ID. The number of slots is fixed at 16 for the PCALL16() command. For the INITIATE() command, there is no slot and the SRIX4K answers after the command is issued. SRIX4K devices are allowed to answer only once during the anticollision sequence. Consequently, even if there are several SRIX4K devices present in the reader field, there will probably be a slot in which only one SRIX4K answers, allowing the reader to capture its Chip_ID. Using the Chip_ID, the reader can then establish a communication channel with the identified SRIX4K. The purpose of the anti-collision sequence is to allow the reader to select one SRIX4K at a time.

The SRIX4K is given an 8-bit Chip_ID value used by the reader to select only one among up to 256 tags present within its field range. The Chip_ID is initialized with a random value during the READY state, or after an INITIATE() command in the INVENTORY state.

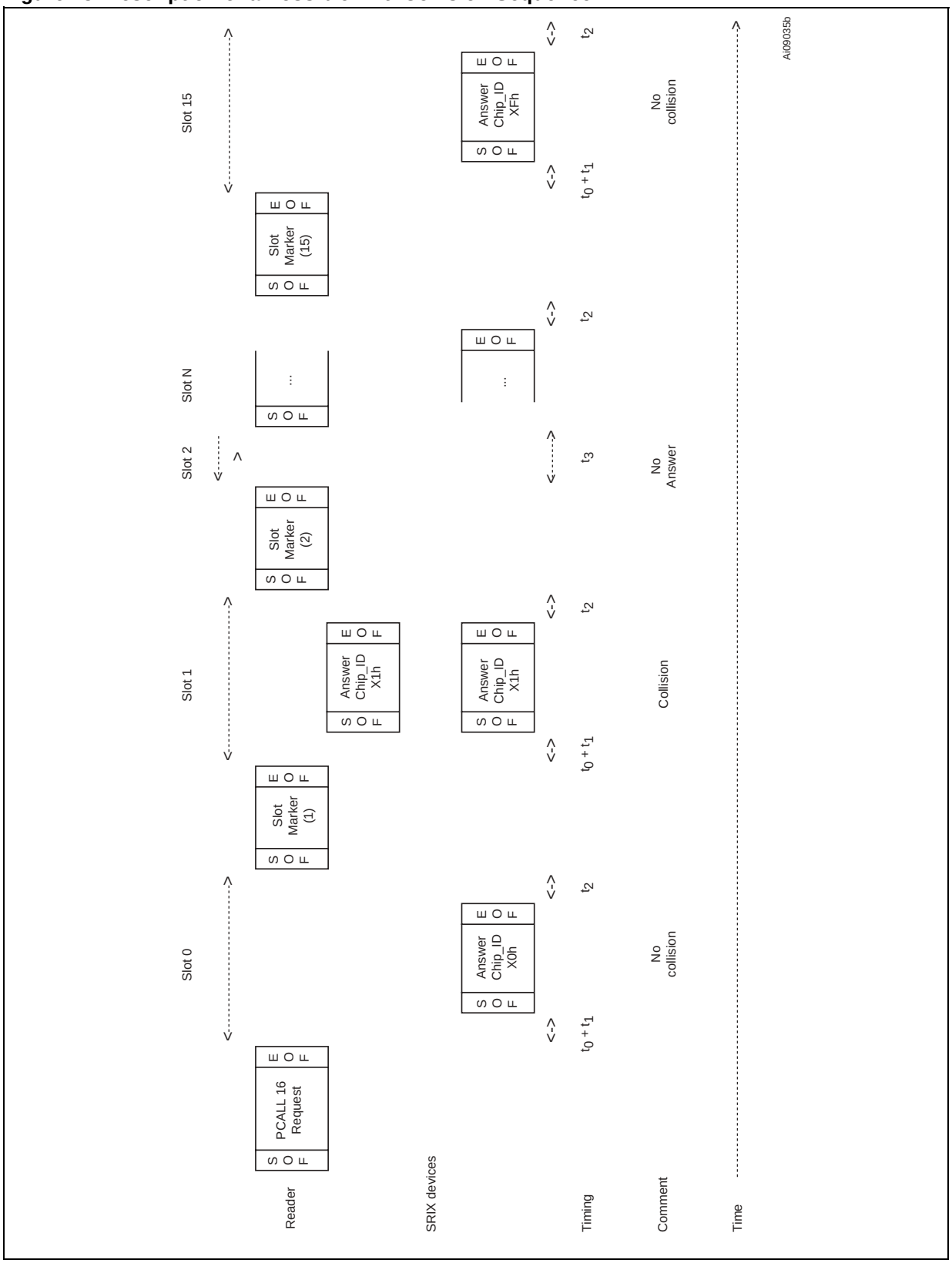
The four least significant bits (b_0 to b_3) of the Chip_ID are also known as the CHIP_SLOT_NUMBER. This 4-bit value is used by the PCALL16() and SLOT_MARKER() commands during the anti-collision sequence in the INVENTORY state.

Figure 22. SRIX4K Chip_ID Description



Each time the SRIX4K receives a PCALL16() command, the CHIP_SLOT_NUMBER is given a new 4-bit random value. If the new value is 0000_b, the SRIX4K returns its whole 8-bit Chip_ID in its answer to the PCALL16() command. The PCALL16() command is also used to define the slot number 0 of the anti-collision sequence. When the SRIX4K receives the SLOT_MARKER(SN) command, it compares its CHIP_SLOT_NUMBER with the SLOT_NUMBER parameter (SN). If they match, the SRIX4K returns its Chip_ID as a response to the command. If they do not, the SRIX4K does not answer. The SLOT_MARKER(SN) command is used to define all the anti-collision slot numbers from 1 to 15.

Figure 23. Description of a Possible Anti-Collision Sequence



Note: The value X in the Answer Chip_ID means a random hexadecimal character from 0 to F.

Description of an Anti-Collision Sequence

The anti-collision sequence is initiated by the INITIATE() command which triggers all the SRIX4K devices that are present in the reader field range, and that are in INVENTORY state. Only SRIX4K devices in INVENTORY state will respond to the PCALL16() and SLOT_MARKER(SN) anti-collision commands.

A new SRIX4K introduced in the field range during the anti-collision sequence will not be taken into account as it will not respond to the PCALL16() or SLOT_MARKER(SN) command (READY state). To be considered during the anti-collision sequence, it must have received the INITIATE() command and entered the INVENTORY state.

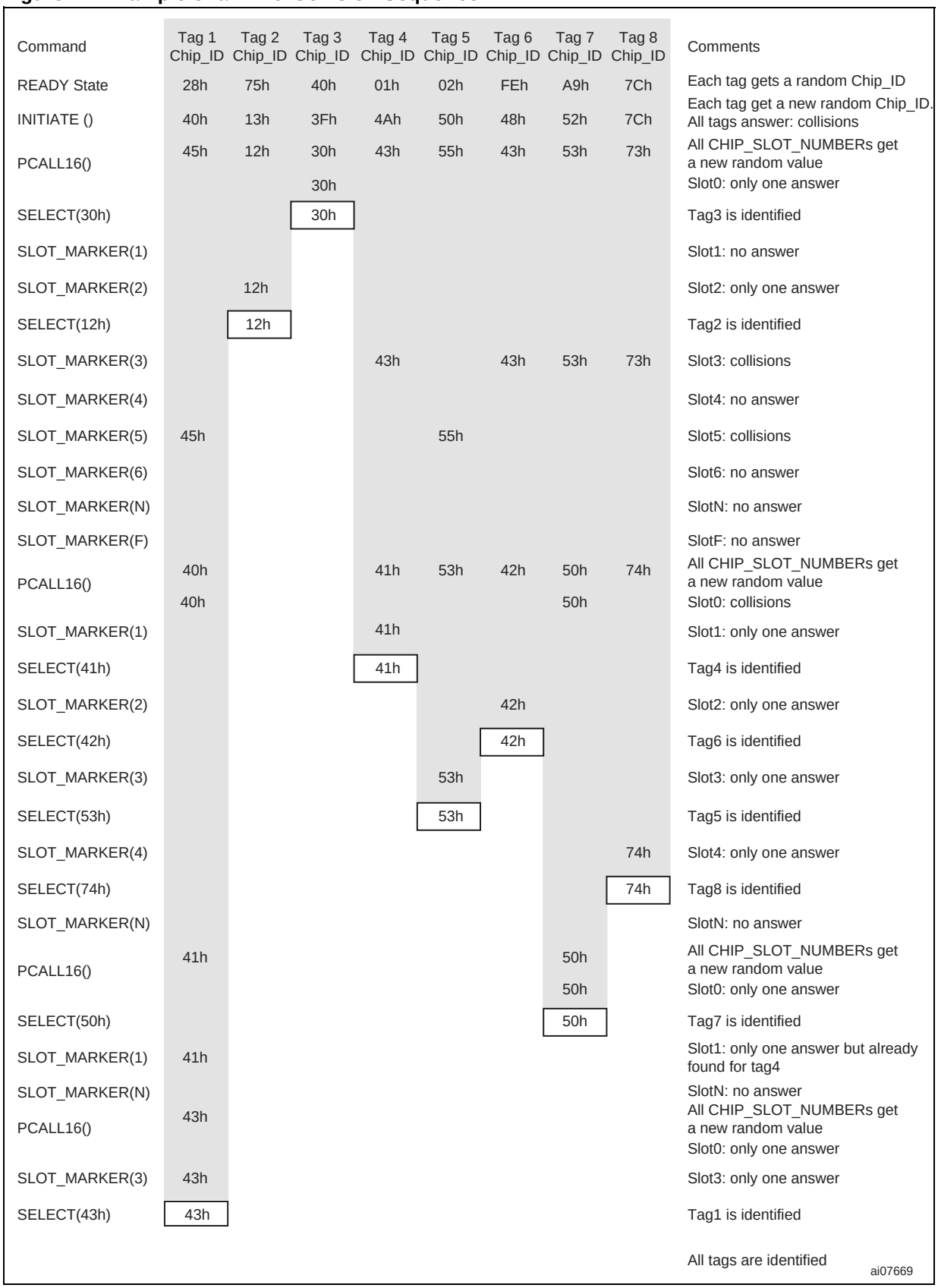
Table 3. shows the elements of a standard anti-collision sequence. (See Figure 24. for an example.)

Table 3. Standard Anti-Collision Sequence

Step 1	Init:	Send INITIATE(). – If no answer is detected, go to step1. – If only 1 answer is detected, select and access the SRIX4K. After accessing the SRIX4K, deselect the tag and go to step1. – If a collision (many answers) is detected, go to step2.
Step 2	Slot 0	Send PCALL16(). – If no answer or collision is detected, go to step3. – If 1 answer is detected, store the Chip_ID, Send SELECT() and go to step3.
Step 3	Slot 1	Send SLOT_MARKER(1). – If no answer or collision is detected, go to step4. – If 1 answer is detected, store the Chip_ID, Send SELECT() and go to step4.
Step 4	Slot 2	Send SLOT_MARKER(2). – If no answer or collision is detected, go to step5. – If 1 answer is detected, store the Chip_ID, Send SELECT() and go to step5.
Step N	Slop N	Send SLOT_MARKER(3 up to 14) ... – If no answer or collision is detected, go to stepN+1. – If 1 answer is detected, store the Chip_ID, Send SELECT() and go to stepN+1.
Step 17	Slot 15	Send SLOT_MARKER(15). – If no answer or collision is detected, go to step18. – If 1 answer is detected, store the Chip_ID, Send SELECT() and go to step18.
Step 18		All the slots have been generated and the Chip_ID values should be stored into the reader memory. Issue the SELECT(Chip_ID) command and access each identified SRIX4K one by one. After accessing each SRIX4K, switch them into DESELECTED or DEACTIVATED state, depending on the application needs. – If collisions were detected between Step2 and Step17, go to Step2. – If no collision was detected between Step2 and Step17, go to Step1.

After each SLOT_MARKER() command, there may be several, one or no answers from the SRIX4K devices. The reader must handle all the cases and store all the Chip_IDs, correctly decoded. At the end of the anti-collision sequence, after SLOT_MARKER(15), the reader can start working with one SRIX4K by issuing a SELECT() command containing the desired Chip_ID. If a collision is detected during the anti-collision sequence, the reader has to generate a new sequence in order to identify all unidentified SRIX4K devices in the field. The anti-collision sequence can stop when all SRIX4K devices have been identified.

Figure 24. Example of an Anti-Collision Sequence



ANTI-CLONE FUNCTION

The SRIX4K provides an anti-clone function that allows the application to authentication the device. This function uses reserved data that is stored in the SRIX4K memory at its time of manufacture.

The Authentication system is based on a proprietary challenge/response mechanism which allows the application software to authenticate any member of the secure memory tag SRXxxx family from STMicroelectronics (of which the SRIX4K is the prime example). A reader system, based on the ST CRX14 chip coupler, can check each SRIX4K tag for authenticity, and protect the application system against silicon copies or emulators.

A complete description of the Authentication system is available under Non Disclosure Agreement (NDA) with STMicroelectronics. For more details about this SRIX4K function, please contact your nearest STMicroelectronics sales office.

SRIX4K COMMANDS

See the paragraphs below for a detailed description of the Commands available on the SRIX4K. The commands and their hexadecimal codes are summarized in Table 4.. A brief is given in APPENDIX B..

Table 4. Command Code

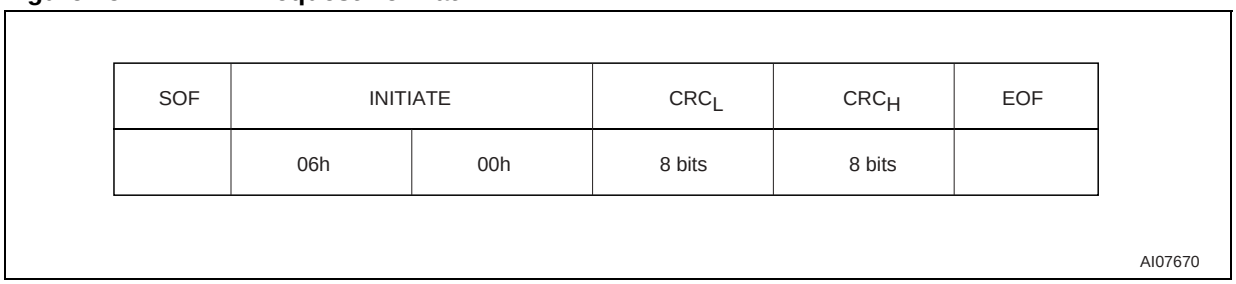
Hexadecimal Code	Command
06h-00h	INITIATE()
06h-04h	PCALL16()
x6h	SLOT_MARKER (SN)
08h	READ_BLOCK(Addr)
09h	WRITE_BLOCK(Addr, Data)
0Ah	AUTHENTICATE(RND)
0Bh	GET_UID()
0Ch	RESET_TO_INVENTORY
0Eh	SELECT(Chip_ID)
0Fh	COMPLETION()

INITIATE() Command

Command Code = 06h - 00h

INITIATE() is used to initiate the anti-collision sequence of the SRIX4K. On receiving the INITIATE() command, all SRIX4K devices in READY state switch to INVENTORY state, set a new 8-bit Chip_ID random value, and return their Chip_ID value. This command is useful when only one SRIX4K in READY state is present in the reader field range. It speeds up the Chip_ID search process. The CHIP_SLOT_NUMBER is not used during INITIATE() command access.

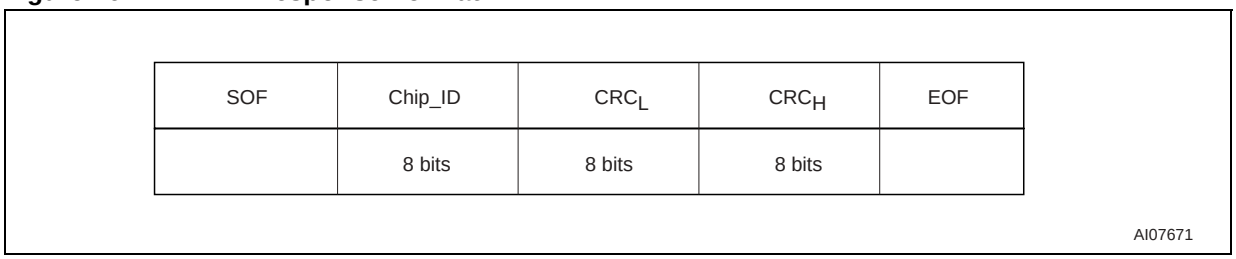
Figure 25. INITIATE Request Format



Request parameter:

– No parameter

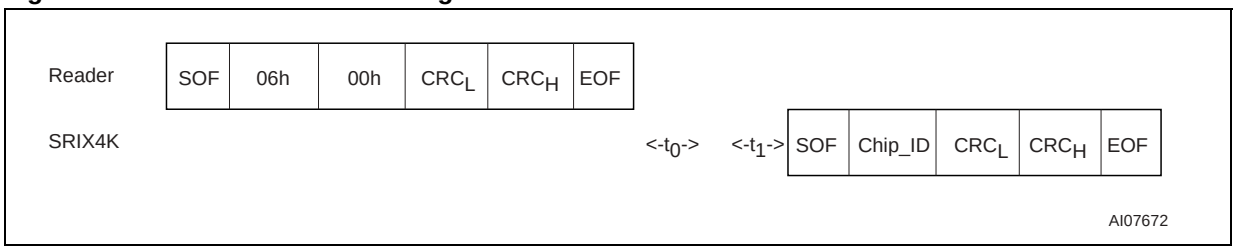
Figure 26. INITIATE Response Format



Response parameter:

– Chip_ID of the SRIX4K

Figure 27. INITIATE Frame Exchange Between Reader and SRIX4K



PCALL16() Command

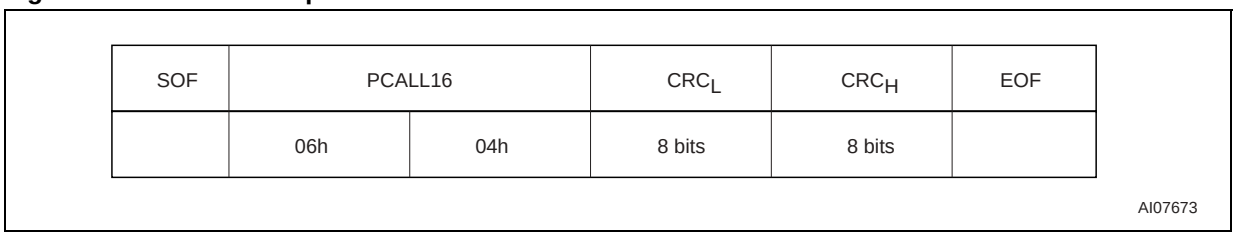
Command Code = 06h - 04h

The SRIX4K must be in INVENTORY state to interpret the PCALL16() command.

On receiving the PCALL16() command, the SRIX4K first generates a new random CHIP_SLOT_NUMBER value (in the 4 least significant bits of the Chip_ID). CHIP_SLOT_NUMBER can take on a value between 0 an 15 (1111_b). The value is retained until a new PCALL16() or INITIATE() command is issued, or until the SRIX4K is powered off. The new CHIP_SLOT_NUMBER value is then compared with the value 0000_b. If they match, the SRIX4K returns its Chip_ID value. If not, the SRIX4K does not send any response.

The PCALL16() command, used together with the SLOT_MARKER() command, allows the reader to search for all the Chip_IDs when there are more than one SRIX4K device in INVENTORY state present in the reader field range.

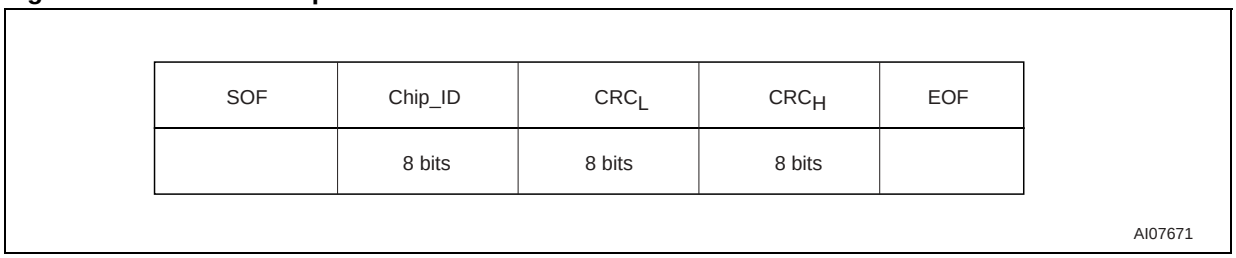
Figure 28. PCALL16 Request Format



Request parameter:

– No parameter

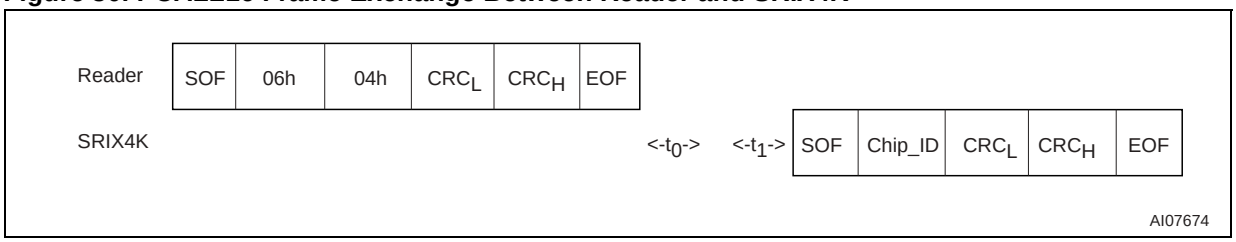
Figure 29. PCALL16 Response Format



Response parameter:

– Chip_ID of the SRIX4K

Figure 30. PCALL16 Frame Exchange Between Reader and SRIX4K



SLOT_MARKER(SN) Command

Command Code = x6h

The SRIX4K must be in INVENTORY state to interpret the SLOT_MARKER(SN) command.

The SLOT_MARKER Byte code is divided into two parts:

– b_3 to b_0: 4-bit command code with fixed value 6.

– b_7 to b_4: 4 bits known as the SLOT_NUMBER (SN). They assume a value between 1 and 15. The value 0 is reserved by the PCALL16() command.

On receiving the SLOT_MARKER() command, the SRIX4K compares its CHIP_SLOT_NUMBER value with the SLOT_NUMBER value given in the command code. If they match, the SRIX4K returns its Chip_ID value. If not, the SRIX4K does not send any response.

The SLOT_MARKER() command, used together with the PCALL16() command, allows the reader to search for all the Chip_IDs when there are more than one SRIX4K device in INVENTORY state present in the reader field range.

Figure 31. SLOT_MARKER Request Format

SOF	SLOT_MARKER	CRC_L	CRC_H	EOF
	X6h	8 bits	8 bits	

AI07675

Request parameter:

– x: Slot number

Figure 32. SLOT_MARKER Response Format

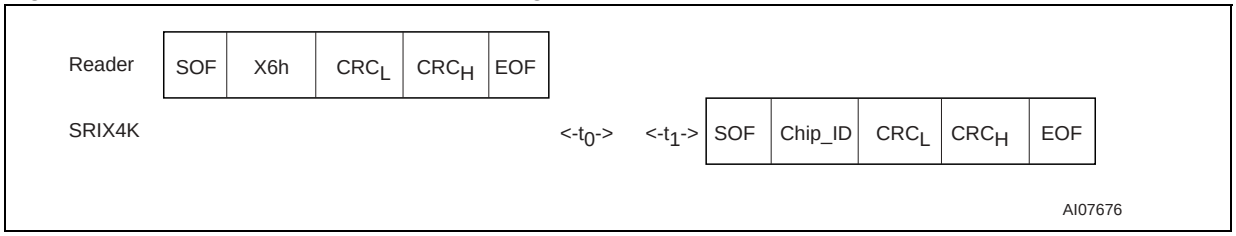
SOF	Chip_ID	CRC_L	CRC_H	EOF
	8 bits	8 bits	8 bits	

AI07671

Response parameters:

– Chip_ID of the SRIX4K

Figure 33. SLOT_MARKER Frame Exchange Between Reader and SRIX4K



AI07676

SELECT(Chip_ID) Command

Command Code = 0Eh

The SELECT() command allows the SRIX4K to enter the SELECTED state. Until this command is issued, the SRIX4K will not accept any other command, except for INITIATE(), PCALL16() and SLOT_MARKER(). The SELECT() command returns the 8 bits of the Chip_ID value. An SRIX4K in SELECTED state, that receives a SELECT() command with a Chip_ID that does not match its own is automatically switched to DESELECTED state.

Figure 34. SELECT Request Format

SOF	SELECT	Chip_ID	CRC_L	CRC_H	EOF
	0Eh	8 bits	8 bits	8 bits	

AI07677

Request parameter:

– 8-bit Chip_ID stored during the anti-collision sequence

Figure 35. SELECT Response Format

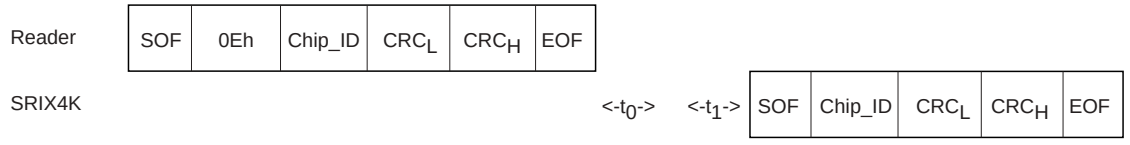
SOF	Chip_ID	CRC_L	CRC_H	EOF
	8 bits	8 bits	8 bits	

AI07671

Response parameters:

– Chip_ID of the selected tag. Must be equal to the transmitted Chip_ID

Figure 36. SELECT Frame Exchange Between Reader and SRIX4K



AI07678

COMPLETION() Command

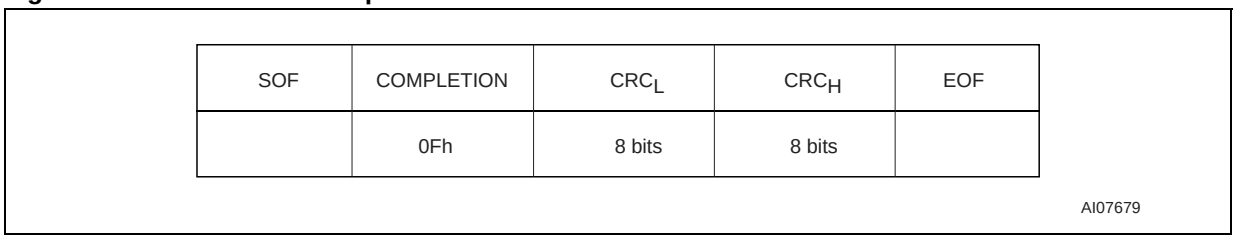
Command Code = 0Fh

On receiving the COMPLETION() command, a SRIX4K in SELECTED state switches to DEACTIVATED state and stops decoding any new commands. The SRIX4K is then locked in this state until a complete reset (tag out of the field range).

A new SRIX4K can thus be accessed through a SELECT() command without having to remove the previous one from the field. The COMPLETION() command does not generate a response.

All SRIX4K devices not in SELECTED state ignore the COMPLETION() command.

Figure 37. COMPLETION Request Format



Request parameters:

– No parameter

Figure 38. COMPLETION Response Format

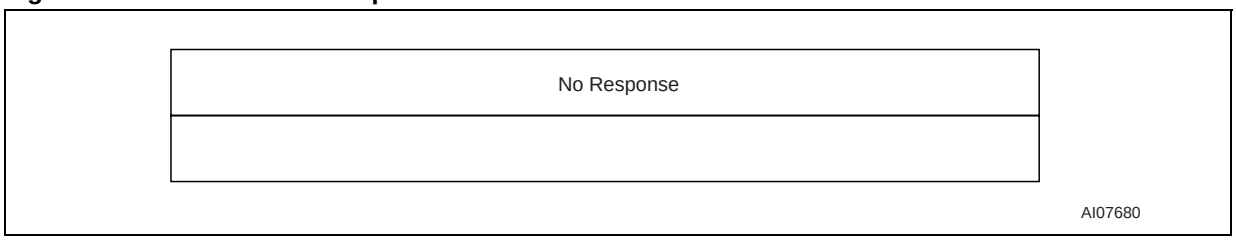
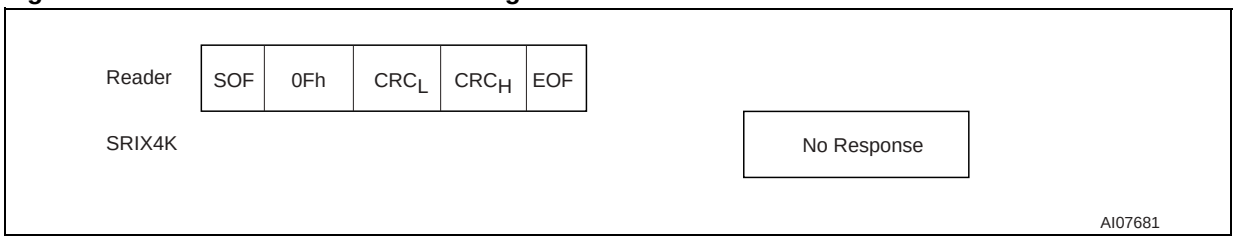


Figure 39. COMPLETION Frame Exchange Between Reader and SRIX4K



RESET_TO_INVENTORY() Command

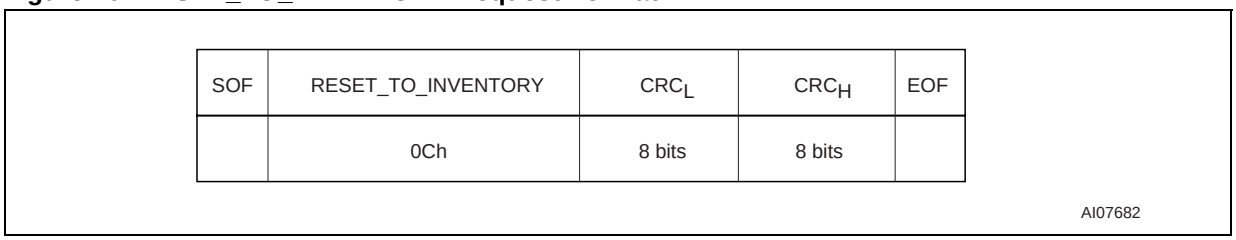
Command Code = 0Ch

On receiving the RESET_TO_INVENTORY() command, all SRIX4K devices in SELECTED state revert to INVENTORY state. The concerned SRIX4K devices are thus resubmitted to the anti-collision sequence. This command is useful when two SRIX4K devices with the same 8-bit Chip_ID happen to be in SELECTED state at the same time. Forcing them to go through the anti-collision sequence again allows the reader to generates new PCALL16() commands and so, to set new random Chip_IDs.

The RESET_TO_INVENTORY() command does not generate a response.

All SRIX4K devices that are not in SELECTED state ignore the RESET_TO_INVENTORY() command.

Figure 40. RESET_TO_INVENTORY Request Format



Request parameter:

– No parameter

Figure 41. RESET_TO_INVENTORY Response Format

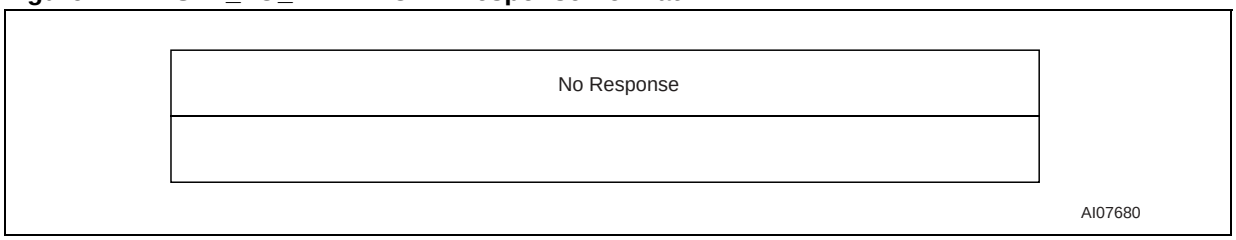
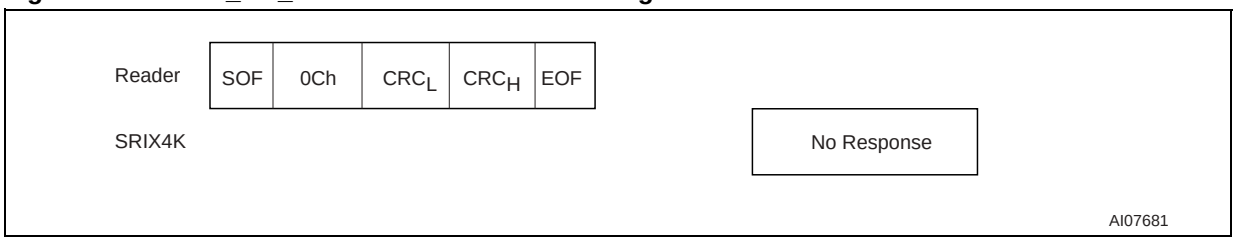


Figure 42. RESET_TO_INVENTORY Frame Exchange Between Reader and SRIX4K



READ_BLOCK(Addr) Command

Command Code = 08h

On receiving the READ_BLOCK command, the SRIX4K reads the desired block and returns the 4 data Bytes contained in the block. Data Bytes are transmitted with the Least Significant Byte first and each byte is transmitted with the least significant bit first.

The address byte gives access to the 128 blocks of the SRIX4K (addresses 0 to 127). READ_BLOCK commands issued with a block address above 127 will not be interpreted and the SRIX4K will not return any response, except for the System area located at address 255.

The SRIX4K must have received a SELECT() command and be switched to SELECTED state before any READ_BLOCK() command can be accepted. All READ_BLOCK() commands sent to the SRIX4K before a SELECT() command is issued are ignored.

Figure 43. READ_BLOCK Request Format

SOF	READ_BLOCK	ADDRESS	CRC_L	CRC_H	EOF
	08h	8 blts	8 bits	8 bits	

AI07684

Request parameter:

– ADDRESS: block addresses from 0 to 127, or 255

Figure 44. READ_BLOCK Response Format

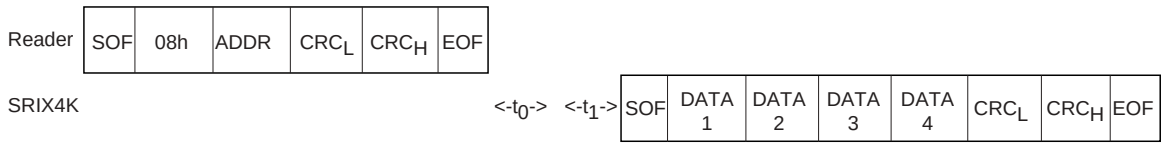
SOF	DATA 1	DATA 2	DATA 3	DATA 4	CRC_L	CRC_H	EOF
	8 blts	8 blts	8 blts	8 blts	8 bits	8 blts	

AI07685

Response parameters:

– DATA 1: Less significant data Byte
– DATA 2: Data Byte
– DATA 3: Data Byte
– DATA 4: Most significant data Byte

Figure 45. READ_BLOCK Frame Exchange Between Reader and SRIX4K



AI07686

WRITE_BLOCK (Addr, Data) Command

Command Code = 09h

On receiving the WRITE_BLOCK command, the SRIX4K writes the 4 bytes contained in the command to the addressed block, provided that the block is available and not Write-protected. Data Bytes are transmitted with the Least Significant Byte first, and each byte is transmitted with the least significant bit first.

The address Byte gives access to the 128 blocks of the SRIX4K (addresses 0 to 127). WRITE_BLOCK commands issued with a block address above 127 will not be interpreted and the SRIX4K will not return any response, except for the System area located at address 255.

The result of the WRITE_BLOCK command is submitted to the addressed block. See the following paragraphs for a complete description of the WRITE_BLOCK command:

- Resettable OTP Area (addresses 0 to 4).
- Binary Counter (addresses 5 to 6).
- EEPROM (Addresses 7 to 127).

The WRITE_BLOCK command does not give rise to a response from the SRIX4K. The reader must check after the programming time, t_W, that the data was correctly programmed. The SRIX4K must have received a SELECT() command and be switched to SELECTED state before any WRITE_BLOCK command can be accepted. All WRITE_BLOCK commands sent to the SRIX4K before a SELECT() command is issued, are ignored.

Figure 46. WRITE_BLOCK Request Format

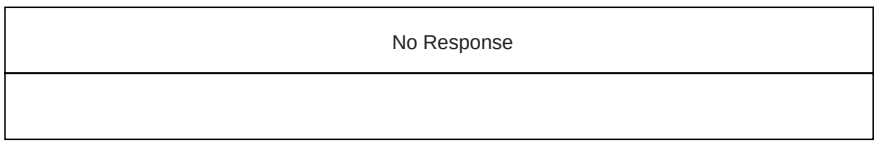
SOF	WRITE_BLOCK	ADDRESS	DATA 1	DATA 2	DATA 3	DATA 4	CRC_L	CRC_H	EOF
	09h	8 blts	8 blts	8 blts	8 blts	8 blts	8 bits	8 blts	

AI07687

Request parameters:

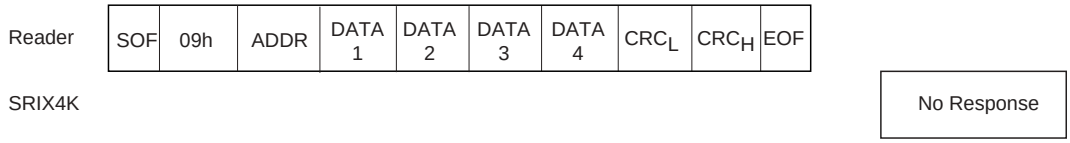
- ADDRESS: block addresses from 0 to 127, or 255
- DATA 1: Less significant data Byte
- DATA 2: Data Byte
- DATA 3: Data Byte
- DATA 4: Most significant data Byte.

Figure 47. WRITE_BLOCK Response Format



AI07680

Figure 48. WRITE_BLOCK Frame Exchange Between Reader and SRIX4K



AI07688

GET_UID() Command

Command Code = 0Bh

On receiving the GET_UID command, the SRIX4K returns its 8 UID Bytes. UID Bytes are transmitted with the Least Significant Byte first, and each byte is transmitted with the least significant bit first.

The SRIX4K must have received a SELECT() command and be switched to SELECTED state before any GET_UID() command can be accepted. All GET_UID() commands sent to the SRIX4K before a SELECT() command is issued, are ignored.

Figure 49. GET_UID Request Format

SOF	GET_UID	CRC_L	CRC_H	EOF
	0Bh	8 bits	8 bits	

AI07693

Request parameter:

– No parameter

Figure 50. GET_UID Response Format

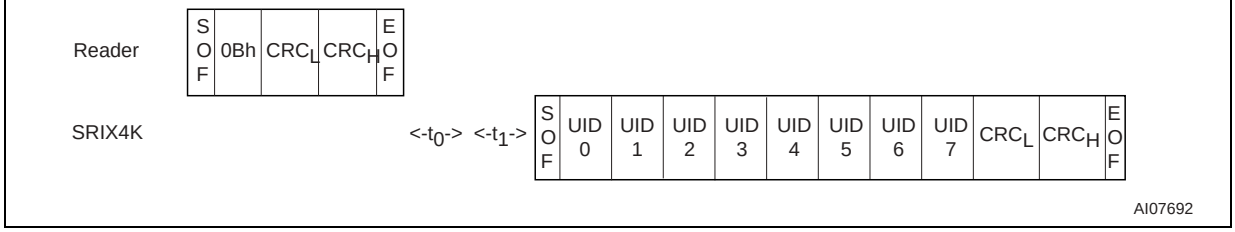
SOF	UID 0	UID 1	UID 2	UID 3	UID 4	UID 5	UID 6	UID 7	CRC_L	CRC_H	EOF
	8 bits	8 bits	8 bits	8 bits	8 bits	8 bits	8 bits	8 bits	8 bits	8 bits	

AI07694

Response parameters:

– UID 0: Less significant UID Byte
– UID 1 to UID 6: UID Bytes
– UID 7: Most significant UID Byte.

Figure 51. GET_UID Frame Exchange Between Reader and SRIX4K



AI07692

Power-On State

After Power-On, the SRIX4K is in the following state:

– It is in the low-power state.
– It is in READY state.
– It shows highest impedance with respect to the reader antenna field.
– It will not respond to any command except INITIATE().

MAXIMUM RATING

Stressing the device above the rating listed in the Absolute Maximum Ratings table may cause permanent damage to the device. These are stress ratings only and operation of the device at these or any other conditions above those indicated in the Operating sections of this specification is not implied. Exposure to Absolute Maximum Rating conditions for extended periods may affect device reliability. Refer also to the STMicroelectronics SURE Program and other relevant quality documents.

Table 5. Absolute Maximum Ratings

Symbol	Parameter		Min.	Max.	Unit
T_{STG}, h_{STG}, t_{STG}	Storage Conditions	Wafer	15	25	°C
				23	months
			kept in its antistatic bag		
		A3, A4, A5	15	25	°C
			40%	60%	RH
				2	years
I_{CC}	Supply Current on AC0 / AC1		–20	20	mA
V_{MAX}	Input Voltage on AC0 / AC1		–7	7	V
V_{ESD}	Electrostatic Discharge Voltage	Machine model [1]	–100	100	V
		Human Body model [1]	–1000	1000	V
		Human Body model [2]	–4000	4000	V

Note: 1. Mil. Std. 883 - Method 3015
2. ESD test: ISO10373-6 for proximity cards

DC AND AC PARAMETERS

Table 6. Operating Conditions

Symbol	Parameter	Min.	Max.	Unit
T_A	Ambient Operating Temperature	–20	85	°C

Table 7. DC Characteristics

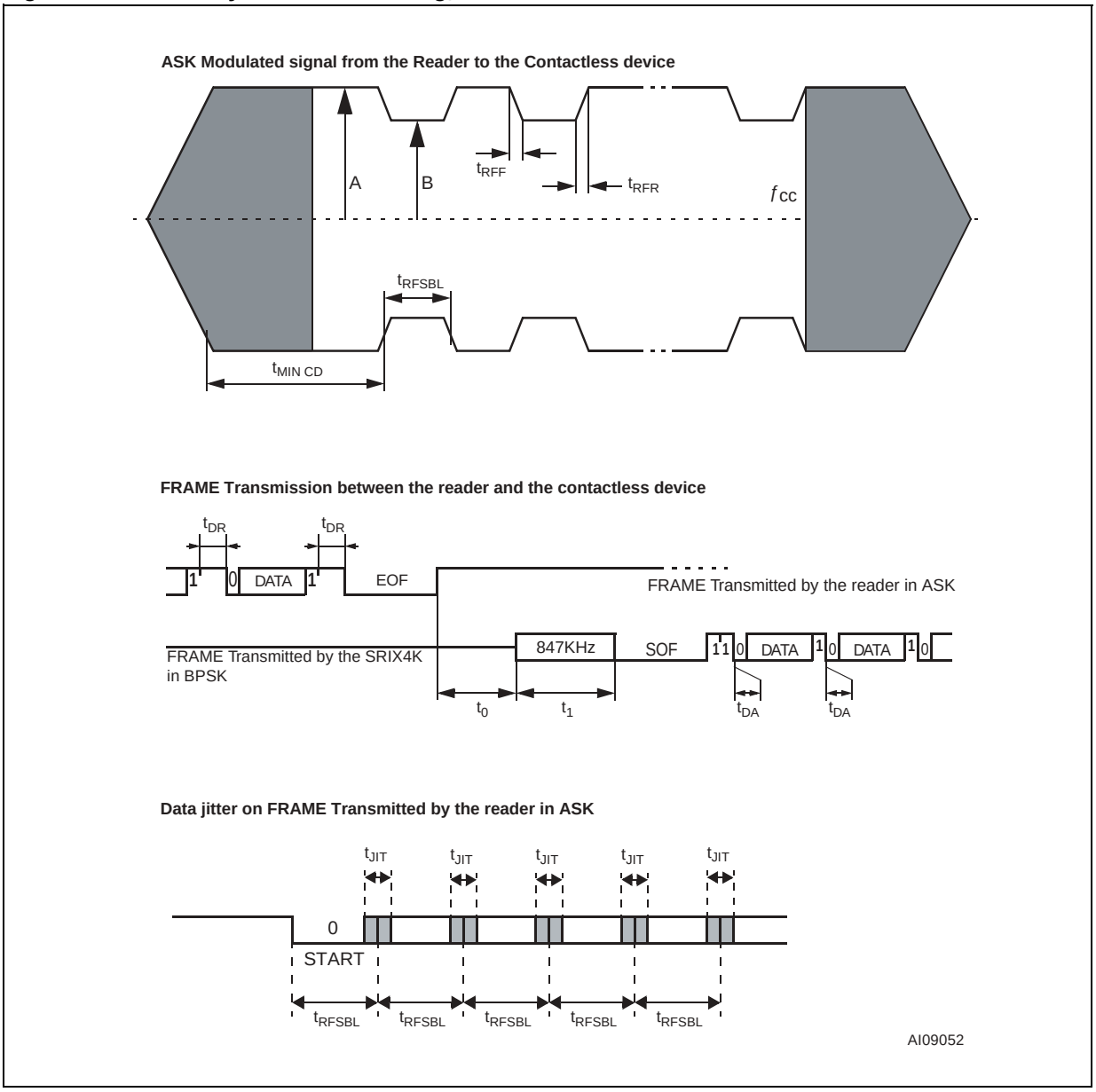
Symbol	Parameter	Condition	Min	Max	Unit
V_{CC}	Regulated Voltage		2.5	3.5	V
I_{CC}	Supply Current (Active in Read)	V_{CC} = 3.0V		100	µA
I_{CC}	Supply Current (Active in Write)	V_{CC} = 3.0V		250	µA

Table 8. AC Characteristics

Symbol	Parameter	Condition	Min	Max	Unit
f_{CC}	External RF Signal Frequency		13.553	13.567	MHz
$MI_{CARRIER}$	Carrier Modulation Index	MI=(A-B)/(A+B)	8	14	%
t_{RFR}, t_{RFF}	10% Rise and Fall times		0.8	2.5	µs
t_{RFSBL}	Minimum Pulse Width for Start bit	ETU = 128/f_{CC}	9.44		µs
t_{JIT}	ASK modulation Data Jitter	Coupler to SRIX4K	–2	+2	µs
$t_{MIN\ CD}$	Minimum Time from Carrier Generation to First Data		5		ms
f_S	Subcarrier Frequency	f_{CC}/16	847.5		kHz
t_0	Antenna Reversal Delay	128/f_S	151		µs
t_1	Synchronization Delay	128/f_S	151		µs
t_2	Answer to New Request Delay	14 ETU	132		µs
t_{DR}	Time Between Request Characters	Coupler to SRIX4K	0	57	µs
t_{DA}	Time Between Answer Characters	SRIX4K to Coupler	0		µs
t_W	Programming Time for WRITE	With no Auto-Erase Cycle (OTP)		3	ms
		With Auto-Erase Cycle (EEPROM)		5	ms
		Binary Counter Decrement		7	ms

Note: 1. All timing measurements were performed on a reference antenna with the following characteristics:
External size: 75mm x 48mm
Number of turns: 3
Width of conductor: 1mm
Space between 2 conductors: 0.4mm
Value of the coil: 1.4µH
Tuning Frequency: 14.4MHz.

Figure 52. SRIX4K Synchronous Timing, Transmit and Receive



PACKAGE MECHANICAL

Figure 53. A3 Antenna on Tape Specification

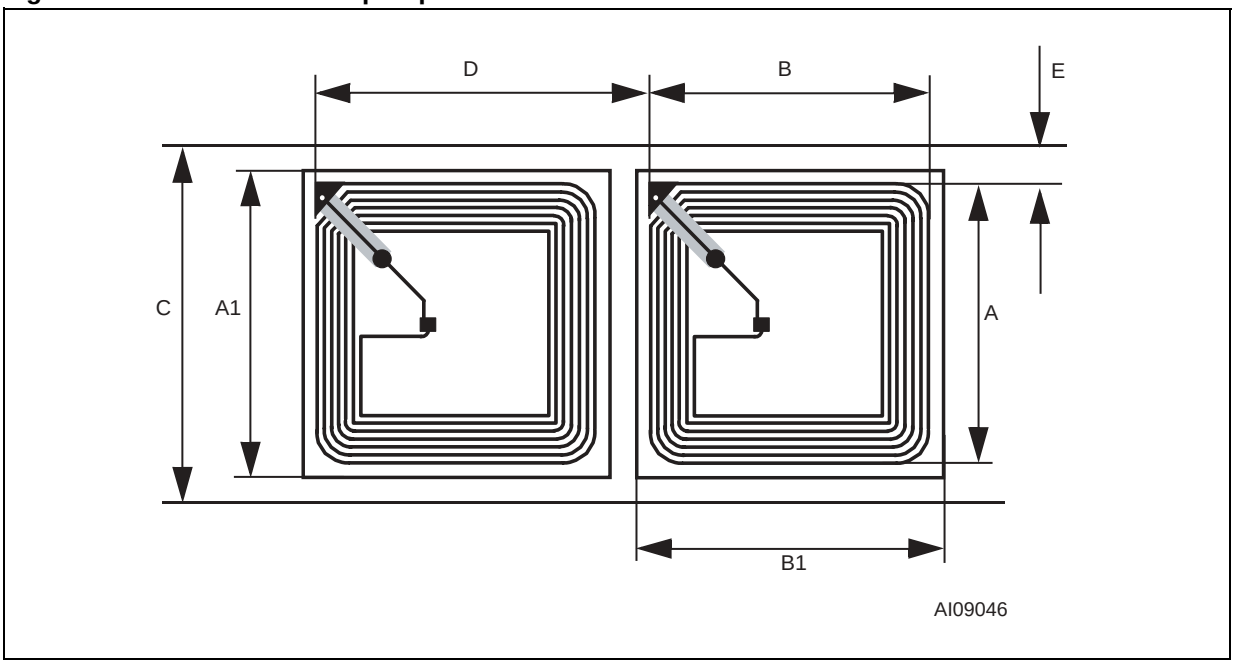


Table 9. A3 Antenna on Tape Specification

Symbol	Parameter	Type	Min	Max	Unit
A	Coil Width	38	37.5	38.5	mm
B	Coil Length	38	37.5	38.5	mm
A1	Inlay Width	43	42.5	43.5	mm
B1	Inlay Length	43	42.5	43.5	mm
C	Web Width	48	47.5	48.5	mm
D	Pitch	48	47.5	48.5	mm
E	Coil Distance from Web Edge	5	4.5	5.5	mm
	Overall Thickness of Copper Antenna Coil	110	90	130	µm
	Silicon Thickness	180	165	195	µm
Q	Unloaded Q Value	40			
F_{NOM}	Unloaded Free-air Resonance	15.1			MHz
P_A	H-field Energy for Device Operation		0.5 114		A/m dbµA/m

Figure 54. A4 Antenna on Tape Specification

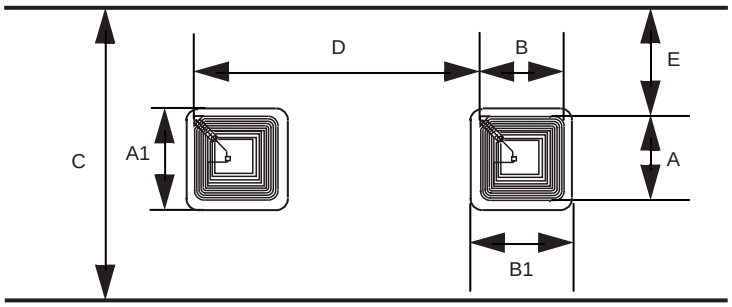


Table 10. A4 Antenna on Tape Specification

Symbol	Parameter	Type	Min	Max	Unit
A	Coil Width	15	14.5	15.5	mm
B	Coil Length	15	14.5	15.5	mm
A1	Inlay Width	19	18.5	19.5	mm
B1	Inlay Length	19	18.5	19.5	mm
C	Web Width	48	47.5	48.5	mm
D	Pitch	48	47.5	48.5	mm
E	Coil Distance from Web Edge	16.5	17	16	mm
	Overall Thickness of Copper Antenna Coil	110	90	130	µm
	Silicon Thickness	180	165	195	µm
Q	Unloaded Q Value	30			
F_{NOM}	Unloaded Free-air Resonance	14.5			MHz
P_A	H-field Energy for Device Operation		1.5 123.5		A/m dbµA/m

Figure 55. A5 Antenna on Tape Specification

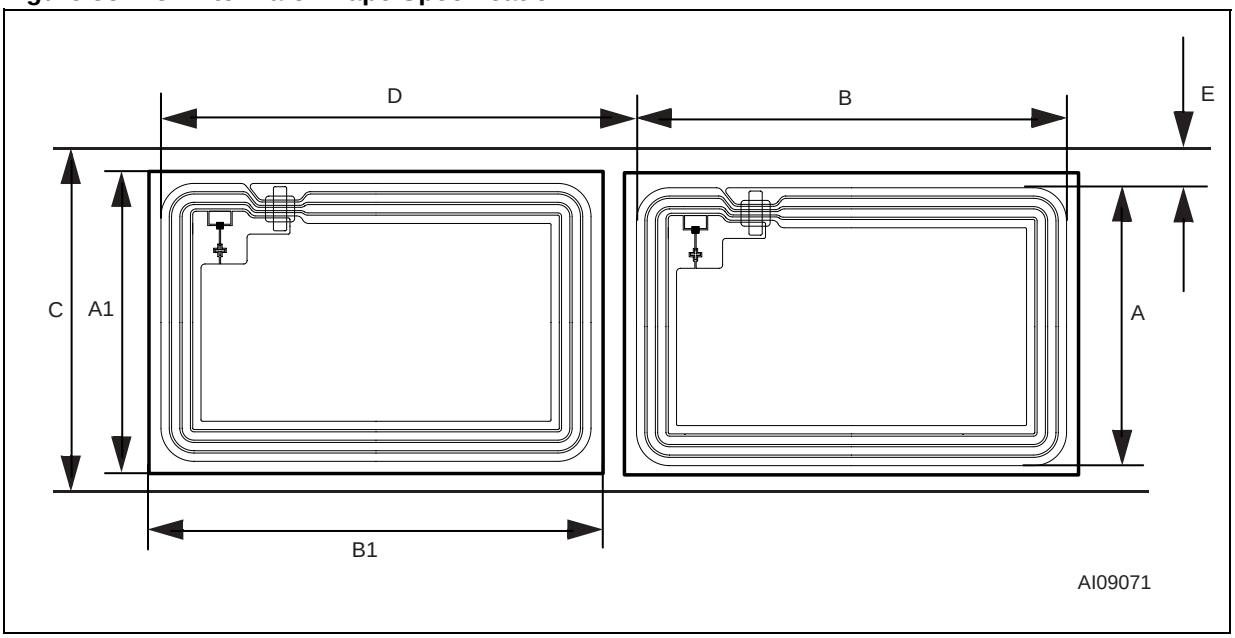


Table 11. A5 Antenna on Tape Specification

Symbol	Parameter	Type	Min	Max	Unit
A	Coil Width	42	41.5	42.5	mm
B	Coil Length	65	64.5	65.5	mm
A1	Inlay Width	46	45.5	46.5	mm
B1	Inlay Length	70	69.5	70.5	mm
C	Web Width	53	52.5	53.5	mm
D	Pitch	80	79	81	mm
E	Coil Distance from Web Edge	5.5	3.5	7.5	mm
	Overall Thickness of Copper Antenna Coil	140	130	150	µm
	Silicon Thickness	180	165	195	µm
Q	Unloaded Q Value	30			
F_{NOM}	Unloaded Free-air Resonance	14.8			MHz
P_A	H-field Energy for Device Operation		0.25 108		A/m dbµA/m

PART NUMBERING

Table 12. Ordering Information Scheme

Example: SRIX4K – W4 / XXX

Device Type

SRIX4K

Package

W4 =180 µm ± 15 µm Unsawn Wafer
SBN18= 180µm ± 15 µm Bumped and Sawn Wafer on 8-inch Frame
A3T= 38mm x 38mm Copper Antenna on Continuous Tape
A3S= 38mm x 38mm Copper Singulated Adhesive Antenna on Tape
A4T= 15mm x 15mm Copper Antenna on Continuous Tape
A4S= 15mm x 15mm Copper Singulated Adhesive Antenna on Tape
A5T= 42mm x 65mm Copper Antenna on Continuous Tape
A5S= 42mm x 65mm Copper Singulated Adhesive Antenna on Tape

Customer Code

XXX = Given by STMicroelectronics

Note: Devices are shipped from the factory with the memory content bits erased to 1.

For a list of available options (Speed, Package, etc.) or for further information on any aspect of this device, please contact your nearest ST Sales Office.

APPENDIX A. ISO14443 TYPE B CRC CALCULATION

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <ctype.h>
#define BYTE unsigned char
#define USHORT unsigned short

unsigned short UpdateCrc(BYTE ch, USHORT
*lpwCrc)
{
   ch = (ch^(BYTE)((*lpwCrc) & 0x00FF));
   ch = (ch^(ch<<4));
   *lpwCrc = (*lpwCrc >> 8)^((USHORT)ch <<
8)^((USHORT)ch<<3)^((USHORT)ch>>4);
   return(*lpwCrc);
}

void ComputeCrc(char *Data, int Length,
BYTE *TransmitFirst, BYTE *TransmitSecond)
{
BYTE chBlock; USHORTt wCrc;
   wCrc = 0xFFFF; // ISO 3309
   do
      {
         chBlock = *Data++;
         UpdateCrc(chBlock, &wCrc);
         } while (--Length);
   wCrc = ~wCrc; // ISO 3309
   *TransmitFirst = (BYTE) (wCrc & 0xFF);
   *TransmitSecond = (BYTE) ((wCrc >> 8) &
0xFF);
   return;
}

int main(void)
{
BYTE BuffCRC_B[10] = {0x0A, 0x12, 0x34,
0x56}, First, Second, i;
   printf("Crc-16 G(x) = x^16 + x^12 + x^5
+ 1”);
   printf("CRC_B of [ ");
   for(i=0; i<4; i++)
      printf("%02X ",BuffCRC_B[i]);
   ComputeCrc(BuffCRC_B, 4, &First,
&Second);
   printf("] Transmitted: %02X then
%02X.”, First, Second);
   return(0);
```

APPENDIX B. SRIX4K COMMAND SUMMARY

Figure 56. INITIATE Frame Exchange Between Reader and SRIX4K

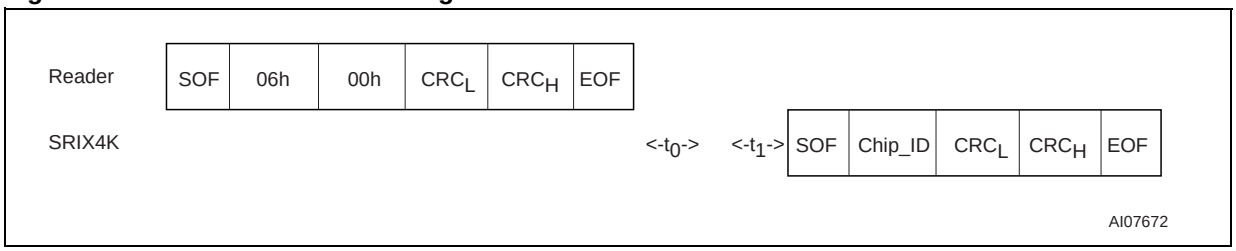


Figure 57. PCALL16 Frame Exchange Between Reader and SRIX4K

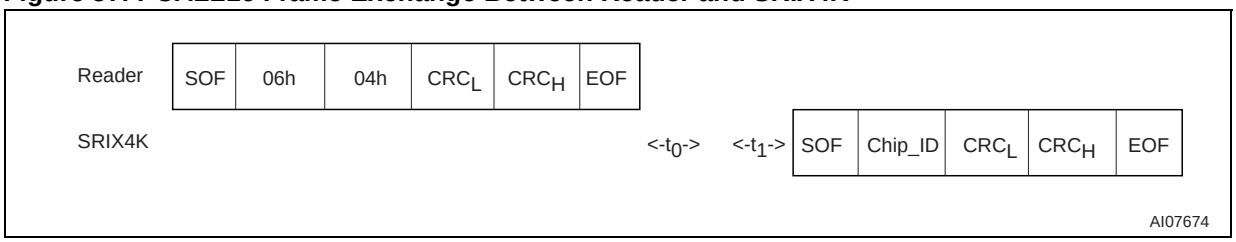


Figure 58. SLOT_MARKER Frame Exchange Between Reader and SRIX4K

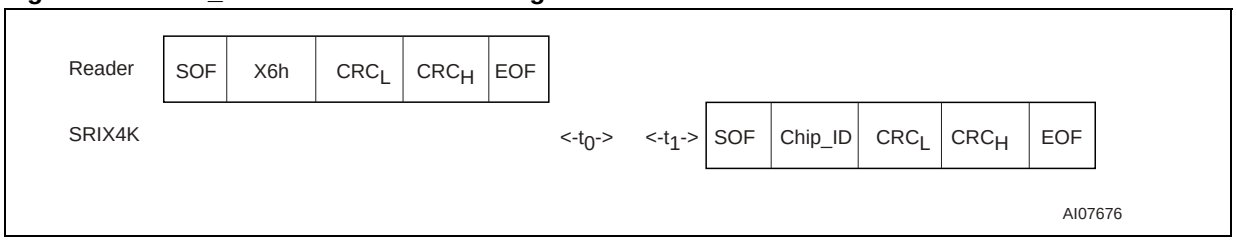


Figure 59. SELECT Frame Exchange Between Reader and SRIX4K

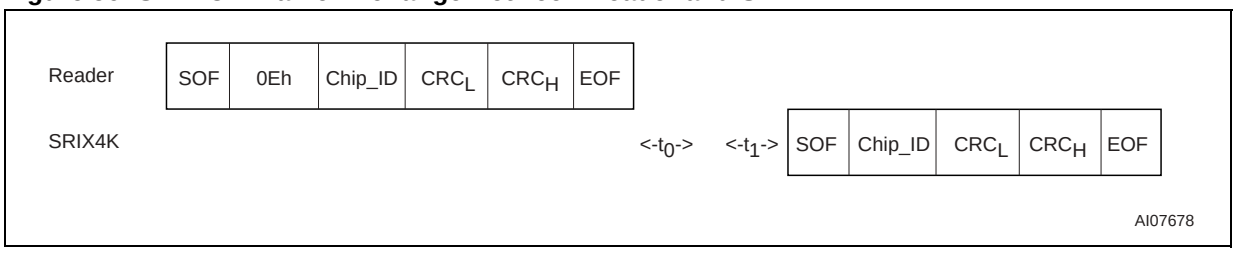


Figure 60. COMPLETION Frame Exchange Between Reader and SRIX4K

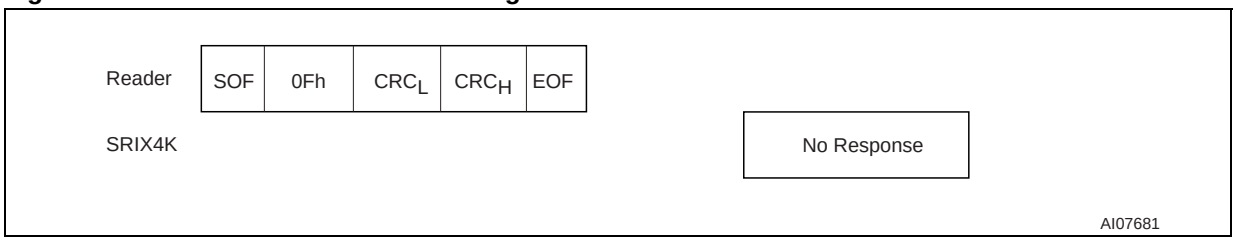


Figure 61. RESET_TO_INVENTORY Frame Exchange Between Reader and SRIX4K

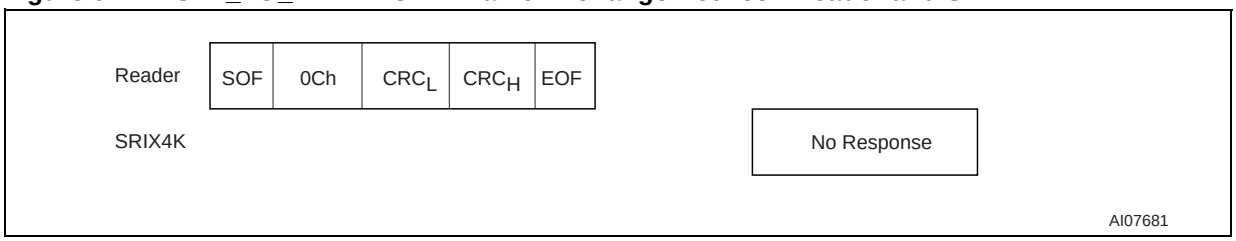


Figure 62. READ_BLOCK Frame Exchange Between Reader and SRIX4K

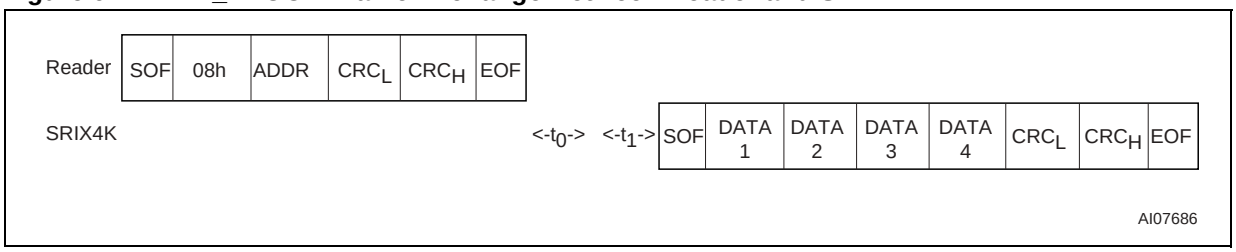


Figure 63. WRITE_BLOCK Frame Exchange Between Reader and SRIX4K

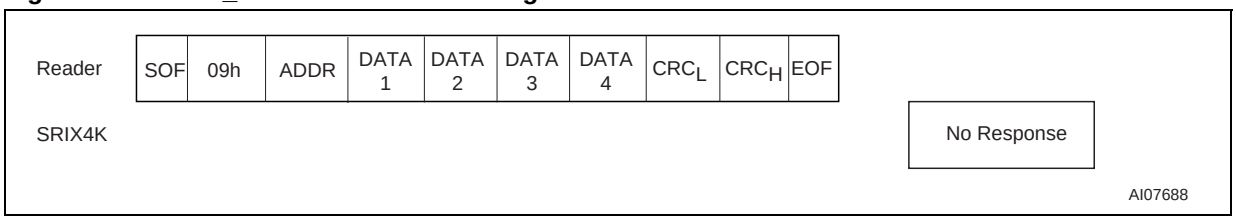
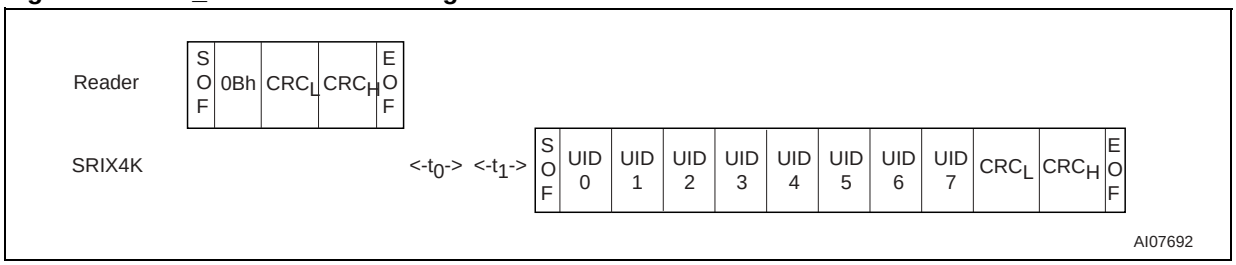


Figure 64. GET_UID Frame Exchange Between Reader and SRIX4K



REVISION HISTORY

Table 13. Document Revision History

Date	Version	Description of Revision
28-Nov-2002	1.0	Document written
17-Jul-2003	1.1	Data briefing extracted
12-Mar-2004	2.0	First public release of full datasheet
26-Apr-2004	3.0	Correction to memory map