

**ISO/IEC JTC 1/SC 17/WG8 N 1295R1**

Date: 2007-01-31

**ISO/IEC CD 14443-3**

ISO/IEC JTC 1/SC 17/WG 8

Secretariat: DIN

## **Identification cards — Contactless integrated circuit(s) cards - Proximity cards — Part 3: Initialization and anticollision**

*Cartes d'identification — Cartes à circuit(s) intégré(s) sans contact - Cartes de proximité — Partie 3: Initialisation et anticollision*

### **Warning**

This document is not an ISO International Standard. It is distributed for review and comment. It is subject to change without notice and may not be referred to as an International Standard.

Recipients of this draft are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

### Copyright notice

This ISO document is a working draft or committee draft and is copyright-protected by ISO. While the reproduction of working drafts or committee drafts in any form for use by participants in the ISO standards development process is permitted without prior permission from ISO, neither this document nor any extract from it may be reproduced, stored or transmitted in any form for any other purpose without prior written permission from ISO.

Requests for permission to reproduce this document for the purpose of selling it should be addressed as shown below or to ISO's member body in the country of the requester:

[Indicate the full address, telephone number, fax number, telex number, and electronic mail address, as appropriate, of the Copyright Manger of the ISO member body responsible for the secretariat of the TC or SC within the framework of which the working document has been prepared.]

Reproduction for sales purposes may be subject to royalty payments or a licensing agreement.

Violators may be prosecuted.

# Contents

Page

|       |  |    |
|-------|--|----|
| 1     | Scope .....  | 1  |
| 2     | Normative references .....                                   | 1  |
| 3     | Terms and definitions.....                                   | 2  |
| 4     | Symbols and abbreviated terms .....                          | 3  |
| 5     | Alternating between Type A and Type B commands.....          | 5  |
| 5.1   | Polling .....  | 5  |
| 5.2   | Influence of Type A commands on PICC Type B operation.....   | 5  |
| 5.3   | Influence of Type B commands on PICC Type A operation.....   | 5  |
| 5.4   | Transition to Power OFF state.....                           | 6  |
| 6     | Type A – Initialization and anticollision.....               | 6  |
| 6.1   | Bit rates.....   | 6  |
| 6.2   | Frame format and timing.....                                 | 6  |
| 6.2.1 | Frame delay time.....  | 7  |
| 6.2.2 | Frame delay time PCD to PICC .....                           | 7  |
| 6.2.3 | Frame delay time PICC to PCD .....                           | 8  |
| 6.2.4 | Request Guard Time.....                                      | 8  |
| 6.2.5 | Frame formats .....  | 8  |
| 6.2.6 | CRC_A.....   | 11 |
| 6.3   | PICC states .....  | 11 |
| 6.3.1 | POWER-OFF State .....  | 13 |
| 6.3.2 | IDLE State .....   | 14 |
| 6.3.3 | READY State.....   | 14 |
| 6.3.4 | ACTIVE State .....   | 14 |
| 6.3.5 | HALT State.....  | 14 |
| 6.3.6 | READY* State .....   | 14 |
| 6.3.7 | ACTIVE* State.....   | 15 |
| 6.4   | Command set .....  | 15 |
| 6.4.1 | REQA and WUPA Commands .....                                 | 15 |
| 6.4.2 | ANTICOLLISION and SELECT Commands .....                      | 17 |
| 6.4.3 | HLTA Command.....  | 17 |
| 6.5   | Select sequence.....   | 17 |
| 6.5.1 | Select sequence flowchart.....                               | 18 |
| 6.5.2 | ATQA - Answer To Request.....                                | 18 |
| 6.5.3 | Anticollision and Select .....                               | 20 |
| 6.5.4 | UID contents and cascade levels .....                        | 23 |
| 7     | Type B – Initialization and anticollision.....               | 26 |
| 7.1   | Character, frame format and timing.....                      | 26 |
| 7.1.1 | Character transmission format .....                          | 26 |
| 7.1.2 | Character separation .....                                   | 26 |
| 7.1.3 | Frame format .....   | 27 |
| 7.1.4 | SOF.....   | 27 |
| 7.1.5 | EOF.....   | 28 |
| 7.1.6 | Timing before the PICC SOF.....                              | 29 |
| 7.1.7 | Timing before the PCD SOF.....                               | 30 |
| 7.2   | CRC_B.....   | 31 |
| 7.3   | Anticollision sequence.....                                  | 31 |
| 7.4   | PICC states description.....                                 | 31 |
| 7.4.1 | Initialization and anticollision flowchart.....              | 33 |
| 7.4.2 | General statement for state description and transitions..... | 33 |

|  |   |           |
|--|---|-----------|
| 7.4.3  | POWER-OFF State .....                         | 34        |
| 7.4.4  | IDLE State .....                              | 34        |
| 7.4.5  | READY-REQUESTED sub-state .....               | 34        |
| 7.4.6  | READY-DECLARED sub-state .....                | 35        |
| 7.4.7  | ACTIVE State .....                            | 35        |
| 7.4.8  | HALT State .....                              | 35        |
| 7.5  | Command set.....                              | 36        |
| 7.6  | Anticollision response rules .....            | 36        |
| 7.7  | REQB/WUPB Command .....                       | 36        |
| 7.7.1  | REQB/WUPB Command format .....                | 37        |
| 7.7.2  | Coding of Anticollision Prefix byte APf ..... | 37        |
| 7.7.3  | Coding of AFI.....                            | 37        |
| 7.7.4  | Coding of PARAM .....                         | 38        |
| 7.8  | Slot-MARKER Command.....                      | 38        |
| 7.8.1  | Slot-MARKER Command format.....               | 39        |
| 7.8.2  | Coding of Anticollision Prefix byte APn ..... | 39        |
| 7.9  | ATQB Response.....                            | 39        |
| 7.9.1  | ATQB Response format.....                     | 39        |
| 7.9.2  | PUPI (Pseudo-Unique PICC Identifier) .....    | 40        |
| 7.9.3  | Application Data.....                         | 40        |
| 7.9.4  | Protocol Info .....                           | 40        |
| 7.10   | ATTRIB Command .....                          | 44        |
| 7.10.1   | ATTRIB Command format .....                   | 44        |
| 7.10.2   | Identifier .....                              | 44        |
| 7.10.3   | Coding of Param 1 .....                       | 44        |
| 7.10.4   | Coding of Param 2 .....                       | 46        |
| 7.10.5   | Coding of Param 3 .....                       | 46        |
| 7.10.6   | Coding of Param 4 .....                       | 46        |
| 7.10.7   | Higher layer INF.....                         | 47        |
| 7.11   | Answer to ATTRIB Command.....                 | 47        |
| 7.12   | HLTB Command and Answer.....                  | 48        |
| <b>Annex A (informative) Communication example Type A.....</b> |   | <b>49</b> |
| <b>Annex B (informative) CRC_A and CRC_B encoding .....</b>    |   | <b>51</b> |

## Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of the joint technical committee is to prepare International Standards. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

ISO/IEC 14443-3 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information Technology*, Subcommittee SC 17, *Identification cards and related devices*.

This second/third/... edition cancels and replaces the first/second/... edition (), [clause(s) / subclause(s) / table(s) / figure(s) / annex(es)] of which [has / have] been technically revised.

ISO/IEC 14443 consists of the following parts, under the general title *Identification cards — Contactless integrated circuit(s) cards - Proximity cards*:

- *Part 3: Initialization and anticollision*
- *Part [n]:*
- *Part [n+1]:*

## Introduction

ISO/IEC 14443 is one of a series of International Standards describing the parameters for identification cards as defined in ISO/IEC 7810, and the use of such cards for international interchange.

This part of ISO/IEC 14443 describes polling for proximity cards entering the field of a proximity coupling device, the byte format and framing, the initial Request and Answer to Request command content, methods to detect and communicate with one proximity card among several proximity cards (anticollision) and other parameters required to initialize communications between a proximity card and a proximity coupling device. Protocols and commands used by higher layers and by applications and which are used after the initial phase are described in ISO/IEC 14443-4.

ISO/IEC 14443 is intended to allow operation of proximity cards in the presence of other contactless cards conforming to ISO/IEC 10536 and ISO/IEC 15693.

# Identification cards — Contactless integrated circuit(s) cards - Proximity cards — Part 3: Initialization and anticollision

## 1 Scope

This part of ISO/IEC 14443 describes:

- polling for proximity cards (PICCs) entering the field of a proximity coupling device (PCD);
- the byte format, the frames and timing used during the initial phase of communication between PCDs and PICCs;
- the initial Request and Answer to Request command content;
- methods to detect and communicate with one PICC among several PICCs (anticollision);
- other parameters required to initialize communications between a PICC and PCD;
- optional means to ease and speed up the selection of one PICC among several PICCs based on application criteria.

Protocol and commands used by higher layers and by applications and which are used after the initial phase are described in ISO/IEC 14443-4.

This part of ISO/IEC 14443 is applicable to PICCs of Type A and of Type B (as described in ISO/IEC 14443-2).

NOTE Part of the timing of data communication is defined in ISO/IEC 14443-2.

## 2 Normative references

The following normative documents contain provisions which, through reference in this text, constitute provisions of this part of ISO/IEC 14443. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. However, parties to agreements based on this part of ISO/IEC 14443 are encouraged to investigate the possibility of applying the most recent editions of the normative documents indicated below. For undated references, the latest edition of the normative document referred to applies. Members of ISO and IEC maintain registers of currently valid International Standards.

ISO/IEC 13239, *Information technology - Telecommunications and information exchange between systems - High-level data link control (HDLC) procedures*.

ISO/IEC 7816-3:1997, *Identification cards - Integrated circuit(s) cards with contacts - Part 3: Electronic signals and transmission protocols*.

ISO/IEC 7816-5, *Identification cards - Integrated circuit(s) cards with contacts - Part 5: Numbering system and registration procedure for application identifiers*.

ISO/IEC 7816-6:1996/AM1, *Identification cards - Integrated circuit(s) cards with contacts - Part 6: Interindustry data elements - Amendment 1: IC manufacturer's register*.

ISO/IEC 14443-2, *Identification cards – Contactless integrated circuit(s) cards – Proximity cards – Part 2: Radio frequency power and signal interface.*

ISO/IEC 14443-4, *Identification cards – Contactless integrated circuit(s) cards – Proximity cards – Part 4: Transmission protocol.*

ITU-T X.25, *Interface between Data Terminal Equipment (DTE) and Data Circuit-terminating Equipment (DCE) for terminals operating in the packet mode and connected to public data network by dedicated circuit.*

ITU-T V.41, *Code-independent error-control system.*

ITU-T V.42, *Error-correcting procedures for DLEs using asynchronous – to – synchronous conversion.*

### 3 Terms and definitions

For the purposes of this part of ISO/IEC 14443, the terms and definitions given in ISO/IEC 14443-2, ISO/IEC 7816-3 and the following apply.

#### 3.1 anticollision loop

algorithm used to prepare for dialogue between PCD and one or more PICCs out of the total number of PICCs responding to a request command.

#### 3.2 bit collision detection protocol

anticollision method for PICCs of Type A, employing collision detection at bit level within a frame.

#### 3.3 byte

a byte consists of 8 bits of data designated b8 to b1, from the most significant bit (MSB, b8) to the least significant bit (LSB, b1)

#### 3.4 collision

transmission by two PICCs in the same PCD energizing field and during the same time period, such that the PCD is unable to distinguish from which PICC the data originated

#### 3.5 elementary time unit (etu)

calculated by the following formula:

$$1 \text{ etu} = 128 / (D \times fc) \text{ where } D \in \{1,2,4,8\}$$

The initial value of the divisor D is 1, giving the initial etu as follows:

$$1 \text{ etu} = 128 / fc$$

where *fc* is the carrier frequency as defined in ISO/IEC 14443-2.3.6

#### 3.6 frame

a frame is a sequence of data bits and optional error detection bits, with frame delimiters at start and end

**3.7****higher layer protocol**

a protocol layer (not described in this part of ISO/IEC 14443) that makes use of the protocol layer defined in this part of ISO/IEC 14443 to transfer information belonging to the application or higher layers of protocol that is not described in this part of ISO/IEC 14443

**3.8****timeslot protocol**

method whereby a PCD establishes logical channels with one or more PICCs of Type B, which makes use of timeslot allocation for PICC response

**3.9****request command**

command requesting PICC of the appropriate type to respond if they are available for initialization

**4 Symbols and abbreviated terms**

For the purposes of this part of ISO/IEC 14443, the following abbreviations are used:

|        |  |
|--------|--|
| ADC    | Application Data Coding, Type B  |
| AFI    | Application Family Identifier. Card preselection criteria by application, Type B |
| APf    | Anticollision Prefix f, used in REQb/WUPb, Type B                                |
| APn    | Anticollision Prefix n, used in Slot-MARKER Command, Type B                      |
| ATQA   | Answer To Request, Type A  |
| ATQB   | Answer To Request, Type B  |
| ATTRIB | PICC selection command, Type B   |
| BCC    | UID CLn check byte, calculated as exclusive-or over the 4 previous bytes, Type A |
| CID    | Card IDentifier  |
| CLn    | Cascade Level n, Type A  |
| CT     | Cascade Tag, Type A  |
| CRC_A  | Cyclic Redundancy Check error detection code A                                   |
| CRC_B  | Cyclic Redundancy Check error detection code B                                   |
| D      | Divisor  |
| E      | End of communication, Type A   |
| EGT    | Extra Guard Time, Type B   |
| EOF    | End Of Frame, Type B   |
| etu    | Elementary time unit.  |
| FDT    | Frame Delay Time, Type A   |
| fc     | Carrier frequency  |

|        |   |
|--------|---|
| FO     | Frame Option  |
| fs     | Subcarrier frequency  |
| FWI    | Frame Waiting time Integer  |
| FWT    | Frame Waiting Time  |
| HLTA   | Halt Command, Type A  |
| HLTB   | Halt Command, Type B  |
| ID     | IDentification number, Type A   |
| INF    | INFormation field belonging to higher layer, Type B                             |
| LSB    | Least Significant Bit   |
| MBL    | Maximum Buffer Length, Type B   |
| MBLI   | Maximum Buffer Length Index, Type B   |
| MSB    | Most Significant Bit  |
| N      | Number of anticollision slots or PICC response probability in each slot, Type B |
| n      | Variable integer value as defined in the specific clause                        |
| NAD    | Node ADDRESS  |
| NVB    | Number of Valid Bits, Type A  |
| P      | Odd Parity bit, Type A  |
| PCD    | Proximity Coupling Device   |
| PICC   | Proximity Card or Object  |
| PUPI   | Pseudo-Unique PICC Identifier, Type B   |
| R      | Slot number chosen by the PICC during the anticollision sequence, Type B        |
| REQA   | Request Command, Type A   |
| REQB   | Request Command, Type B   |
| RFU    | Reserved for Future Use by ISO/IEC  |
| S      | Start of communication, Type A  |
| SAK    | Select Acknowledge, Type A  |
| SEL    | SElect code, Type A   |
| SELECT | Select Command, Type A  |
| SFGI   | Start-up Frame Guard time Integer   |
| SFGT   | Start-up Frame Guard Time   |

|         |  |
|---------|--|
| SOF     | Start Of Frame, Type B                                     |
| TR0     | Guard Time as defined in ISO/IEC 14443-2, Type B           |
| TR1     | Synchronization Time as defined in ISO/IEC 14443-2, Type B |
| TR2     | Frame delay Time PICC to PCD, Type B                       |
| UID     | Unique Identifier, Type A                                  |
| uid $n$ | Byte number $n$ of Unique Identifier, $n \geq 0$           |
| WUPA    | Wake-UP Command, Type A                                    |
| WUPB    | Wake-UP Command, Type B                                    |

For the purposes of this part of ISO/IEC 14443, the following notation applies:

- (xxxxx)b Data bit representation;
- 'XY' Hexadecimal notation, equal to XY to the base 16.

## 5 Alternating between Type A and Type B commands

### 5.1 Polling

In order to detect PICCs which are in the operating field, a PCD shall send repeated Request commands. The PCD shall send REQA and REQB described herein in any sequence and in addition may send other commands as described in Annex C.

When a PICC is exposed to an unmodulated operating field (see ISO/IEC 14443-2) it shall be able to accept a request within 5 ms.

EXAMPLE 1 When a PICC Type A receives any Type B command it shall be able to accept a REQA within 5 ms of unmodulated operating field.

EXAMPLE 2 When a PICC Type B receives any Type A command it shall be able to accept a REQB within 5 ms of unmodulated operating field.

EXAMPLE 3 When a PICC Type A is exposed to field activation it shall be able to accept a REQA within 5 ms of unmodulated operating field.

EXAMPLE 4 When a PICC Type B is exposed to field activation it shall be able to accept a REQB within 5 ms of unmodulated operating field.

NOTE The PCD should apply a additional marginal waiting time of 0.1 ms.

### 5.2 Influence of Type A commands on PICC Type B operation

A PICC Type B should either go to IDLE state (be able to accept a REQB) or be able to continue a transaction in progress after receiving any Type A command.

### 5.3 Influence of Type B commands on PICC Type A operation

A PICC Type A should either go to IDLE state (be able to accept a REQA) or be able to continue a transaction in progress after receiving any Type B command

**5.4 Transition to Power OFF state**

The PICC shall be in the Power OFF state no later than 5 ms after the operating field is switched off.

NOTE The envelope of the PCD field shall remain less than 5 % of  $H_{INITIAL}$  for 5,1 ms for PICC transition to power off.

**6 Type A – Initialization and anticollision**

This section describes the initialization and collision detection protocol applicable for PICCs of Type A.

The PCD shall be designed to detect a collision that occurs when at least two PICCs simultaneously transmit bit patterns with one or more bit positions in which at least two PICCs transmit complementary values. In this case the bit patterns merge and the carrier is modulated with the subcarrier for the whole (100%) bit duration (see ISO/IEC 14443-2).

**6.1 Bit rates**

Communication between PCD and PICC can be achieved with four different bit rates.

Bit rates of  $fc/64$ ,  $fc/32$  and  $fc/16$  are optional and may be independently supported by PCD and PICC.

**Table 1 — Bit rates**

| Divisor D    | etu                        | Bit rate               |
|--------------|----------------------------|------------------------|
| 1            | $128/fc$ (~9,4 $\mu$ s)    | $fc/128$ (~106 kbit/s) |
| 2 (optional) | $128/(2fc)$ (~4,7 $\mu$ s) | $fc/64$ (~212 kbit/s)  |
| 4 (optional) | $128/(4fc)$ (~2,4 $\mu$ s) | $fc/32$ (~424 kbit/s)  |
| 8 (optional) | $128/(8fc)$ (~1,2 $\mu$ s) | $fc/16$ (~847 kbit/s)  |

NOTE The initial bit rate is  $fc/128$ . This applies for the whole initialization and anticollision sequence.

**6.2 Frame format and timing**

This section defines the frame format and timing used during communication initialization and anticollision. For bit representation and coding refer to ISO/IEC 14443-2.

Frames shall be transferred in pairs, PCD to PICC followed by PICC to PCD, using the sequence:

- PCD frame:
  - PCD start of communication
  - information and, where required, error detection bits sent by the PCD
  - PCD end of communication
- Frame delay time PCD to PICC
- PICC frame:
  - PICC start of communication

- information and, where required, error detection bits sent by the PICC
- PICC end of communication
- Frame delay time PICC to PCD

The frame delay time (FDT) from PCD to PICC overlaps the PCD end of communication.

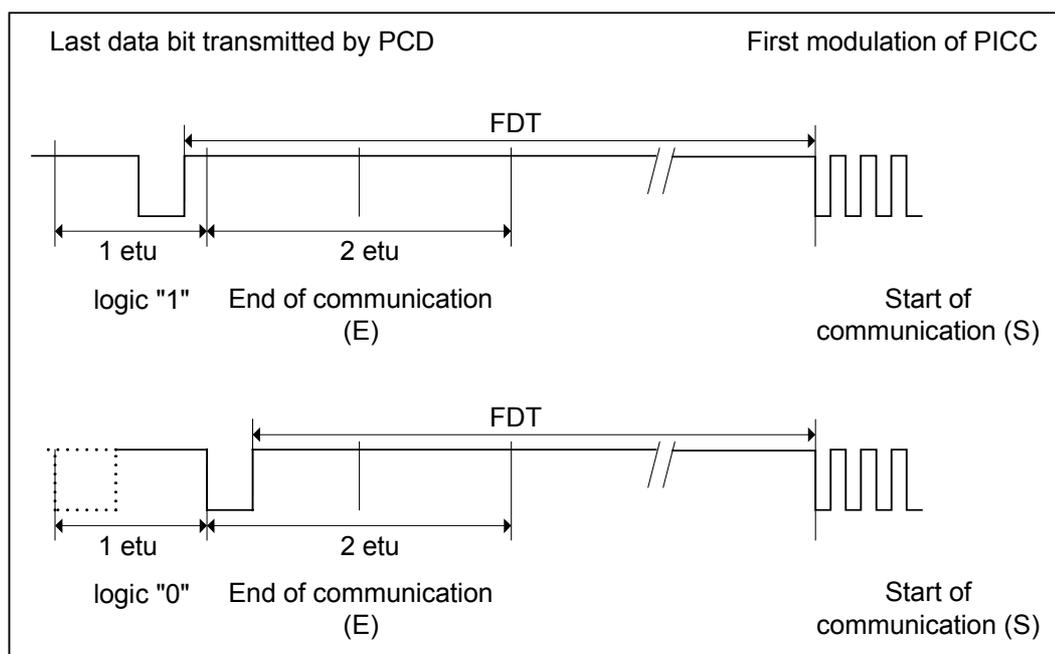
**6.2.1 Frame delay time**

The frame delay time FDT is defined as the time between two frames transmitted in opposite directions.

**6.2.2 Frame delay time PCD to PICC**

This is the time between the end of the last pause transmitted by the PCD and the first modulation edge within the start bit transmitted by the PICC and shall respect the timing defined in Figure 1, where *n* is an integer value.

Table 1 defines values for *n* and FDT depending on the command type and the logic state of the last transmitted data bit in this command.



**Figure 1 — Frame delay time PCD to PICC**

**Table 2 — Frame delay time PCD to PICC**

| Command type  |             | n (integer value) | FDT  |  |
|---|-------------|-------------------|--|--|
|   |             |                   | last bit = (1)b                                  | last bit = (0)b                                  |
| REQA Command<br>WUPA Command<br>ANTICOLLISION Command<br>SELECT Command |             | 9                 | $(n \cdot 128 + 84) / f_c$<br>[ = 1236 / $f_c$ ] | $(n \cdot 128 + 20) / f_c$<br>[ = 1172 / $f_c$ ] |
| <b>All other commands at bit rates</b>                                  |             |                   |  |  |
| PCD to PICC   | PICC to PCD |                   |  |  |

|  |  |                |                  |                  |
|--|--|----------------|------------------|------------------|
| <i>fc/128</i>  | <i>fc/128</i>                                | $\geq 9$       | $(n*128+84)/fc$  | $(n*128+20)/fc$  |
| <i>fc/64</i>   |  | $\geq 8$       | $(n*128+148)/fc$ | $(n*128+116)/fc$ |
| <i>fc/32</i>   |  | $\geq 8$       | $(n*128+116)/fc$ | $(n*128+100)/fc$ |
| <i>fc/16</i>   |  | $\geq 8$       | $(n*128+100)/fc$ | $(n*128+92)/fc$  |
| <i>fc/128</i> or <i>fc/64</i> or <i>fc/32</i> or <i>fc/16</i>  | <i>fc/64</i> or <i>fc/32</i> or <i>fc/16</i> | Not applicable | $\geq 1116/fc$   | $\geq 1116/fc$   |
| All PICCs in the field shall respond in a synchronous way to the commands: REQA, WUPA, ANTICOLLISION and SELECT. This is needed for anticollision. |  |                |                  |                  |

The FDT measurement starts at the beginning of the rising edge as specified in ISO/IEC 14443-2 and illustrated with small circles in figure 3 for *fc/128* and figure 6 for other bit rates.

The FDT tolerance is in the range of -0 to +0,4  $\mu$ s.

NOTE: The PCD should accept a response with a FDT tolerance of -1/*fc* to 0,4  $\mu$ s +1/*fc*.

### 6.2.3 Frame delay time PICC to PCD

This is the time between the last modulation transmitted by the PICC and the first pause transmitted by the PCD and shall be at least 1172/*fc*.

NOTE The PCD should apply an additional marginal waiting time of 10/*fc*.

### 6.2.4 Request Guard Time

The Request Guard Time is defined as the minimum time between the start bits of two consecutive REQA or WUPA commands. It has the value 7000/*fc*.

NOTE The PCD should apply an additional marginal waiting time of 10/*fc*.

### 6.2.5 Frame formats

The following frame types are defined:

- short frames for commands defined in Table 2;
- standard frames for regular commands;
- bit oriented anticollision frame for anticollision command.

#### 6.2.5.1 Short frame

A short frame is used to initiate communication and consists of, in the following order:

- start of communication;
- 7 data bits transmitted LSB first (for coding see Table 2);
- end of communication.

No parity bit is added.

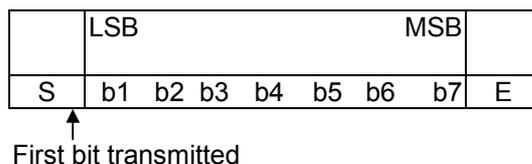


Figure 2 — Short frame

6.2.5.2 Standard frame

Standard frames are used for data exchange and consist of:

- start of communication;
- $n * (8 \text{ data bits} + \text{odd parity bit})$ , with  $n \geq 1$ . The LSB of each byte is transmitted first. Each byte is followed by an odd parity bit. The parity bit P is set such that the number of 1s is odd in (b1 to b8, P);
- end of communication.

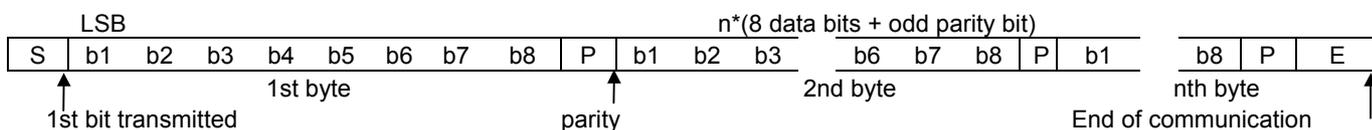


Figure 3 — Standard Frame

As an exception the last parity bit of a PICC standard frame shall be inverted if this frame is transmitted with bit rate of  $fc/64$ ,  $fc/32$  or  $fc/16$ .

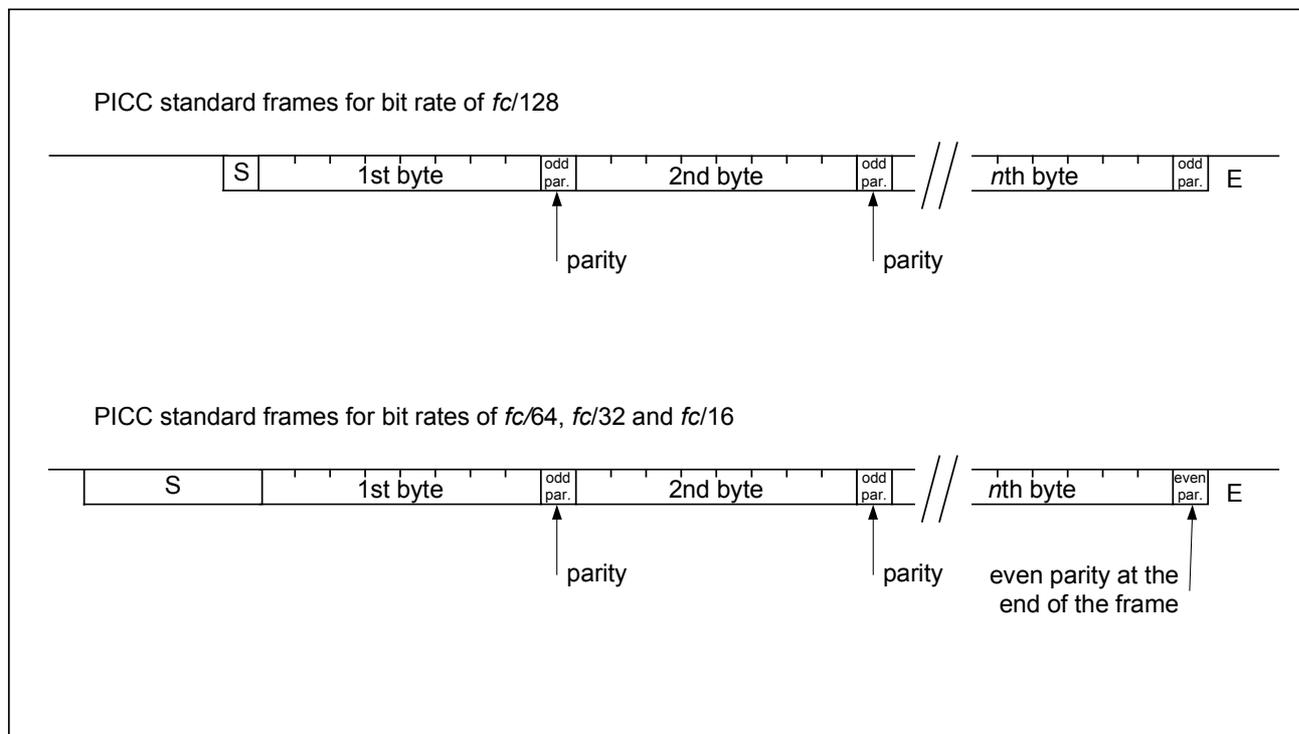


Figure 4 — ICC standard frames

6.2.5.3 Bit oriented anticollision frame

A collision is detected when at least two PICCs transmit different bit patterns to the PCD. In this case the carrier is modulated with the subcarrier for the whole bit duration for at least one bit.

Bit oriented anticollision frames shall only be used during bit frame anticollision loops and are standard frames with a length of 7 bytes, split into two parts:

- part 1 for transmission from PCD to PICC;
- part 2 for transmission from PICC to PCD.

For the length of part 1 and part 2, the following rules shall apply:

- rule 1: The sum of data bits shall be 56;
- rule 2: The minimum length of part 1 shall be 16 data bits;
- rule 3: The maximum length of part 1 shall be 48 data bits.

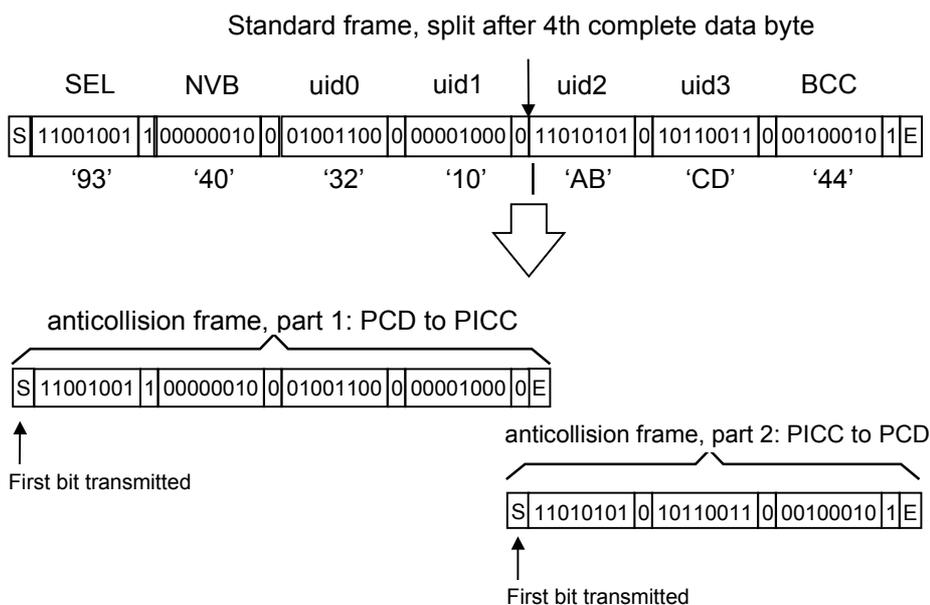
Consequently, the minimum length of part 2 shall be 8 data bit and the maximum length shall be 40 data bits.

Since the split can occur at any bit position within a byte, two cases are defined:

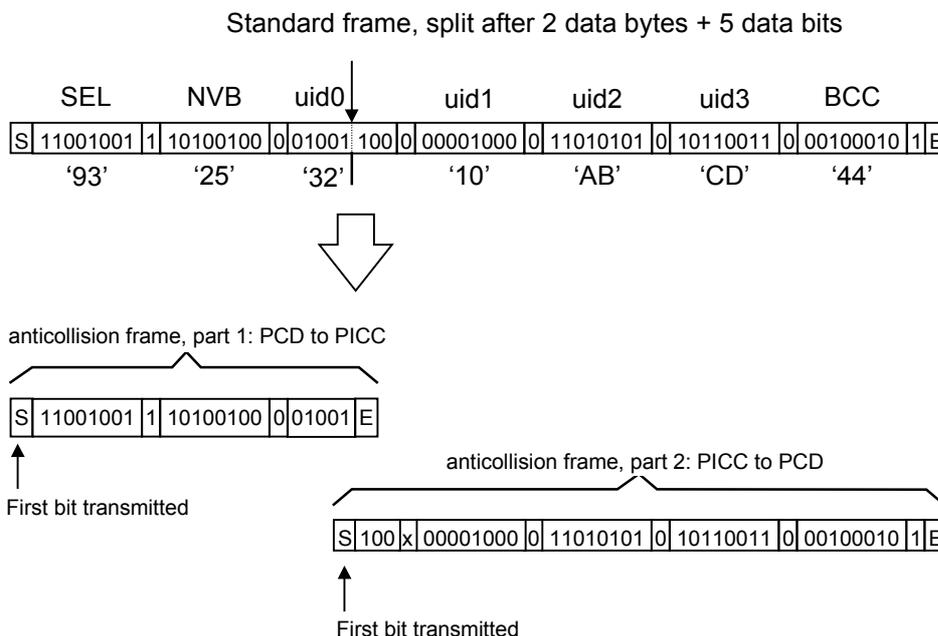
- case FULL BYTE: Split after a complete byte. A parity bit is added after the last data bit of part 1;
- case SPLIT BYTE: Split within a byte. No parity bit is added after the last data bit of part 1.

The following examples for case FULL BYTE and case SPLIT BYTE define the bit organization and order of bit transmission.

NOTE These examples include proper values for NVB and BCC.



**Figure 5 — Bit organization and transmission of bit oriented anticollision frame, case FULL BYTE**



**Figure 6 — Bit organization and transmission of bit oriented anticollision frame, case SPLIT BYTE**

For a SPLIT BYTE, the first parity bit of part 2 shall be ignored by the PCD.

### 6.2.6 CRC\_A

The frame CRC\_A is a function of k data bits, which consist of all the data bits in the frame, excluding parity bits, S and E, and the CRC\_A itself. Since data is encoded in bytes, the number of bits k is a multiple of 8. For error checking, the two CRC\_A bytes are sent in the standard frame, after the bytes and before the E. The CRC\_A is as defined in ISO/IEC 13239, but the initial register content shall be '6363' and the register content shall not be inverted after calculation.

For an example refer to annex B.

### 6.3 PICC states

The following sections provide descriptions of the states for a PICC of Type A specific to the bit collision detection protocol.

The following state diagram takes all possible state transitions caused by commands of this part of ISO/IEC 14443 into account.

PICCs react to valid received frames only. No response is sent when transmission errors are detected.

The following symbols apply for the state diagram shown in Figure 6 below.

- AC                ANTICOLLISION Command (matched UID)
- nAC             ANTICOLLISION Command (not matched UID)
- SELECT         SELECT Command (matched UID)

- nSELECT      SELECT Command (not matched UID)
- DESELECT    DESELECT Command, defined in ISO/IEC 14443-4
- RATS         RATS Command defined in ISO/IEC 14443-4
- Error        transmission error detected

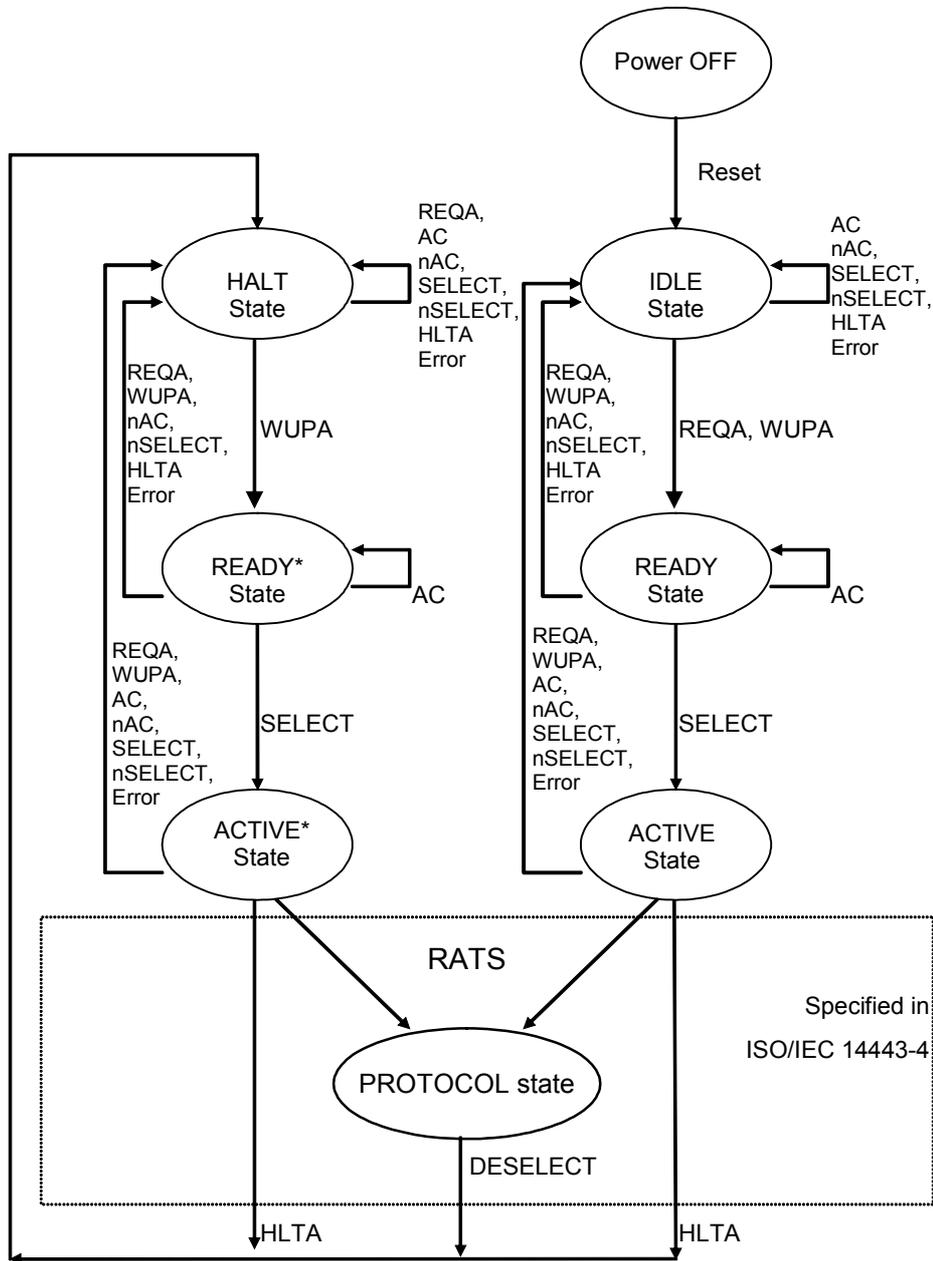


Figure 7 — PICC Type A state diagram

PICCs being compliant with ISO/IEC 14443-3 but not using ISO/IEC 14443-4 may leave the ACTIVE or ACTIVE\* State by proprietary commands.

### **6.3.1 POWER-OFF State**

**Description:**

In the POWER-OFF State, the PICC is not powered due to a lack of carrier energy.

**State exit conditions and transitions:**

If the PICC is in an energizing magnetic field greater than  $H_{min}$  (see ISO/IEC 14443-2), it shall enter its IDLE State within a delay not greater than defined in clause 5.

**6.3.2 IDLE State**

**Description:**

In the IDLE State, the PICC is powered. It listens for commands and shall recognize REQA and WUPA Commands.

**State exit conditions and transitions:**

The PICC enters the READY State after it has received a valid REQA or WUPA Command and transmitted its ATQA.

**6.3.3 READY State**

**Description:**

In the READY State, the bit frame anticollision method shall be applied. Cascade levels are handled inside this state to get the complete UID.

**State exit conditions and transitions:**

The PICC enters the ACTIVE State when it is selected with its complete UID.

**6.3.4 ACTIVE State**

**Description:**

In the ACTIVE State, the PICC listens to any higher layer message.

**State exit conditions and transitions:**

The PICC enters the HALT State when a valid HLTA Command is received.

NOTE In the higher layer protocol, specific commands may be defined to return the PICC to its HALT State.

**6.3.5 HALT State**

**Description:**

In the HALT State, the PICC shall respond only to a WUPA Command.

**State exit conditions and transitions:**

The PICC enters the READY\* State after it has received a valid WUPA Command and transmitted its ATQA.

**6.3.6 READY\* State**

**Description:**

The READY\* State is similar to the READY State. The bit frame anticollision method shall be applied. Cascade levels are handled inside this state to get complete UID.

**State exit conditions and transitions:**

The PICC enters the ACTIVE\* State when it is selected with its complete UID.

**6.3.7 ACTIVE\* State****Description:**

The ACTIVE\* State is similar to the ACTIVE State, the PICC is selected and listens to any higher layer message.

**State exit conditions and transitions:**

The PICC enters the HALT State when a valid HLTA Command is received.

**6.4 Command set**

The commands used by the PCD to manage communication with several PICCs are:

- REQA;
- WUPA;
- ANTICOLLISION;
- SELECT;
- HLTA.

The commands use the byte and frame formats described above.

**6.4.1 REQA and WUPA Commands**

The REQA and WUPA Commands are sent by the PCD to probe the field for PICCs of Type A. They are transmitted within a short frame.

See Figure 6 to check in which cases PICC actually have to answer to these respective commands.

Particularly the WUPA Command is sent by the PCD to put PICCs which have entered the HALT State back into the READY\* State. They shall then participate in further anticollision and selection procedures.

Table 2 shows the coding of REQA and WUPA Commands which use the Short frame format.

**Table 3 — Coding of Short Frame**

| b7               | b6 | b5 | b4 | b3 | b2 | b1 | Meaning                                      |
|------------------|----|----|----|----|----|----|--|
| 0                | 1  | 0  | 0  | 1  | 1  | 0  | '26' = REQA                                  |
| 1                | 0  | 1  | 0  | 0  | 1  | 0  | '52' = WUPA                                  |
| 0                | 1  | 1  | 0  | 1  | 0  | 1  | '35' = Optional timeslot method, see Annex C |
| 1                | 0  | 0  | x  | x  | x  | x  | '40' to '4F' = Proprietary                   |
| 1                | 1  | 1  | 1  | x  | x  | x  | '78' to '7F' = Proprietary                   |
| all other values |    |    |    |    |    |    | RFU  |

A PCD sending an RFU value is not compliant with this standard.

A PICC receiving an RFU value should consider the short frame as an error (see figure 6) and should not send a response.

### 6.4.2 ANTICOLLISION and SELECT Commands

These commands are used during an anticollision loop (see Figure 4 and 5). The ANTICOLLISION and SELECT Commands consist of:

- select code SEL (1 byte);
- number of valid bits NVB (1 byte, for coding see Table 7);
- 0 to 40 data bits of UID CL<sub>n</sub> according to the value of NVB.

SEL specifies the cascade level CL<sub>n</sub>.

The ANTICOLLISION Command is transmitted within bit oriented anticollision frame.

The SELECT Command is transmitted within standard frame

As long as NVB does not specify 40 valid bits, the command is called ANTICOLLISION Command, where the PICC remains in READY or READY\* State.

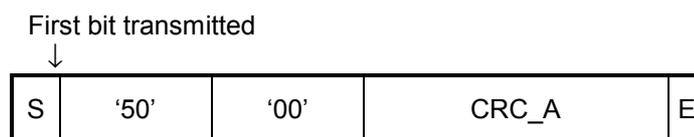
If NVB specifies 40 data bits of UID CL<sub>n</sub> (NVB='70'), a CRC\_A shall be appended. This command is called SELECT Command.

If the PICC has transmitted the complete UID, it transits from READY State to ACTIVE State or from READY\* State to ACTIVE\* State and indicates in its SAK response that UID is complete.

Otherwise, the PICC remains in READY or READY\* State and the PCD shall initiate a new anticollision loop with increased cascade level.

### 6.4.3 HLTA Command

The HLTA Command consists of two bytes followed by CRC\_A and shall be transmitted within Standard Frame.



**Figure 8 — Standard frame containing HLTA Command**

If the PICC responds with any modulation during a period of 1 ms after the end of the frame containing the HLTA Command, this response shall be interpreted as 'not acknowledge'.

NOTE The PCD should apply a additional marginal waiting time of 0.1 ms.

## 6.5 Select sequence

The purpose of the select sequence is to get the UID from one PICC and to select this PICC for further communication.

6.5.1 Select sequence flowchart

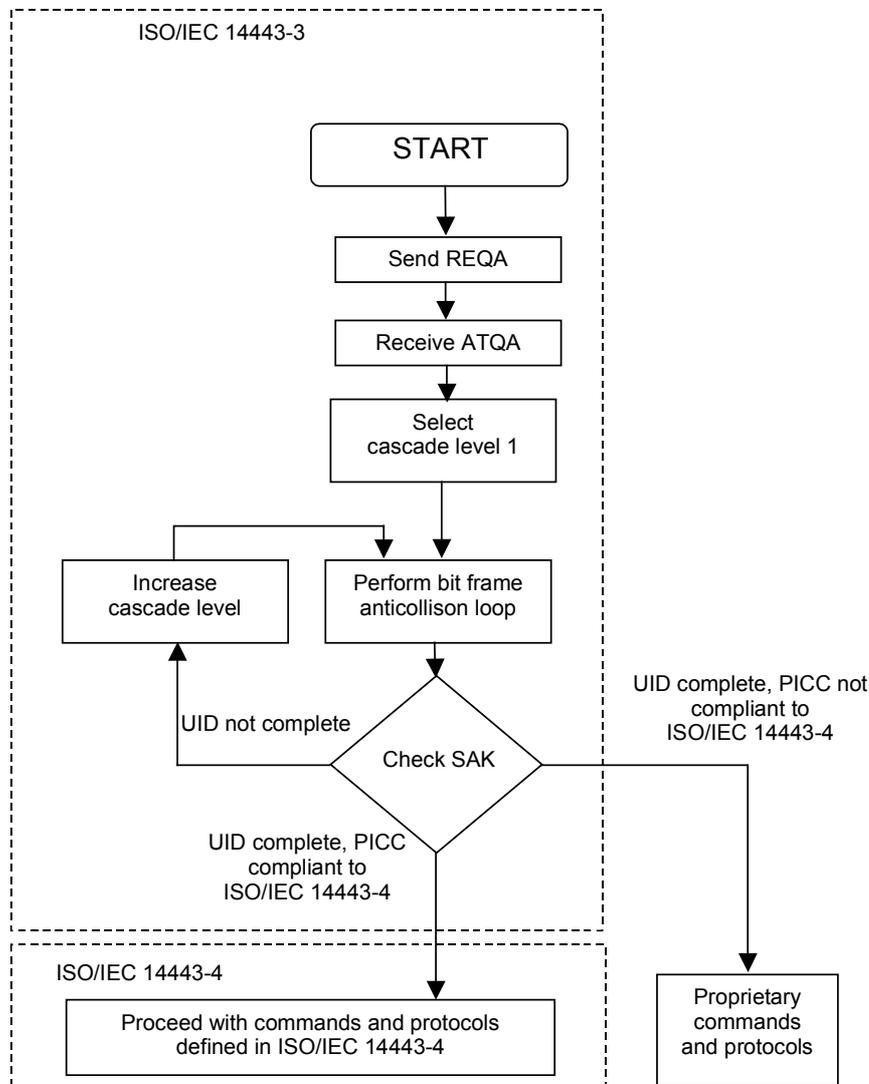


Figure 9 — Initialization and anticollision flowchart for PCD

NOTE PICCs may use ATQA bit combinations of b9 to b12 for indication of proprietary methods.

PICCs that do not support the mandatory bit frame anticollision are not compliant with this standard.

6.5.2 ATQA - Answer To Request

After a REQA Command is transmitted by the PCD, all PICCs in the IDLE State shall respond synchronously with ATQA.

After a WUPA Command is transmitted by the PCD, all PICCs in the IDLE or HALT State shall respond synchronously with ATQA.

The PCD shall detect any collision that may occur when multiple PICCs respond.

An example is given in annex A.

### 6.5.2.1 Coding of ATQA

**Table 4 — Coding of ATQA**

| MSB |     |     |     |                    |     |     |    |                       |    |     | LSB                     |    |    |    |    |
|-----|-----|-----|-----|--------------------|-----|-----|----|-----------------------|----|-----|-------------------------|----|----|----|----|
| b16 | b15 | b14 | b13 | b12                | b11 | b10 | b9 | b8                    | b7 | b6  | b5                      | b4 | b3 | b2 | b1 |
| RFU |     |     |     | Proprietary coding |     |     |    | UID size<br>bit frame |    | RFU | Bit frame anticollision |    |    |    |    |

All RFU bits shall be set to 0.

A PICC sending an ATQA with  $(b8,b7) = (11)_b$  or  $(b16 \text{ to } b13) \neq (0000)_b$  or  $b6 \neq (0)_b$  is not compliant with this standard.

A PCD detecting a collision in any bit of  $(b16 \text{ to } b1)$  shall commence with the first step of the anticollision loop (see clause 6.4.3.1 *renumbered to 6.5.3.1*).

The PCD shall commence with the first step of the anticollision loop regardless of any value in the proprietary field  $b12 \text{ to } b9$ .

A PCD receiving  $(b8,b7) = (11)_b$  or  $(b16 \text{ to } b13) \neq (0000)_b$  or  $b6 \neq (0)_b$  should ignore the values and should commence with the first step of the anticollision loop (see clause 6.4.3.1 *renumbered to 6.5.3.1*).

### 6.5.2.2 Coding rules for bit frame anticollision

- Rule 1: Bits  $b7$  and  $b8$  code the UID size (single, double or triple, see Table 4).
- Rule 2: One out of the five bits  $b1$ ,  $b2$ ,  $b3$ ,  $b4$  or  $b5$  shall be set to  $(1)_b$  to indicate bit frame anticollision (see Table 5)

**Table 5 — Coding of  $b7$ ,  $b8$  for bit frame anticollision**

| b8 | b7 | Meaning          |
|----|----|------------------|
| 0  | 0  | UID size: single |
| 0  | 1  | UID size: double |
| 1  | 0  | UID size: triple |
| 1  | 1  | RFU              |

**Table 6 — Coding of  $b1$  -  $b5$  for bit frame anticollision**

| b5 | b4 | b3 | b2 | b1 | Meaning                 |
|----|----|----|----|----|-------------------------|
| 1  | 0  | 0  | 0  | 0  | bit frame anticollision |
| 0  | 1  | 0  | 0  | 0  | bit frame anticollision |
| 0  | 0  | 1  | 0  | 0  | bit frame anticollision |
| 0  | 0  | 0  | 1  | 0  | bit frame anticollision |
| 0  | 0  | 0  | 0  | 1  | bit frame anticollision |

**6.5.3 Anticollision and Select**

**6.5.3.1 Anticollision loop within each cascade level**

The following algorithm shall apply to the anticollision loop:

|         |   |
|---------|---|
| Step 1  | The PCD shall assign SEL with the code for the selected anticollision type and cascade level.   |
| Step 2  | The PCD shall assign NVB with the value of '20'.<br>NOTE This value defines that the PCD will transmit no part of UID CLn. Consequently this command forces all PICCs in the field to respond with their complete UID CLn.  |
| Step 3  | The PCD shall transmit SEL and NVB.   |
| Step 4  | All PICCs in the field shall respond with their complete UID CLn.   |
| Step 5  | Assuming the PICCs in the field have unique serial numbers, then if more than one PICC responds, a collision occurs. If no collision occurs, steps 6 to 10 shall be skipped.  |
| Step 6  | The PCD shall recognize the position of the first collision.  |
| Step 7  | The PCD shall assign NVB with a value that specifies the number of valid bits of UID CLn. The valid bits shall be part of the UID CLn that was received before a collision occurred followed by a (0)b or (1)b, decided by the PCD. A typical implementation adds a (1)b. |
| Step 8  | The PCD shall transmit SEL and NVB, followed by the valid bits.   |
| Step 9  | Only PICCs of which the part of UID CLn is equal to the valid bits transmitted by the PCD shall transmit their remaining bits of the UID CLn.   |
| Step 10 | If further collisions occur, steps 6 to 9 shall be repeated. The maximum number of loops will be 32.  |
| Step 11 | If no further collision occurs, the PCD shall assign NVB with the value of '70'.<br>NOTE This value defines that the PCD will transmit the complete UID CLn.  |
| Step 12 | The PCD shall transmit SEL and NVB, followed by all 40 bits of UID CLn, followed by CRC_A checksum.   |
| Step 13 | The PICC which UID CLn matches the 40 bits shall respond with its SAK.  |
| Step 14 | If the UID is complete, the PICC shall transmit SAK with cleared cascade bit and shall transit from READY State to ACTIVE State or from READY* State to ACTIVE* State.  |
| Step 15 | The PCD shall check if the cascade bit of SAK is set to decide whether further anticollision loops with increased cascade level shall follow.   |

If the UID of a PICC is complete and known by the PCD, the PCD may skip step 2 - step 10 to select this PICC without performing the anticollision loop.

NOTE Figure 9 explains steps 1 to 13.

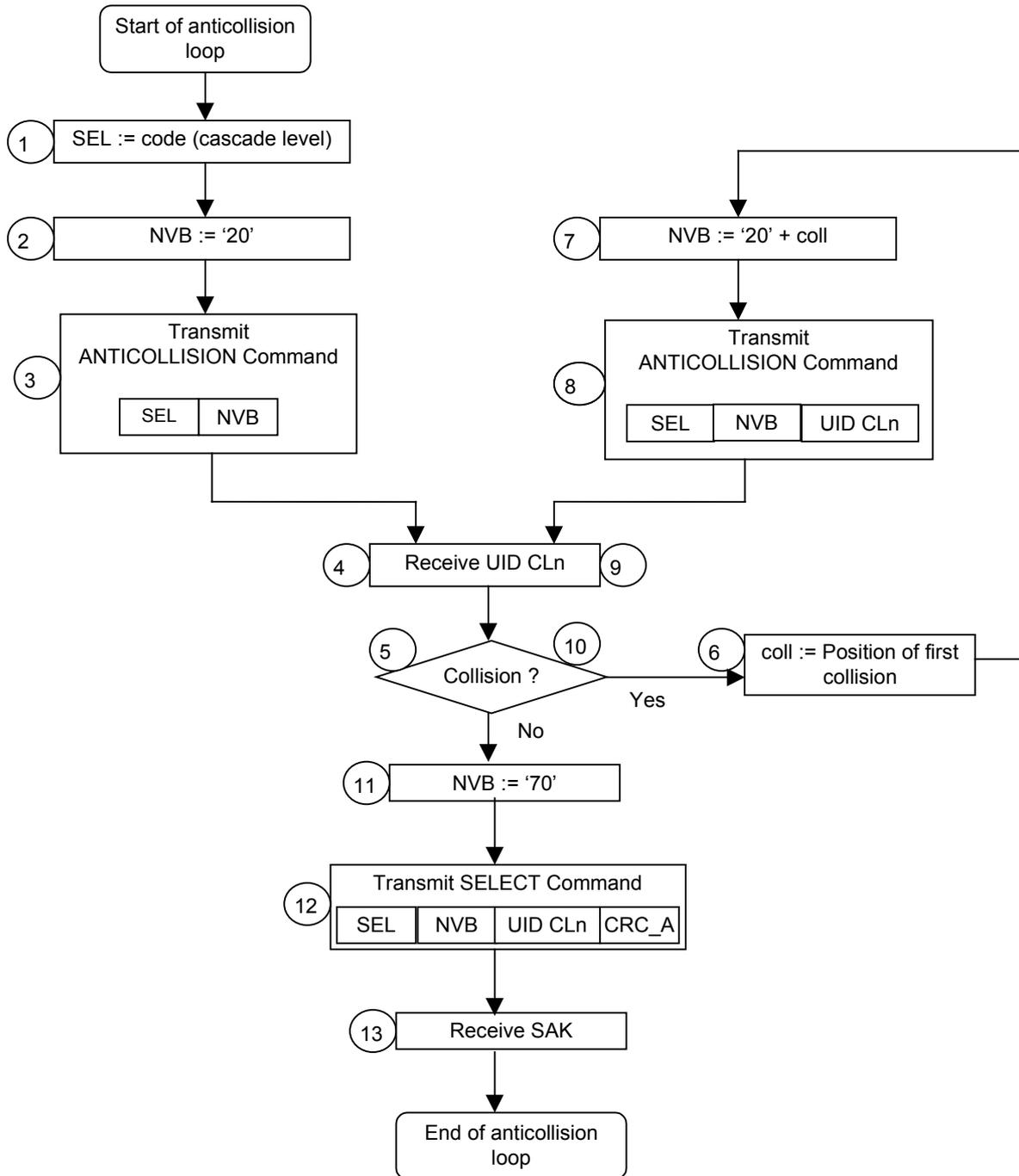


Figure 10 — Anticollision loop, flowchart for PCD

NOTE The circled numbers correspond to the steps of the algorithm.

### 6.5.3.2 Coding of SEL (Select code)

Length: 1 byte

Possible values: '93', '95', '97'

**Table 7 — Coding of SEL**

| b8 | b7 | b6 | b5 | b4                                   | b3 | b2 | b1 | Meaning                      |
|----|----|----|----|--------------------------------------|----|----|----|------------------------------|
| 1  | 0  | 0  | 1  | 0                                    | 0  | 1  | 1  | '93': Select cascade level 1 |
| 1  | 0  | 0  | 1  | 0                                    | 1  | 0  | 1  | '95': Select cascade level 2 |
| 1  | 0  | 0  | 1  | 0                                    | 1  | 1  | 1  | '97': Select cascade level 3 |
| 1  | 0  | 0  | 1  | other values except those here above |    |    |    | RFU                          |

**6.5.3.3 Coding of NVB (Number of Valid Bits)**

Length: 1 byte

The upper 4 bits are called “Byte count” and specify the integer part of the number of all valid data bits transmitted by the PCD (including SEL and NVB) divided by 8. Consequently, the minimum value of “Byte count” is 2 and the maximum value is 7.

The lower 4 bits are called “bit count” and specify the number of all valid data bits transmitted by the PCD (including SEL and NVB) modulo 8.

**Table 8 — Coding of NVB**

| b8 | b7 | b6 | b5 | Meaning        |
|----|----|----|----|----------------|
| 0  | 0  | 1  | 0  | Byte count = 2 |
| 0  | 0  | 1  | 1  | Byte count = 3 |
| 0  | 1  | 0  | 0  | Byte count = 4 |
| 0  | 1  | 0  | 1  | Byte count = 5 |
| 0  | 1  | 1  | 0  | Byte count = 6 |
| 0  | 1  | 1  | 1  | Byte count = 7 |

| b4 | b3 | b2 | b1 | Meaning       |
|----|----|----|----|---------------|
| 0  | 0  | 0  | 0  | bit count = 0 |
| 0  | 0  | 0  | 1  | bit count = 1 |
| 0  | 0  | 1  | 0  | bit count = 2 |
| 0  | 0  | 1  | 1  | bit count = 3 |
| 0  | 1  | 0  | 0  | bit count = 4 |
| 0  | 1  | 0  | 1  | bit count = 5 |
| 0  | 1  | 1  | 0  | bit count = 6 |
| 0  | 1  | 1  | 1  | bit count = 7 |

The PCD shall set NVB only to values defined in Table 7 except that for byte counts 6 and 7 only bit count of 0 is allowed. A PCD setting NVB to any forbidden value is not compliant with this standard.

A PCD setting the byte count (b8 to b5) to any value outside the range 2 to 7 is not compliant with this standard.

A PCD setting the bit count (b4 to b1) > 7 for byte count equal 2 to 5 or setting the bit count (b4 to b1) to any value other than 0 for byte count equal 6 or 7 is not compliant with this standard.

**6.5.3.4 Coding of SAK (Select acknowledge)**

SAK is transmitted by the PICC when NVB has specified 40 valid data bits and when all these data bits match with UID CLn.

|                        |   |
|------------------------|---|
| 1 <sup>st</sup> byte   | 2 <sup>nd</sup> , 3 <sup>rd</sup> bytes |
| <b>SAK</b><br>(1 byte) | <b>CRC_A</b><br>(2 bytes)               |
| MSB    LSB             | MSB                      LSB            |

Figure 11 — Select acknowledge (SAK)

The coding of bits b3 (cascade bit) and b6 is given in Table 8.

Table 9 — Coding of SAK

| b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | Meaning  |
|----|----|----|----|----|----|----|----|--|
| x  | x  | x  | x  | x  | 1  | x  | x  | Cascade bit set: UID not complete                        |
| x  | x  | 1  | x  | x  | 0  | x  | x  | UID complete,<br>PICC compliant with ISO/IEC 14443-4     |
| x  | x  | 0  | x  | x  | 0  | x  | x  | UID complete,<br>PICC not compliant with ISO/IEC 14443-4 |

NOTE: 'x' in this table means "a don't care value."

For b3 = (1)b the PCD shall ignore any other bit of SAK. For b3 = (0)b the PCD shall interpret b6 and shall ignore any of the remaining bits of SAK. A PCD behaving differently is not compliant with this standard.

NOTE: When b3 is set to 1, all other bits of SAK should be set to 0

#### 6.5.4 UID contents and cascade levels

The UID consists of 4, 7 or 10 UID bytes. Consequently, the PICC shall handle up to 3 cascade levels to get all UID bytes. Within each cascade level, a part of UID shall be transmitted to the PCD. According to the cascade level, three types of UID size are defined. This UID size shall be consistent with Table 4.

Table 10 — UID size

| Cascade level | UID size | Number of UID bytes |
|---------------|----------|---------------------|
| 1             | single   | 4                   |
| 2             | double   | 7                   |
| 3             | triple   | 10                  |

The UID is a fixed unique number or a random number which is dynamically generated by the PICC. The first byte (uid0) of the UID assigns the content of the following bytes of the UID.

Table 11 — Single size UIDs

| uid0        | Description  |
|-------------|--|
| '08'        | uid1 to uid3 is a random number which is dynamically generated |
| 'x0' - 'x7' | Proprietary number   |
| 'x9' - 'xE' | Proprietary number   |

|  |     |
|--|-----|
| '18', '28', '38',<br>'48', '58', '68',<br>'78', '98', 'A8',<br>'B8', 'C8', 'D8',<br>'E8', 'F8' | RFU |
| 'xF'   | RFU |

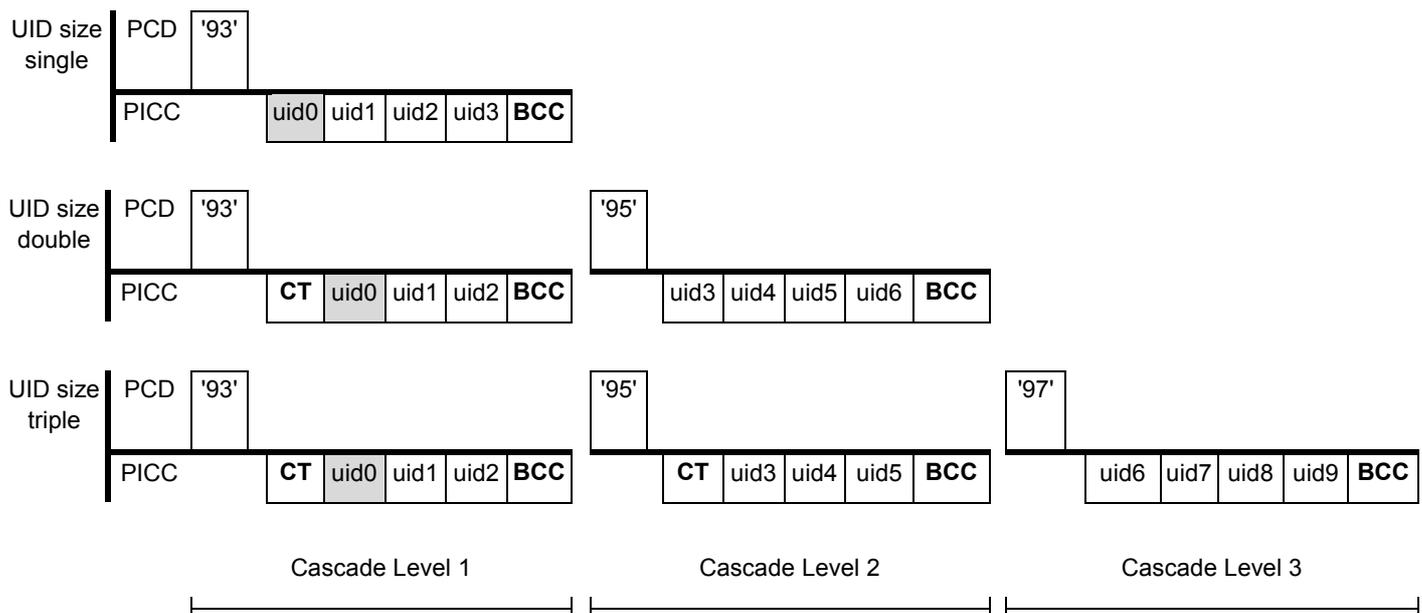
A random UID shall be generated only on state transition from "Power OFF" to "IDLE" state.

The value '88' of the cascade tag CT shall not be used for uid0 in single size UID.

**Table 12 — Double and triple size UIDs**

| uid0  | Description   |
|---|---|
| Manufacturer ID according to ISO/IEC 7816-6/AM1*  | Each manufacturer is responsible for the uniqueness of the value of the other bytes of the unique number. |
| * The values '81' to 'FE', which are marked for 'Private use' in ISO/IEC 7816-6/AM1 shall not be allowed in this context. |   |

The value '88' of the cascade tag CT shall not be used for uid3 in double size UID.



**Figure 12 — Usage of cascade levels**

NOTE The purpose of the cascade tag is to force a collision with PICCs that have a smaller UID size.

The following algorithm shall apply to the PCD to get the complete UID:

|        |   |
|--------|---|
| Step 1 | The PCD selects cascade level 1   |
| Step 2 | The anticollision loop shall be performed   |
| Step 3 | The PCD shall check the cascade bit of SAK  |
| Step 4 | If the cascade bit is set, the PCD shall increase the cascade level and initiate a new anticollision loop |

"A PICC sending uid0 with an RFU value is not compliant with this standard. A PICC sending a proprietary number shall fulfil all other requirements of the anticollision sequence including CT otherwise such a PICC is not compliant with this standard.

During the anticollision, the PCD shall regard uid0 with RFU or proprietary values as a regular uid0.

## 7 Type B – Initialization and anticollision

This section describes the initialization and collision detection protocol applicable for PICCs of Type B.

### 7.1 Character, frame format and timing

This section defines the character, frame format and timing used during communication initialization and anticollision for PICCs of Type B. For bit representation and coding refer to ISO/IEC 14443-2.

#### 7.1.1 Character transmission format

Bytes are transmitted and received between PICCs and a PCD by characters, the format of which during the Anti-collision sequence is as follows:

- 1 start bit at logic "0";
- 8 data bits transmitted, LSB first;
- 1 stop bit at logic "1".

The transmission of one byte is performed with a character requiring 10 etu as illustrated below.

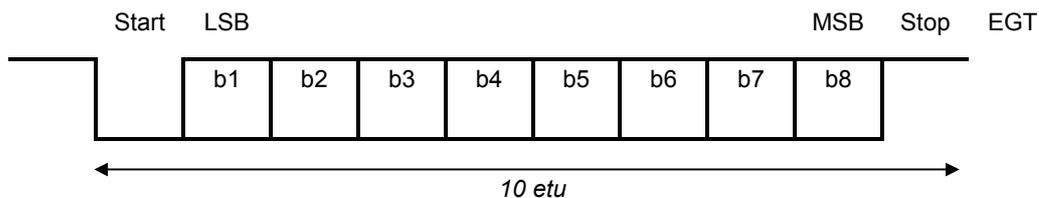


Figure 13 — Character format

From PCD to PICC, bit boundaries within a character shall occur as defined in Table 13, where  $n$  is the number of bit boundaries after the start bit falling edge ( $1 \leq n \leq 9$ ).

Table 13 Bit boundaries from PCD to PICC

|  | PCD to PICC bit rate     |                          |                          |                          |
|--|--------------------------|--------------------------|--------------------------|--------------------------|
|  | $fc/128$                 | $fc/64$                  | $fc/32$                  | $fc/16$                  |
| Bit boundaries from PCD to PICC for the falling edge | $n \text{ etu} \pm 8/fc$ | $n \text{ etu} \pm 1/fc$ | $n \text{ etu} \pm 1/fc$ | $n \text{ etu} \pm 1/fc$ |
| Bit boundaries from PCD to PICC for the rising edge  | $n \text{ etu} \pm 8/fc$ | $n \text{ etu} \pm 4/fc$ | $n \text{ etu} \pm 2/fc$ | $n \text{ etu} \pm 1/fc$ |

#### 7.1.2 Character separation

A character is separated from the next one by the extra guard time EGT.

The EGT between 2 consecutive characters sent by the PCD to the PICC shall be between 0 and 6 etu (not necessarily an integer number of etu).

The EGT between 2 consecutive characters sent by the PICC to the PCD shall be between 0 and 2 etu (not necessarily an integer number of etu).

**Table 14 EGT from PCD to PICC**

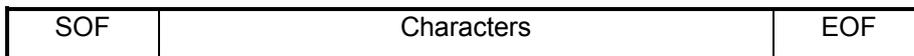
| EGT PCD to PICC |           |       |       |
|-----------------|-----------|-------|-------|
| PCD             |           | PICC  |       |
| Min.            | Max.      | Min.  | Max   |
| 0 etu           | 5,875 etu | 0 etu | 6 etu |

**Table 15 EGT from PICC to PCD**

| EGT PICC to PCD |           |       |       |
|-----------------|-----------|-------|-------|
| PCD             |           | PICC  |       |
| Min.            | Max.      | Min.  | Max   |
| 0 etu           | 2,125 etu | 0 etu | 2 etu |

**7.1.3 Frame format**

PCDs and PICCs shall send characters as frames. The frame is normally delimited by SOF and by EOF.

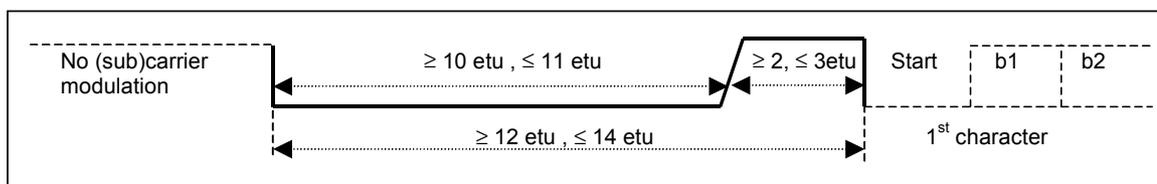


**Figure 14 — Frame Format**

**7.1.4 SOF**

SOF is composed of:

- one falling edge;
- followed by 10 etu with a logic "0";
- followed by one single rising edge located anywhere within the following etu;
- followed by at least 2 etu (but no more than 3 etu) with a logic "1".



**Figure 15 — SOF**

**Table 16 SOF of PCD transmission**

|                     | PCD              |                   | PICC              |                  |
|---------------------|------------------|-------------------|-------------------|------------------|
|                     | Min.             | Max.              | Min.              | Max              |
| <b>PCD SOF low</b>  | 10 etu           | 11 etu + 1/16 etu | 10 etu - 1/16 etu | 11 etu + 1/8 etu |
| <b>PCD SOF high</b> | 2 etu - 1/16 etu | 3 etu + 1/16 etu  | 2 etu - 1/8 etu   | 3 etu + 1/8 etu  |

**Table 17 SOF low of PICC transmission**

| Bit rate                    | PCD                |                    | PICC               |                    |
|-----------------------------|--------------------|--------------------|--------------------|--------------------|
|                             | Min.               | Max.               | Min.               | Max                |
| <b><math>f_c/128</math></b> | 10 etu - $1/f_s$   | 11 etu + $1/f_s$   | 10 etu - $0,5/f_s$ | 11 etu + $0,5/f_s$ |
| <b><math>f_c/64</math></b>  | 10 etu - $0,5/f_s$ | 11 etu + $0,5/f_s$ | 10 etu             | 11 etu             |
| <b><math>f_c/32</math></b>  | 10 etu             | 11 etu             | 10 etu             | 11 etu             |
| <b><math>f_c/16</math></b>  | 10 etu             | 11 etu             | 10 etu             | 11 etu             |

**Table 18 SOF high of PICC transmission**

| Bit rate                    | PCD               |                    | PICC              |                   |
|-----------------------------|-------------------|--------------------|-------------------|-------------------|
|                             | Min.              | Max.               | Min.              | Max               |
| <b><math>f_c/128</math></b> | 2 etu - $1/f_s$   | 3 etu + $1/f_s$    | 2 etu - $0,5/f_s$ | 3 etu + $0,5/f_s$ |
| <b><math>f_c/64</math></b>  | 2 etu - $0,5/f_s$ | 3 etu + $0,5/f_s$  | 2 etu             | 3 etu             |
| <b><math>f_c/32</math></b>  | 2 etu             | 3 etu <sub>c</sub> | 2 etu             | 3 etu             |
| <b><math>f_c/16</math></b>  | 2 etu             | 3 etu              | 2 etu             | 3 etu             |

NOTE all values in tables 17 and 18 comply with 9.2.4 of ISO/IEC 14443-2

**7.1.5 EOF**

EOF is composed of:

- one falling edge;
- followed by 10 etu with a logic "0";
- followed by one single rising edge located anywhere within the following etu.

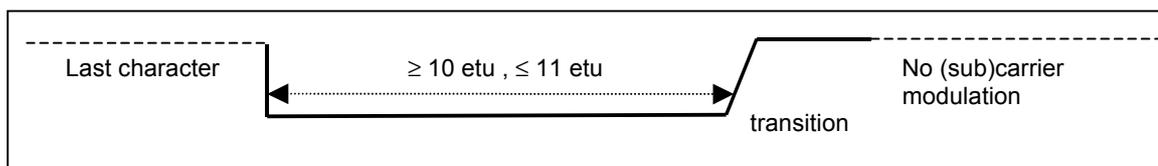


Figure 16 — EOF

NOTE The probability of receiving a false EOF is low and corresponds to the transmission of a '00' character with a wrong reception of the stop bit.

Table 19 EOF of PCD transmission

| EOF low of PCD |                   |                   |                  |
|----------------|-------------------|-------------------|------------------|
| PCD            |                   | PICC              |                  |
| Min.           | Max.              | Min.              | Max              |
| 10 etu         | 11 etu + 1/16 etu | 10 etu - 1/16 etu | 11 etu + 1/8 etu |

Table 20 EOF of PICC transmission

| Bit rate       | PCD                     |                         | PICC                    |                         |
|----------------|-------------------------|-------------------------|-------------------------|-------------------------|
|                | Min.                    | Max.                    | Min.                    | Max                     |
| <i>fc</i> /128 | 10 etu - 1/ <i>fs</i>   | 11 etu + 1/ <i>fs</i>   | 10 etu - 0,5/ <i>fs</i> | 11 etu + 0,5/ <i>fs</i> |
| <i>fc</i> /64  | 10 etu - 0,5/ <i>fs</i> | 11 etu + 0,5/ <i>fs</i> | 10 etu                  | 11 etu                  |
| <i>fc</i> /32  | 10 etu                  | 11 etu                  | 10 etu                  | 11 etu                  |
| <i>fc</i> /16  | 10 etu                  | 11 etu                  | 10 etu                  | 11 etu                  |

NOTE all values in table 20 comply with 9.2.4 of ISO/IEC 14443-2

7.1.6 Timing before the PICC SOF

PICC start of communication after a PCD data transmission shall respect the timing defined in Figure 16.

The default minimum values of TR0 and TR1 are defined in ISO/IEC 14443-2 and may be reduced by the PCD, see 7.10.3.

The maximum value of TR0 is:

- 256/*fs* for ATQB
- 65536/*fc* for S(DESELECT) (see ISO/IEC 14443-4 subclause 8.1)
- (256/*fs*)\*2<sup>FWI</sup> - TR1 for all other frames (see subclause 7.9.4.3).

The maximum value of TR1 is 200/*fs*.

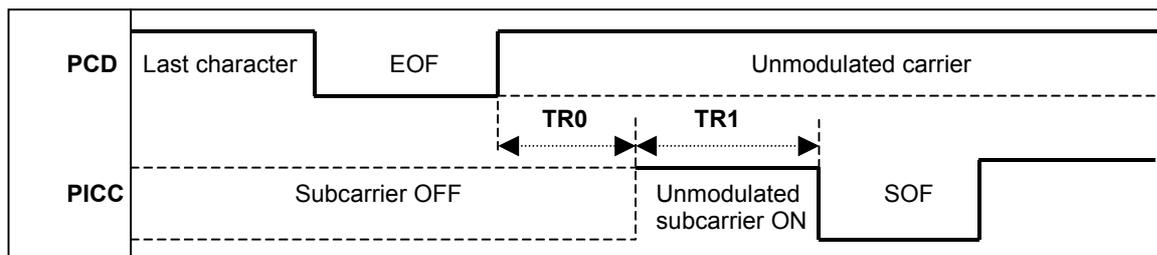


Figure 17 — PICC subcarrier SOF

A PICC may turn on the subcarrier only if it intends to begin transmitting information.

The minimal and maximal values of TR0 and TR1 are applicable to PICCs. PCDs shall accept minimal and maximal values of TR0 and TR1 with a margin of  $1/f_s$ .

7.1.7 Timing before the PCD SOF

PCD start of communication after a PICC data transmission and EOF shall respect the timing in Figure 17.

The PICC shall turn off its subcarrier after the transmission of the EOF. The subcarrier signal shall:

- not be stopped before the end of the EOF;
- be stopped no later than 2 etu after the end of the EOF.

The minimum value of TR2 is coded in ATQB by Protocol\_type in “Protocol Info” field (see 7.9.4.4).

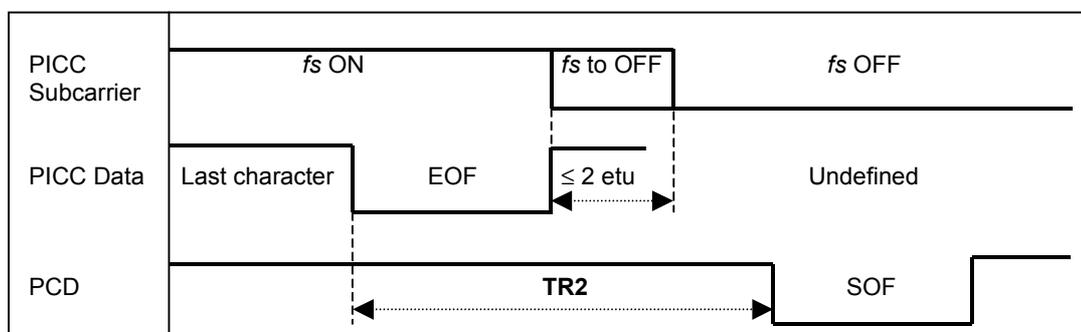


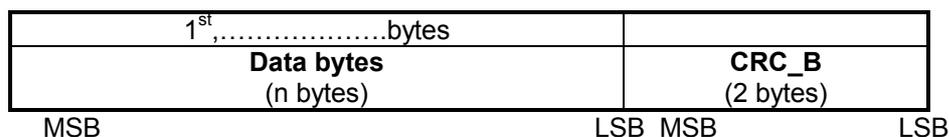
Figure 18 — PICC to PCD EOF

Table 21 Timing before PCD SOF

|                  | PCD   |                         | PICC  |       |
|------------------|-------|-------------------------|-------|-------|
|                  | Min   | Max                     | Min   | Max   |
| <b>fs to OFF</b> | 0 etu | $2 \text{ etu} + 1/f_s$ | 0 etu | 2 etu |

The minimal value of TR2 is applicable to PICCs. PCDs shall respect minimal value of TR2 with a margin of  $100/f_c$ .

## 7.2 CRC\_B



**Figure 19 — Position of a CRC\_B within a frame**

A frame shall only be considered correct if it is received with a valid CRC\_B value.

The frame CRC\_B is a function of k data bits, which consist of all the data bits in the frame, excluding start bits, stop bits, delays between bytes, SOF and EOF, and the CRC\_B itself. Since data is encoded in bytes, the number of bits k is a multiple of 8.

For error checking, the two CRC\_B bytes are included in the frame, after the data bits and before the EOF. The CRC\_B is as defined in ISO/IEC 13239. The initial register content shall be all ones: 'FFFF'.

For example refer to annex B.

## 7.3 Anticollision sequence

An anticollision sequence is managed by the PCD through a set of commands detailed in this section.

The PCD is the master of the communication with one or more PICCs. It initiates PICC communication activity by issuing a REQB Command to prompt for PICCs to respond.

During the anticollision sequence it may happen that two or more PICCs respond simultaneously: this is a collision. The command set allows the PCD to handle sequences to separate PICC transmissions in time. The PCD may repeat its anticollision procedure until it finds all PICCs in the operating volume.

Having completed the anticollision sequence, PICC communication will be under control of the PCD, allowing only one PICC to talk at a time.

The anticollision scheme is based on definition of timeslots in which PICCs are invited to answer with minimum identification data. The number of slots is parameterized in the REQB/WUPB and can vary from one to some integer number. PICC response probability in each timeslot is also controllable. PICCs are allowed to answer only once in the anticollision sequence.

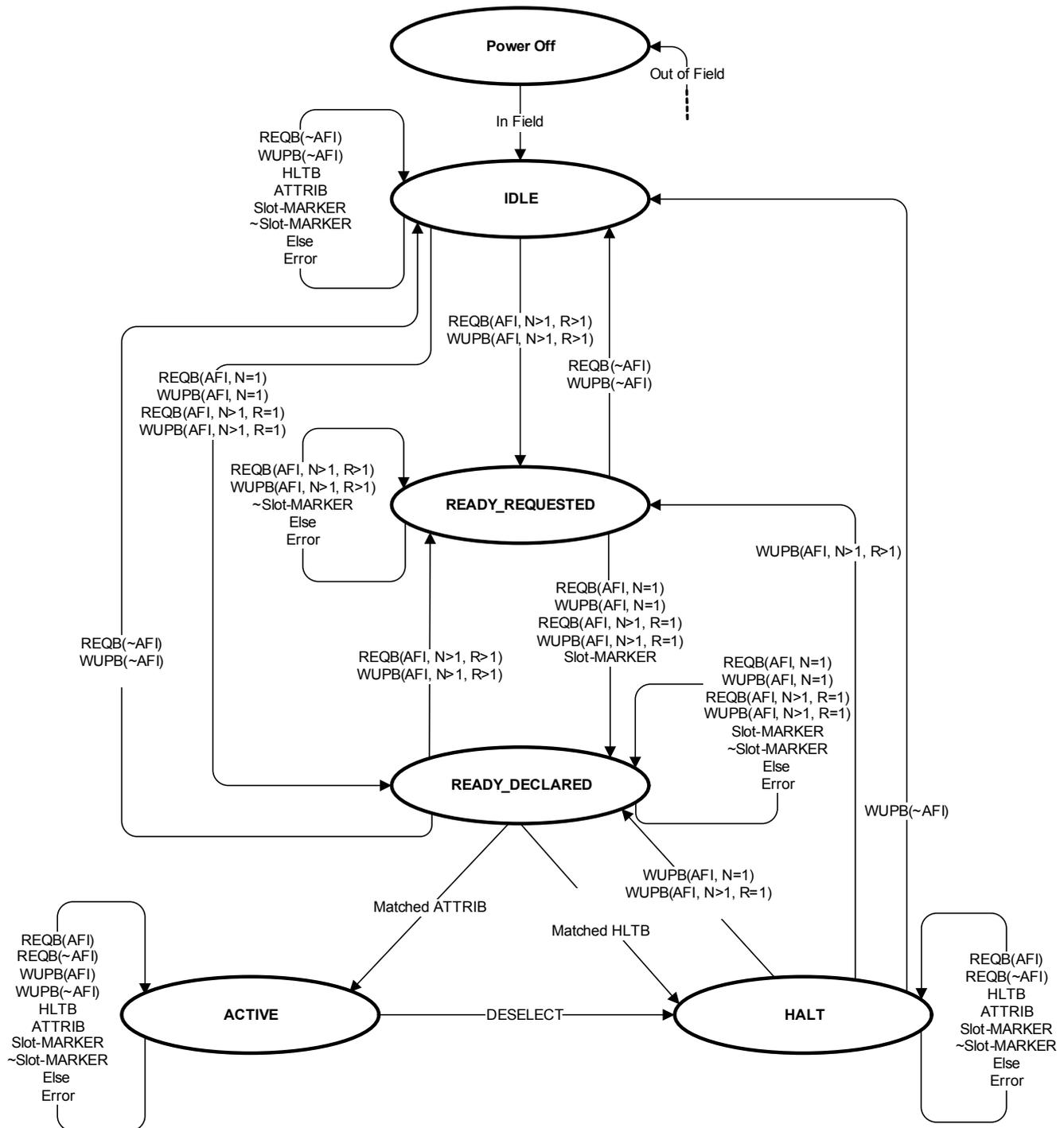
Consequently, even in case of multiple PICCs present in the PCD field, there will probably be a slot in which only one PICC answers and where the PCD is able to capture the identification data. Based on the identification data the PCD is able to establish a communication channel with the identified PICC.

An anticollision sequence allows selection of one or more PICCs for further communication at any time.

## 7.4 PICC states description

Different states and transition conditions between states describe the PICC detailed behaviour during the anticollision sequence.

**Figure 20 — State diagram**



Note:  
 REQB(AFI)/WUPB(AFI) -> Commands with matched AFI  
 REQB(~AFI)/WUPB(~AFI) -> Commands with unmatched AFI  
 Slot-Marker -> Command with matched Slot marker  
 ~Slot-Marker -> Command with unmatched Slot marker  
 Else -> any other command

7.4.1 Initialization and anticollision flowchart

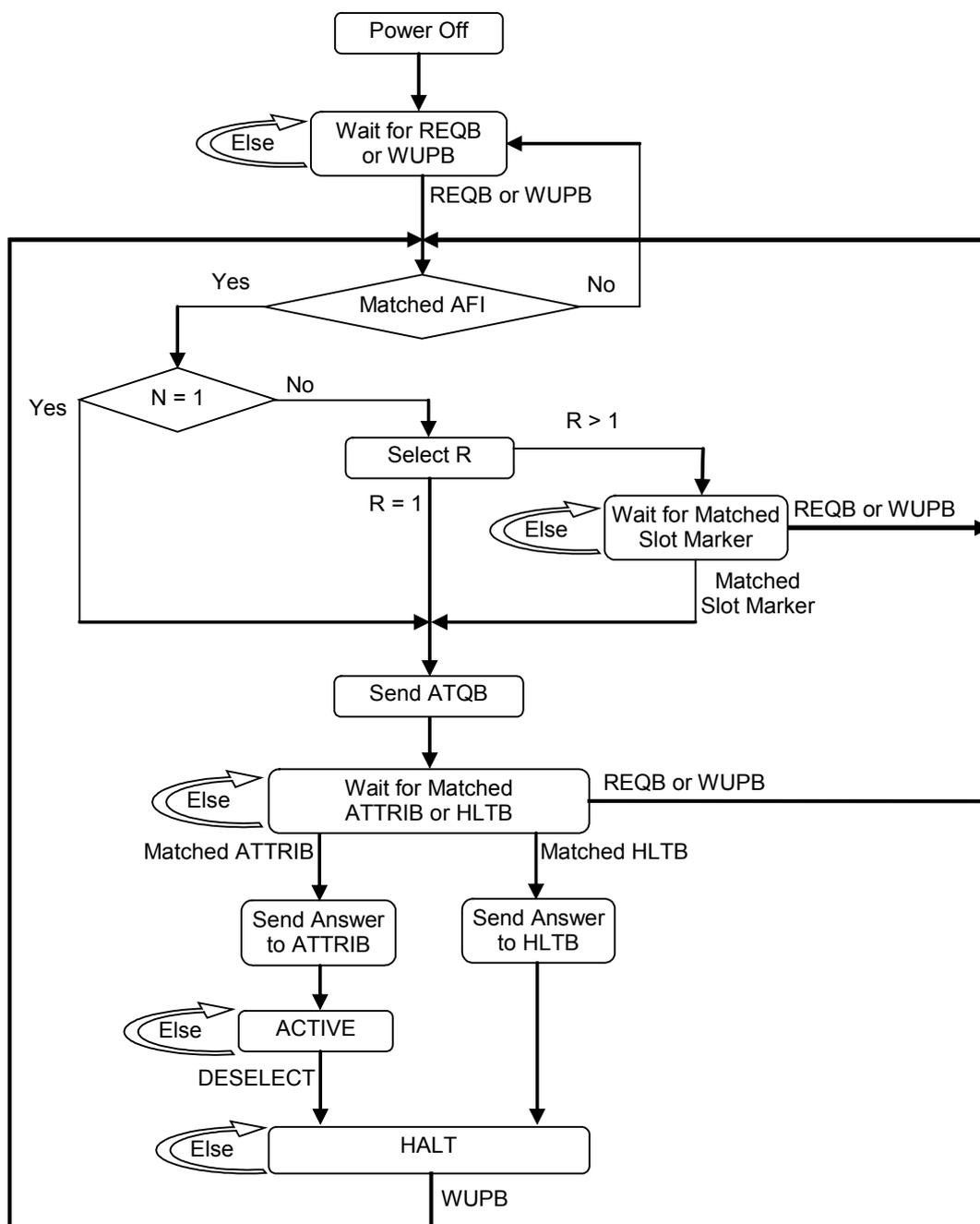


Figure 21 — PICC initialization and anticollision flowchart

NOTE R is a random number chosen by the PICC in the range from 1 to N (for coding of N see 7.7.4).

7.4.2 General statement for state description and transitions

To any state the following shall apply:

- the PICC shall return to POWER\_OFF State if the RF field disappears.

To any state specific to the anticollision sequence (except ACTIVE State) the following shall apply:

- default communication parameters as defined in ISO/IEC 14443-2 and in the previous sections shall be used;
- the PICC shall not emit subcarrier except to transmit response frames as specified in the previous sections;
- if a frame from the PCD is valid (correct CRC\_B), the PICC shall perform the required action and/or response depending on its state;
- as in anticollision commands the first 3 bits of the data in a frame are (101)<sub>b</sub> (3 first bits of anticollision Prefix byte) the PICC shall not answer to any command frame not starting with (101)<sub>b</sub>;
- the PICC shall only react to valid frames received (no response sent when transmission errors are detected).

### 7.4.3 POWER-OFF State

#### Description:

In the POWER-OFF State, the PICC is not powered due to a lack of carrier energy.

#### State exit conditions and transitions:

If the PICC is in an energizing magnetic field greater than H<sub>min</sub> (see ISO/IEC 14443-2), it shall enter its IDLE State within a delay not greater than defined in clause 5.

### 7.4.4 IDLE State

#### Description:

In the IDLE State, the PICC is powered. It listens for frames and shall recognize REQB and WUPB messages.

#### State exit conditions and transitions:

On reception of a valid REQB or WUPB Command frames the PICC moves onto the READY REQUESTED or READY DECLARED sub-state. (Valid REQB/WUPB means valid frame with REQB/WUPB Command and matched AFI. See REQB/WUPB Command specification for more details).

### 7.4.5 READY-REQUESTED sub-state

#### Description:

In the READY-REQUESTED sub-state, the PICC is powered and has received a valid REQB or WUPB Commands with a control parameter N. The PICC calculate a random number R which is used to control its subsequent operation as described in clause 7.6.

#### State exit conditions and transitions:

See 7.6.for details.

#### Specific remarks:

In this state the ATQB has not yet been sent..

#### 7.4.6 READY-DECLARED sub-state

##### Description:

In the READY-DECLARED sub-state, the PICC is powered and has sent its ATQB corresponding to the last valid REQB/WUPB message received.

It listens to frames and shall recognize REQB/WUPB, ATTRIB and HLTB commands.

##### State exit conditions and transitions:

On reception of a valid ATTRIB Command the PICC shall enter the ACTIVE State if the PUPI in the ATTRIB Command matches the PICC PUPI.

If the PUPI in the ATTRIB Command does not match the PICC PUPI the PICC remains in the READY-DECLARED sub-state.

On reception of a valid REQB/WUPB Command frame the same conditions and transitions apply as on reception of a valid REQB/WUPB Command frame in the IDLE State.

On reception of a matched HLTB Command the PICC shall enter the HALT State.

#### 7.4.7 ACTIVE State

##### Description:

The PICC is powered and has entered a higher layer mode since a Card Identifier (CID) has been assigned to this PICC through the ATTRIB Command.

The PICC listens to any higher layer message properly formatted (proper CID and valid CRC<sub>B</sub>).

The PICC shall not emit subcarrier following any frame with invalid CRC<sub>B</sub> or with another CID than the one assigned.

##### State exit conditions and transitions:

The PICC enters the HALT State when a valid DESELECT Command frame is received (DESELECT Command is defined in ISO/IEC 14443-4).

##### Specific remarks:

Valid REQB/WUPB or Slot-MARKER frames shall not be answered.

A valid frame with an ATTRIB Command shall not be answered.

In the higher layer protocol, specific commands may be defined to return the PICC to other states (IDLE or HALT). The PICC may return to these states only following reception of such commands.

#### 7.4.8 HALT State

##### Description:

The PICC shall only react to a WUPB command.

If the AFI matches, then the PICC moves to the READY REQUESTED or READY DECLARED sub-state.

If the AFI does not match, then the PICC moves to IDLE state.

The PUPI shall not change (see 7.9.2) when entering or leaving the HALT state.

### State exit conditions and transitions:

The PICC returns to the POWER\_OFF State if the RF field disappears.

## 7.5 Command set

Four primitive commands are used to manage multi-node communication channels:

- REQB/WUPB;
- Slot-MARKER;
- ATTRIB;
- HLTB.

All four commands use the character, frame format and timing detailed in clause 7.1.

The commands and the responses of the PICC to these commands are described in the following sections.

Any frame received with a wrong format (wrong frame identifiers or invalid CRC\_B) shall be ignored.

## 7.6 Anticollision response rules

A PICC which is in the READY-REQUESTED sub-state, after receiving a valid REQB/WUPB Command (Requested AFI=0 or AFI matched with an internal application), shall respond according to the following rules, where the parameter N has been given in the REQB/WUPB Command:

If  $N = 1$  the PICC shall send an ATQB and shall move to the READY-DECLARED sub-state

If  $N > 1$  the PICC shall internally generate a random number R which shall be evenly distributed between 1 to N

If  $R = 1$  the PICC shall send an ATQB and shall move to the READY-DECLARED sub-state.

If  $R > 1$  the PICC shall wait until it has received a Slot-MARKER Command with a matched slot number (slot number = R) before sending the ATQB and moving to the READY-DECLARED sub-state

If anticollision resolution is not required (e.g. only one PICC is expected in the PCD field), there is no need for a PICC to support either the REQB/WUPB command with  $N > 1$  or the Slot-Marker command. Such PICCs may not operate with all PCDs especially with those not using REQB/WUPB with  $N = 1$ , or in multiple PICC situations.

## 7.7 REQB/WUPB Command

The REQB and WUPB Commands sent by the PCD are used to probe the field for PICCs of Type B.

In addition WUPB is particularly used to also wake up PICCs which are in HALT State.

The number of slots N is included in the command as a parameter to optimize the anti-collision algorithm for a given application. See Figure 19 for detailed description of when the PICC shall respond to these respective commands.

### 7.7.1 REQB/WUPB Command format

REQB/WUPB Command has the following format:

|                        |     |                        |     |                           |     |   |     |
|------------------------|-----|------------------------|-----|---------------------------|-----|---|-----|
| 1 <sup>st</sup> byte   |     | 2 <sup>nd</sup> byte   |     | 3 <sup>rd</sup> byte      |     | 4 <sup>th</sup> , 5 <sup>th</sup> bytes |     |
| <b>APf</b><br>(1 byte) |     | <b>AFI</b><br>(1 byte) |     | <b>PARAM</b><br>(1 byte). |     | <b>CRC_B</b><br>(2 bytes)               |     |
| MSB                    | LSB | MSB                    | LSB | MSB                       | LSB | MSB                                     | LSB |

Figure 22 — REQB/WUPB Command format

### 7.7.2 Coding of Anticollision Prefix byte APf

The Anticollision Prefix byte is APf = '05' = (0000 0101)b.

### 7.7.3 Coding of AFI

AFI (Application Family Identifier) represents the type of application targeted by the PCD and is used to preselect PICCs before the ATQB. Only PICCs with applications of the type indicated by the AFI may answer to a REQB/WUPB Command with AFI different to '00'.

When AFI equals '00', all PICCs shall process the REQB/WUPB.

The most significant half byte of AFI is used to code one specific or all application families, as defined in Table 12.

The least significant half byte of AFI is used to code one specific or all application sub-families. Sub-family codes different from 0 are proprietary unless defined in table 14.

Table 22 — AFI coding

| AFI Most Significant half byte | AFI Least Significant half byte                    | Meaning – PICCs respond from              | Examples / Note                |
|--------------------------------|--|---|--------------------------------|
| '0'                            | '0'  | All families and sub-families             | No application preselection    |
| X                              | '0'  | All sub-families of family X              | Wide application preselection  |
| X                              | Y  | Only the Yth sub-family of family X       |                                |
| '0'                            | Y  | Proprietary sub-family Y only             |                                |
| '1'                            | '0', Y   | Transport                                 | Mass transit, Bus, Airline,... |
| '2'                            | '0', Y   | Financial                                 | IEP, Banking, Retail,...       |
| '3'                            | '0', Y   | Identification                            | Access Control,...             |
| '4'                            | '0', Y   | Telecommunication                         | Public Telephony, GSM,...      |
| '5'                            | '0', Y   | Medical                                   |                                |
| '6'                            | '0', Y   | Multimedia                                | Internet services,...          |
| '7'                            | '0', Y   | Gaming                                    |                                |
| '8'                            | '0', Y   | Data Storage                              | Portable Files, ...            |
| '9'-D'                         | '0', Y   | RFU                                       |                                |
| 'E'                            | '0',<br>Y = 1,<br>Y = 2,<br>Other Y values are RFU | Machine Readable Travel Documents (MRTDs) | Y=1 Passport<br>Y=2 Visa       |
| 'F'                            | '0', Y   | RFU                                       |                                |

NOTE X = '1' to 'F', Y = '1' to 'F'

A PCD sending a REQB/WUPB command with an AFI field set to an RFU value is not compliant with this standard.

The PICC shall not respond when the AFI field is set to a value which is RFU.

**7.7.4 Coding of PARAM**

|     |    |    |                         |             |                     |    |    |
|-----|----|----|-------------------------|-------------|---------------------|----|----|
| b8  | b7 | b6 | b5                      | b4          | b3                  | b2 | b1 |
| RFU |    |    | Extended ATQB supported | REQB / WUPB | N (Number of slots) |    |    |

All RFU bits shall be set to 0.

**Figure 23 — Coding of PARAM**

b4 = 0 defines REQB: PICCs in IDLE state or READY state shall process this command.  
 b4 = 1 defines WUPB: PICCs in IDLE state or READY state or HALT state shall process this command.  
 b1, b2 and b3 are used to code the number of slots N according to Table 13.  
 b5 indicates the PCD capability to support extended ATQB response from the PICC. The use of extended ATQB is optional for the PICC. The coding of b5 is as follows:

b5 = 0 defines: extended ATQB defined in subclause 7.9.4.7 is not supported by the PCD.

b5 = 1 defines: extended ATQB defined in subclause 7.9.4.7 is supported by the PCD.

A PCD sending a REQB/WUPB command with (b8 to b6) <> (000)b is not compliant with this standard.

The PICC should ignore (b8 to b6) and its interpretation of any other field of the whole frame shall not change.

**Table 23 — Coding of N**

| b3 | b2 | b1 | N                   |
|----|----|----|---------------------|
| 0  | 0  | 0  | 1 = 2 <sup>0</sup>  |
| 0  | 0  | 1  | 2 = 2 <sup>1</sup>  |
| 0  | 1  | 0  | 4 = 2 <sup>2</sup>  |
| 0  | 1  | 1  | 8 = 2 <sup>3</sup>  |
| 1  | 0  | 0  | 16 = 2 <sup>4</sup> |
| 1  | 0  | 1  | RFU                 |
| 1  | 1  | x  | RFU                 |

A PCD sending (b3 to b1) = (101)b or (11x)b is not compliant with this standard.

Until the RFU values (101)b or (11x)b are assigned by ISO, a PICC receiving (b3 to b1) = (101)b or (11x)b should interpret it as (b3 to b1) = (100)b (16 slots).

NOTE For each PICC, the probability of response (ATQB) in the first slot is 1/N.

**7.8 Slot-MARKER Command**

After a REQB/WUPB Command, the PCD may send up to (N-1) Slot-MARKER Commands to define the start of each timeslot.

Slot-MARKER Commands can be sent:

- after the end of an ATQB message received by the PCD to mark the start of the next slot;
- or earlier if no ATQB is received (no need to wait until the end of a slot, if this slot is known to be empty).

**7.8.1 Slot-MARKER Command format**

Slot-MARKER Command has the following format:

|                        |     |   |     |
|------------------------|-----|---|-----|
| 1 <sup>st</sup> byte   |     | 2 <sup>nd</sup> , 3 <sup>rd</sup> bytes |     |
| <b>APn</b><br>(1 byte) |     | <b>CRC_B</b><br>(2 bytes)               |     |
| MSB                    | LSB | MSB                                     | LSB |

**Figure 24 — Slot-MARKER Command format**

**7.8.2 Coding of Anticollision Prefix byte APn**

APn = (nnnn 0101)b where nnnn codes the slot number as defined in the Table 14.

**Table 24 — Coding of slot number**

| nnnn  | Slot number |
|-------|-------------|
| 0001  | 2           |
| 0010  | 3           |
| 0011  | 4           |
| ..... | .....       |
| 1110  | 15          |
| 1111  | 16          |

NOTE It is not mandatory that the Slot-MARKER Commands are sent sequentially with incremental slot numbers.

**7.9 ATQB Response**

The response to both REQW/WUPB and Slot-MARKER Commands is named ATQB.

**7.9.1 ATQB Response format**

ATQB Response has one of the following two formats:

|                         |   |   |  |   |
|-------------------------|---|---|--|---|
| 1 <sup>st</sup> byte    | 2 <sup>nd</sup> , 3 <sup>rd</sup> , 4 <sup>th</sup> , 5 <sup>th</sup> bytes | 6 <sup>th</sup> , 7 <sup>th</sup> , 8 <sup>th</sup> , 9 <sup>th</sup> , bytes | 10 <sup>th</sup> , 11 <sup>th</sup> , 12 <sup>th</sup> , bytes | 13 <sup>th</sup> , 14 <sup>th</sup> bytes |
| <b>'50'</b><br>(1 byte) | <b>PUPI</b><br>(4 bytes)  | <b>Application Data</b><br>(4 bytes)  | <b>Protocol Info</b><br>(3 bytes)                              | <b>CRC_B</b><br>(2 bytes)                 |
| MSB LSB                 | MSB LSB   | MSB LSB   | MSB LSB  | MSB LSB                                   |

|                         |   |   |   |   |
|-------------------------|---|---|---|---|
| 1 <sup>st</sup> byte    | 2 <sup>nd</sup> , 3 <sup>rd</sup> , 4 <sup>th</sup> , 5 <sup>th</sup> bytes | 6 <sup>th</sup> , 7 <sup>th</sup> , 8 <sup>th</sup> , 9 <sup>th</sup> , bytes | 10 <sup>th</sup> , 11 <sup>th</sup> , 12 <sup>th</sup> , 13 <sup>th</sup> bytes | 14 <sup>th</sup> , 15 <sup>th</sup> bytes |
| <b>'50'</b><br>(1 byte) | <b>PUPI</b><br>(4 bytes)  | <b>Application Data</b><br>(4 bytes)  | <b>Protocol Info</b><br>(4 bytes)   | <b>CRC_B</b><br>(2 bytes)                 |
| MSB LSB                 | MSB LSB   | MSB LSB   | MSB LSB   | MSB LSB                                   |

**Figure 25 — ATQB Response formats**

**7.9.2 PUPI (Pseudo-Unique PICC Identifier)**

A Pseudo-Unique PICC Identifier (PUPI) is used to differentiate PICCs during anticollision. This 4-byte number may be either a number dynamically generated by the PICC or a diversified fixed number. The PUPI shall only be generated by a state transition from the “Power Off” to the “IDLE” state.

WARNING - PICCs based on ISO/IEC 14443:2001 may change their PUPI when leaving HALT state

**7.9.3 Application Data**

The Application data field is used to inform the PCD which applications are currently installed in the PICC. This information allows the PCD to select the desired PICC in the presence of more than one PICC.

The application data is defined dependent upon the ADC (Application Data Coding) field in the Protocol Info (see 7.9.4.), which defines if either the CRC\_B compressing method described below or proprietary coding is used.

When the CRC\_B compressing coding is used, Application Data field contains the following:

|                      |     |   |     |                                     |     |
|----------------------|-----|---|-----|-------------------------------------|-----|
| 1 <sup>st</sup> byte |     | 2 <sup>nd</sup> , 3 <sup>rd</sup> bytes |     | 4 <sup>th</sup> byte                |     |
| AFI<br>(1 byte)      |     | CRC_B(AID)<br>(2 bytes)                 |     | Numbers of Applications<br>(1 byte) |     |
| MSB                  | LSB | MSB                                     | LSB | MSB                                 | LSB |

**Figure 26 — Application Data format**

**7.9.3.1 AFI**

For mono application PICCs AFI gives the family of the application (see AFI coding in Table 12).

For multi application PICCs AFI gives the family of the application described in CRC\_B(AID).

**7.9.3.2 CRC\_B(AID)**

CRC\_B(AID) is the result of calculation of CRC\_B of the AID (as defined in ISO/IEC 7816-5) of an application in the PICC matching the AFI given in the REQB/WUPB Command.

**7.9.3.3 Numbers of Applications**

Indicates on presence of other applications in the PICC.

The most significant half byte value gives the number of applications corresponding to the AFI given in Application Data with '0' meaning no application and 'F' meaning 15 applications or more.

The least significant half byte value gives the total number of applications in the PICC with '0' meaning no application and 'F' meaning 15 applications or more.

**7.9.4 Protocol Info**

The Protocol Info field indicates the parameters supported by the PICC. It is formatted as detailed below.

|                                 |                            |                           |                      |                 |                |  |                 |
|---------------------------------|----------------------------|---------------------------|----------------------|-----------------|----------------|--|-----------------|
| 1 <sup>st</sup> byte            | 2 <sup>nd</sup> byte       |                           | 3 <sup>rd</sup> byte |                 |                | 4 <sup>th</sup> byte (optional)<br>Extended ATQB |                 |
| Bit_Rate_capability<br>(8 bits) | Max_Frame_Size<br>(4 bits) | Protocol_Type<br>(4 bits) | FWI<br>(4 bits)      | ADC<br>(2 bits) | FO<br>(2 bits) | SFGI<br>(4 bits)                                 | RFU<br>(4 bits) |

MSB      LSB MSB                                  LSB MSB                                  LSB MSB                                  LSB

All RFU bits shall be set to 0.

**Figure 27 — Protocol info format**

**7.9.4.1 FO**

**Table 25 — Frame Option supported by the PICC**

| b2 | b1 | Meaning                   |
|----|----|---------------------------|
| 1  | x  | NAD supported by the PICC |
| x  | 1  | CID supported by the PICC |

**7.9.4.2 ADC**

**Table 26 — Application Data Coding supported by the PICC**

| b4                   | b3 | Meaning  |
|----------------------|----|--|
| 0                    | 0  | Application is proprietary                         |
| 0                    | 1  | Application is coded as described in clause 7.9.3. |
| Other values are RFU |    |  |

A PICC setting (b4,b3) any value other than (00)b or (01)b is not compliant with this standard.

**7.9.4.3 FWI**

Frame Waiting time Integer (4 bits):

FWI codes an integer value used to define the FWT.

The FWT defines the maximum time for a PICC to start its response after the end of a PCD frame.

FWT is calculated by the formula:

$$FWT = (256 \times 16/fc) \times 2^{FWI}$$

where the value of FWI has the range from 0 to 14 and the value of 15 is RFU.

For FWI = 0, FWT is minimal (~ 302 μs);

For FWI = 14, FWT is maximal (~ 4949 ms).

In case of extended ATQB supported by the PICC and the PCD:

- FWT applies after the Answer to ATTRIB command,
- the waiting time for the Answer to ATTRIB command is a fixed value given by the following formula:

$$\text{Answer to ATTRIB waiting time} = (256 \times 16/ fc) \times 2^4 (\sim 4,8 \text{ ms})$$

Note: It is strongly recommended to use a FWT as low as possible to protect communications speed when retry occurs.

A PICC setting FWI = 15 is not compliant with this standard.

Until the RFU value 15 is assigned by ISO, a PCD receiving FWI = 15 should interpret it as FWI = 4.

NOTE This clause is added for PCD's compatibility with future PICCs when ISO defines the behaviour for an RFU value of 15.

7.9.4.4 Protocol\_Type

Table 27 — Protocol Types supported by the PICC

| b1 | Meaning                                 |
|----|---|
| 1  | PICC compliant with ISO/IEC 14443-4     |
| 0  | PICC not compliant with ISO/IEC 14443-4 |

The minimum value of TR2 (delay between PICC EOF start and PCD SOF start) is defined by Protocol\_Type bits (b3,b2) as specified in table 20.

The PCD should not continue communicating with a PICC that sets b4 to (1)b.

Table 28 — Minimum TR2 coding

| b3 | b2 | Minimum TR2 for PICC to PCD |
|----|----|-----------------------------|
| 0  | 0  | 10 etu + 32/fs              |
| 0  | 1  | 10 etu + 128/fs             |
| 1  | 0  | 10 etu + 256/fs             |
| 1  | 1  | 10 etu + 512/fs             |

The bit b4 is RFU and shall be set to 0.

7.9.4.5 Max\_Frame\_Size

Table 29 — Maximum frame size

| Maximum Frame Size Code in ATQB | 0  | 1  | 2  | 3  | 4  | 5  | 6  | 7   | 8   | 9-F       |
|---------------------------------|----|----|----|----|----|----|----|-----|-----|-----------|
| Maximum Frame Size (bytes)      | 16 | 24 | 32 | 40 | 48 | 64 | 96 | 128 | 256 | RFU > 256 |

A PICC setting Maximum Frame Size Code = '9'-F' is not compliant with this standard.

Until the RFU values '9'-F' are assigned by ISO, a PCD receiving Maximum\_Frame\_Size Code = '9'-F' should interpret it as Maximum Frame Size Code = '8' (256 bytes).

NOTE This clause is added for PCD's compatibility with future PICCs when ISO defines the behaviour for a RFU values of '9'-F'.

## 7.9.4.6 Bit\_Rate\_capability

Table 30 — Bit rates supported by the PICC

| b8                                  | b7 | b6 | b5 | b4 | b3 | b2 | b1 | Meaning  |
|-------------------------------------|----|----|----|----|----|----|----|--|
| 0                                   | 0  | 0  | 0  | 0  | 0  | 0  | 0  | PICC supports only 106 kbit/s in both directions               |
| 1                                   | x  | x  | x  | 0  | x  | x  | x  | Same bit rate from PCD to PICC and from PICC to PCD compulsory |
| x                                   | x  | x  | 1  | 0  | x  | x  | x  | PICC to PCD, 1etu = 64 / fc, bit rate supported is 212 kbit/s  |
| x                                   | x  | 1  | x  | 0  | x  | x  | x  | PICC to PCD, 1etu = 32 / fc, bit rate supported is 424 kbit/s  |
| x                                   | 1  | x  | x  | 0  | x  | x  | x  | PICC to PCD, 1etu = 16 / fc, bit rate supported is 847 kbit/s  |
| x                                   | x  | x  | x  | 0  | x  | x  | 1  | PCD to PICC, 1etu = 64 / fc, bit rate supported is 212 kbit/s  |
| x                                   | x  | x  | x  | 0  | x  | 1  | x  | PCD to PICC, 1etu = 32 / fc, bit rate supported is 424 kbit/s  |
| x                                   | x  | x  | x  | 0  | 1  | x  | x  | PCD to PICC, 1etu = 16 / fc, bit rate supported is 847 kbit/s  |
| Other values (with b4 = 1) are RFU. |    |    |    |    |    |    |    |  |

A PICC setting b4 = 1 is not compliant with this standard.

Until the RFU values with b4 = 1 are assigned by ISO, a PCD receiving Bit\_Rate\_capability with b4 = 1 should interpret the Bit\_Rate\_capability byte as if b8 to b1 = 0 (only ~106 kbit/s in both directions).

## 7.9.4.7 Extended ATQB (optional)

"The optional Extended ATQB byte (optional 4<sup>th</sup> byte of protocol info field) consists of two parts:

- the least significant half byte (b4 to b1) is RFU and shall be set to (0000)b;
- the most significant half byte (b8 to b5) defines the Start-up Frame Guard time Integer (SFGI).

The SFGI codes an integer value used to define the Start-up Frame Guard Time (SFGT).

The SFGT defines a specific guard time needed by the PICC before it is ready to receive the next frame after it has sent the Answer to ATTRIB command. SFGI is coded in the range from 0 to 14. The value of 15 is RFU. The values in the range from 0 to 14 are used to calculate the SFGT with the formula given below. The default value of SFGI is 0.

$$\text{SFGT} = (256 \times 16/\text{fc}) \times 2^{\text{SFGI}}$$

For SFGI = 0, SFGT is minimal (~ 302 μs);

For SFGI = 14, SFGT is maximal (~ 4949 ms).

A PICC setting SFGI = 15 is not compliant with this standard.

Until the RFU value 15 is assigned by ISO, a PCD receiving SFGI = 15 shall interpret it as SFGI = 0.

A PICC sending a ATQB response with (b4 to b1) <> (0000)b is not compliant with this standard.

The PCD shall ignore (b4 to b1) and its interpretation of any other field of the whole frame shall not change. A PCD changing its interpretation of any other field of the whole frame when (b4 to b1) <> (0000)b is not compliant with this standard.

When answering a REQB/WUPB command with bit b5 set to 0 (no extended ATQB supported) the PICC shall not send the optional 4<sup>th</sup> byte in its ATQB response.

**7.10 ATTRIB Command**

The ATTRIB Command sent by the PCD shall include information required to select a single PICC.

A PICC receiving an ATTRIB Command with its identifier becomes selected and assigned to a dedicated channel. After being selected, this PICC only responds to commands defined in ISO/IEC 14443-4 which include its unique CID.

The parameters selected in the ATTRIB command shall apply after the Answer to ATTRIB.

**7.10.1 ATTRIB Command format**

ATTRIB Command has the following format:

|                      |   |                      |                      |                      |                      |  |                    |
|----------------------|---|----------------------|----------------------|----------------------|----------------------|--|--------------------|
| 1 <sup>st</sup> byte | 2 <sup>nd</sup> , 3 <sup>rd</sup> , 4 <sup>th</sup> , 5 <sup>th</sup> bytes | 6 <sup>th</sup> byte | 7 <sup>th</sup> byte | 8 <sup>th</sup> byte | 9 <sup>th</sup> byte | 10 <sup>th</sup> , .....bytes                      |                    |
| '1D'<br>(1 byte)     | Identifier<br>(4 bytes)   | Param 1<br>(1 byte)  | Param 2<br>(1 byte)  | Param 3<br>(1 byte)  | Param 4<br>(1 byte)  | Higher layer - INF<br>(optional – 0 or more bytes) | CRC_B<br>(2 bytes) |
| MSB LSB              | MSB      LSB  | MSB LSB              | MSB LSB              | MSB LSB              | MSB LSB              | MSB                      LSB                       | MSB LSB            |

**Figure 28 — ATTRIB Command format**

**7.10.2 Identifier**

This identifier is the value of the PUPI sent by the PICC in the ATQB.

**7.10.3 Coding of Param 1**

|             |    |             |    |     |     |     |    |
|-------------|----|-------------|----|-----|-----|-----|----|
| b8          | b7 | b6          | b5 | b4  | b3  | b2  | b1 |
| Minimum TR0 |    | Minimum TR1 |    | EOF | SOF | RFU |    |

All RFU bits shall be set to 0 if not otherwise specified

**Figure 29 — Coding of Param 1**

A PCD setting (b2,b1) <> (00)b is not compliant with this standard.

The PICC should ignore any value (b2,b1) and its interpretation of any other field of the whole frame shall not change.

**7.10.3.1 Minimum TR0**

Minimum TR0 indicates to the PICC the minimum delay before responding after the end of a command sent by a PCD. The default value has been defined in ISO/IEC 14443-2 clause 9.2.5.

**Table 31 — Minimum TR0 coding**

| b8 | b7 | Minimum TR0 for PCD to PICC bit rate of |               |               |               |
|----|----|---|---------------|---------------|---------------|
|    |    | <i>fc</i> /128                          | <i>fc</i> /64 | <i>fc</i> /32 | <i>fc</i> /16 |
| 0  | 0  | 64/ <i>fs</i>                           | 64/ <i>fs</i> | 64/ <i>fs</i> | 64/ <i>fs</i> |
| 0  | 1  | 48/ <i>fs</i>                           | 32/ <i>fs</i> | 16/ <i>fs</i> | 16/ <i>fs</i> |
| 1  | 0  | 16/ <i>fs</i>                           | 8/ <i>fs</i>  | 4/ <i>fs</i>  | 4/ <i>fs</i>  |
| 1  | 1  | RFU                                     | RFU           | RFU           | RFU           |

NOTE Minimum TR0 is required by the PCD when switching from transmit to receive and its value depends on the PCD performance.

A PCD setting (b8,b7) = (11)b is not compliant with this standard.

Until the RFU value (11)b is assigned by ISO, a PICC receiving (b8,b7) = (11)b should interpret it as (b8,b7) = (00)b, the default value.

### 7.10.3.2 Minimum TR1

Minimum TR1 indicates to the PICC the minimum delay between subcarrier modulation start and beginning of data transmission. The default value has been defined in ISO/IEC 14443-2 clause 9.2.5.

**Table 32 — Minimum TR1 coding**

| b6 | b5 | Minimum TR1 for PCD to PICC bit rate of |               |               |               |
|----|----|---|---------------|---------------|---------------|
|    |    | <i>fc</i> /128                          | <i>fc</i> /64 | <i>fc</i> /32 | <i>fc</i> /16 |
| 0  | 0  | 80/fs                                   | 80/fs         | 80/fs         | 80/fs         |
| 0  | 1  | 64/fs                                   | 32/fs         | 32/fs         | 32/fs         |
| 1  | 0  | 16/fs                                   | 8/fs          | 8/fs          | 8/fs          |
| 1  | 1  | RFU                                     | RFU           | RFU           | RFU           |

NOTE Minimum TR1 is required by the PCD for synchronization with the PICC and its value depends on the PCD performance.

A PCD setting (b6,b5) = (11)b is not compliant with this standard.

Until the RFU value (11)b is assigned by ISO, a PICC receiving (b6,b5) = (11)b should interpret it as (b6,b5) = (00)b, the default value.

### 7.10.3.3 EOF/SOF

b3 and b4 indicate the PCD capability to support suppression of the EOF and/or SOF from PICC to PCD, which may reduce communication overhead. The suppression of EOF and/or SOF is optional for the PICC. The coding of b3 and b4 is as follows:

**Table 33 — SOF handling**

| b3 | SOF required |
|----|--------------|
| 0  | Yes          |
| 1  | No           |

**Table 34 — EOF handling**

| b4 | EOF required |
|----|--------------|
| 0  | Yes          |
| 1  | No           |

SOF/EOF suppression applies only for communications at *fc*/128 (~106kbit/s). For bit rates higher than *fc*/128 (~106kbit/s) the PICC shall always provide SOF and EOF.

**7.10.4 Coding of Param 2**

The least significant half byte (b4 to b1) is used to code the maximum frame size that can be received by the PCD as specified in Table 24.

**Table 35 — Coding of b4 to b1 of Param 2**

|                                   |    |    |    |    |    |    |    |     |     |           |
|-----------------------------------|----|----|----|----|----|----|----|-----|-----|-----------|
| Maximum Frame Size Code in ATTRIB | 0  | 1  | 2  | 3  | 4  | 5  | 6  | 7   | 8   | 9-F       |
| Maximum Frame Size (bytes)        | 16 | 24 | 32 | 40 | 48 | 64 | 96 | 128 | 256 | RFU > 256 |

The most significant half byte (b8 to b5) is used for bit rate selection, as specified in Tables 25 and 26.

**Table 36 — Coding of b6 & b5 of Param 2**

| b6 | b5 | Meaning  |
|----|----|--|
| 0  | 0  | PCD to PICC, 1etu = 128 / fc, bit rate is 106 kbit/s |
| 0  | 1  | PCD to PICC, 1etu = 64 / fc, bit rate is 212 kbit/s  |
| 1  | 0  | PCD to PICC, 1etu = 32 / fc, bit rate is 424 kbit/s  |
| 1  | 1  | PCD to PICC, 1etu = 16 / fc, bit rate is 847 kbit/s  |

**Table 37 — Coding of b8 & b7 of Param 2**

| b8 | b7 | Meaning  |
|----|----|--|
| 0  | 0  | PICC to PCD, 1etu = 128 / fc, bit rate is 106 kbit/s |
| 0  | 1  | PICC to PCD, 1etu = 64 / fc, bit rate is 212 kbit/s  |
| 1  | 0  | PICC to PCD, 1etu = 32 / fc, bit rate is 424 kbit/s  |
| 1  | 1  | PICC to PCD, 1etu = 16 / fc, bit rate is 847 kbit/s  |

A PCD setting Maximum Frame Size Code = '9'-F' is not compliant with this standard.

Until the RFU values '9'-F' are assigned by ISO, a PICC receiving Maximum Frame Size Code = '9'-F' should interpret it as Maximum\_Frame\_Size Code = '8' (256 bytes).

**7.10.5 Coding of Param 3**

The least significant half byte (b4 to b1) is used for confirmation of the protocol type as specified in Table 17.

The most significant half byte (b8 to b5) is set to (0000)b, all other values are RFU.

A PCD setting (b8 to b5) <> (0000)b is not compliant with this standard.

The PICC should ignore and not answer the ATTRIB command when (b8 to b5) <> (0000)b.

**7.10.6 Coding of Param 4**

The Param 4 byte consists of two parts:

- the least significant half byte (b4 to b1) is named Card Identifier (CID) and defines the logical number of the addressed PICC in the range from 0 to 14. The value 15 is RFU. The CID is specified by the PCD and shall be unique for each active PICC. If the PICC does not support CID, code value (0000)b shall be used;

— the most significant half byte (b8 to b5) is set to (0000)b, all other values are RFU.

A PCD setting CID = 15 is not compliant with this standard.

The PICC should ignore and not answer the ATTRIB command when received value of CID = 15 as any action in the PICC for CID = 15 may be decided in the future by ISO.

A PCD setting (b8 to b5) <> (0000)b is not compliant with this standard.

The PICC should ignore (b8 to b5) and its interpretation of any other field of the whole frame shall not change.

### 7.10.7 Higher layer INF

Any higher layer command transferable as the INF field of ISO/IEC 14443-4 may be included, but it is not mandatory for the PICC to process it.

The PICC processing of the ATTRIB command shall not be altered by the inclusion of the high layer command.

## 7.11 Answer to ATTRIB Command

The PICC shall answer to any valid ATTRIB Command (correct PUPI and valid CRC\_B) with the format described below.

| 1 <sup>st</sup> byte |            | 2 <sup>nd</sup> .....bytes   |     |              |     |
|----------------------|------------|------------------------------|-----|--------------|-----|
| <b>MBLI</b>          | <b>CID</b> | <b>Higher layer Response</b> |     | <b>CRC_B</b> |     |
| (1 byte)             |            | (optional 0 or more bytes)   |     | (2 bytes)    |     |
| MSB                  | LSB        | MSB                          | LSB | MSB          | LSB |

**Figure 30 — Format of the Answer to an ATTRIB Command**

The first byte consists of two parts:

- the least significant half byte (b4 to b1) contains the returned CID. If the PICC does not support CID, code value (0000)b is returned;
- the most significant half byte (b8 to b5) is called the Maximum Buffer Length Index (MBLI). It is used by the PICC to let the PCD know the limit of its internal buffer to received chained frames. The coding of MBLI is as follows:
  - MBLI = 0 means that the PICC provides no information on its internal input buffer size;
  - MBLI > 0 is used to calculate the actual internal maximum buffer length (MBL) according to the following formula:  $MBL = (\text{PICC Maximum Frame Size}) * 2^{(MBLI-1)}$  where the PICC maximum frame size is returned by the PICC in its ATQB. When it send chained frames to a PICC, the PCD shall ensure that the accumulated length is never greater than MBL.

Remaining bytes are optional and used for higher layer response.

As illustrated below, a PICC shall answer the empty (no higher layer INF field) ATTRIB Command with an empty higher layer response:

|                      |            |   |     |
|----------------------|------------|---|-----|
| 1 <sup>st</sup> byte |            | 2 <sup>nd</sup> , 3 <sup>rd</sup> bytes |     |
| <b>MBLI</b>          | <b>CID</b> | <b>CRC_B</b>                            |     |
| (1 byte)             |            | (2 bytes)                               |     |
| MSB                  | LSB        | MSB                                     | LSB |

**Figure 31 — PICC Answer to ATTRIB format without higher layer response**

NOTE 1 A valid Answer (same CID and valid CRC\_B) to an ATTRIB Command (as defined in Figure 28 or 29) is the means for a PCD to verify that PICC selection has been successful.

NOTE 2 Higher layer response indicating that the higher layer command is not supported by the PICC within this context is allowed as long as the PICC response meets the format described above.

**7.12 HLTB Command and Answer**

The HLTB Command is used to set a PICC in HALT State and stop responding to a REQb.

After answering to this command the PICC shall ignore any commands except the WUPB Command (see 7.7)

HLTB Command has the following format:

|                         |   |   |
|-------------------------|---|---|
| 1 <sup>st</sup> byte    | 2 <sup>nd</sup> , 3 <sup>rd</sup> , 4 <sup>th</sup> , 5 <sup>th</sup> bytes | 6 <sup>th</sup> , 7 <sup>th</sup> bytes |
| <b>'50'</b><br>(1 byte) | <b>Identifier</b><br>(4 bytes)  | <b>CRC_B</b><br>(2 bytes)               |
| MSB LSB                 | MSB   | LSB                                     |

**Figure 32 — Format of the HLTB Command**

The 4 bytes identifier is the value of the PUPI sent by the PICC in the ATQB.

The format of Answer to a HLTB Command from the PICC is as follows:

|                         |   |
|-------------------------|---|
| 1 <sup>st</sup> byte    | 2 <sup>nd</sup> , 3 <sup>rd</sup> bytes |
| <b>'00'</b><br>(1 byte) | <b>CRC_B</b><br>(2 bytes)               |
| MSB LSB                 | MSB                                     |

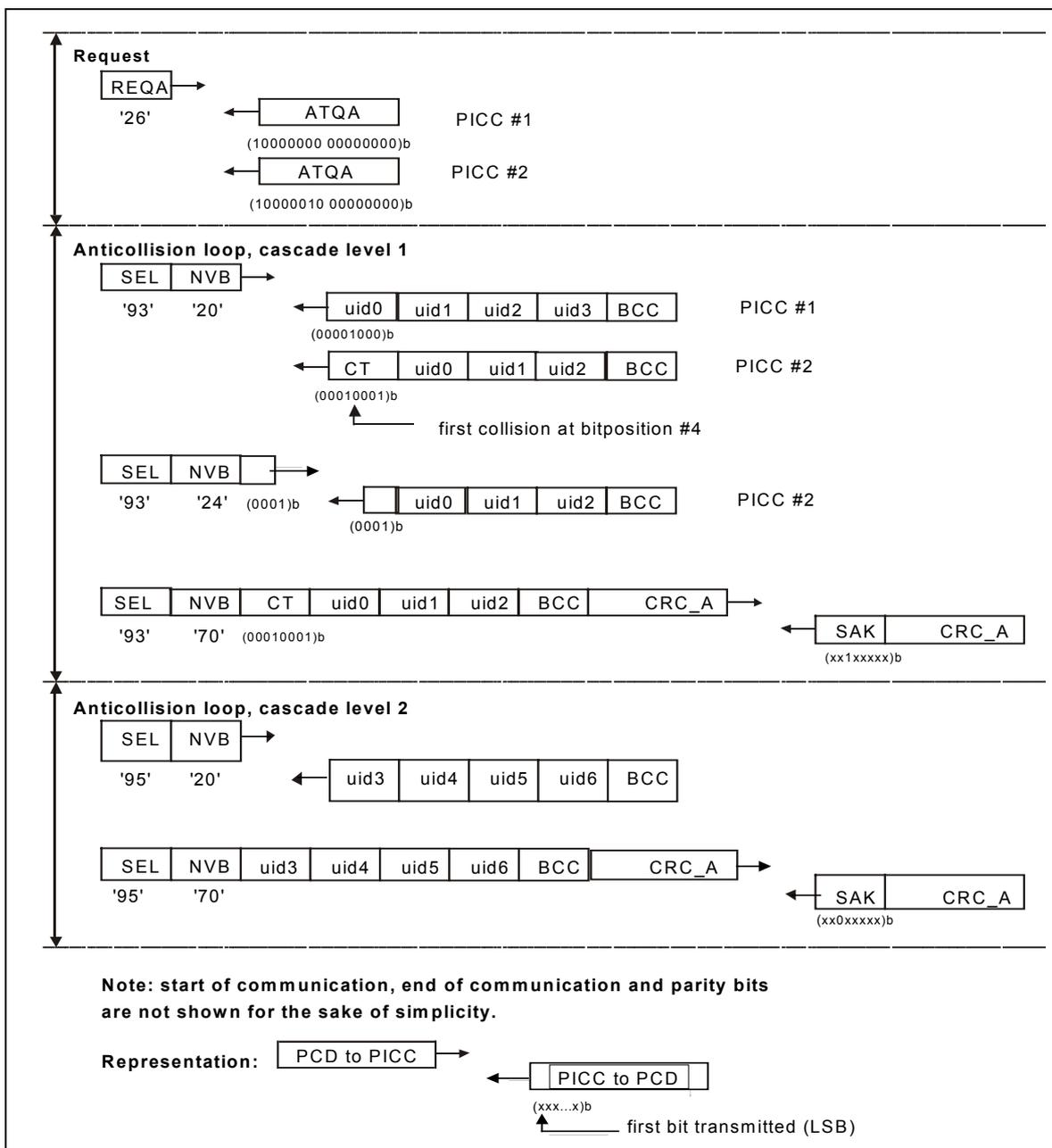
**Figure 33 — Format of PICC Answer to HLTB Command**

## Annex A (informative)

### Communication example Type A

This example shows the select sequence with 2 PICCs in the field on the assumption of:

- PICC #1 with UID size: single, value of uid0 is '10';
- PICC #2 with UID size: double.



**Figure A.1 —Select sequence with bit frame anticollision**

**Explanations to Figure A1:**

|                                     |   |
|-------------------------------------|---|
| Request                             | <ul style="list-style-type: none"> <li>— PCD transmits the REQA Command</li> <li>— All PICCs respond with their ATQA:             <ul style="list-style-type: none"> <li>— PICC #1 indicates bit frame anticollision and UID size: single;</li> <li>— PICC #2 indicates bit frame anticollision and UID size: double.</li> </ul> </li> </ul>  |
| Anticollision loop, cascade level 1 | <ul style="list-style-type: none"> <li>— PCD transmits an ANTICOLLISION Command:             <ul style="list-style-type: none"> <li>— SEL specifies bit frame anticollision and cascade level 1;</li> <li>— the value '20' of NVB specifies that the PCD will transmit no part of UID CL1;</li> </ul> </li> <li>— consequently all PICCs in the field respond with their complete UID CL1;</li> <li>— due to the value '88' of the cascade tag, the first collision occurs at bit position #4;</li> <li>— PCD transmits another ANTICOLLISION Command that includes the first 3 bits of UID CL1 that were received before the collision occurs, followed by a (1)b. Consequently the PCD assigns NVB with the value '24';</li> <li>— these 4 bits correspond to the first bits of UID CL1 of PICC #2;</li> <li>— PICC #2 responds with its 36 remaining bits of UID CL1. Since PICC #1 does not respond, no collision occurs;</li> <li>— since the PCD "knows" all bits of UID CL1 of PICC #2, it transmits a SELECT Command for PICC #2;</li> <li>— PICC #2 responds with SAK, indicating that UID is not complete;</li> <li>— consequently, the PCD increases the cascade level.</li> </ul> |
| Anticollision loop, cascade level 2 | <ul style="list-style-type: none"> <li>— PCD transmits another ANTICOLLISION Command:             <ul style="list-style-type: none"> <li>— SEL specifies bit frame anticollision and cascade level 2;</li> <li>— NVB is reset to '20' to force PICC #2 to respond with its complete UID CL2;</li> </ul> </li> <li>— PICC #2 responds with all 40 bits of its UID CL2;</li> <li>— PCD transmits the SELECT Command for PICC #2, cascade level 2;</li> <li>— PICC #2 responds with SAK, indicating that UID is complete, and transits from READY State to ACTIVE State.</li> </ul>  |

## Annex B (informative)

### CRC\_A and CRC\_B encoding

#### CRC\_A encoding

This annex is provided for explanatory purposes and indicates the bit patterns that will exist in the physical layer. It is included for the purpose of checking an ISO/IEC 14443-3 Type A implementation of CRC\_A encoding.

The process of encoding and decoding may be conveniently carried out by a 16-stage cyclic shift register with appropriate feedback gates. According to ITU-T Recommendation V.41, ANNEX I, figures I-1/V.41 and I-2/V.41 the flip-flops of the register shall be numbered from FF0 to FF15. FF0 shall be the leftmost flip-flop where data is shifted in. FF15 shall be the rightmost flip-flop where data is shifted out.

Table B1 defines the initial content of the register.

**Table B.1 — Initial content of 16-stage shift register according to value '6363'**

| FF0 | FF1 | FF2 | FF3 | FF4 | FF5 | FF6 | FF7 | FF8 | FF9 | FF10 | FF11 | FF12 | FF13 | FF14 | FF15 |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|------|------|------|------|------|------|
| 0   | 1   | 1   | 0   | 0   | 0   | 1   | 1   | 0   | 1   | 1    | 0    | 0    | 0    | 1    | 1    |

Consequently, FF0 corresponds to the MSB and FF15 to the LSB.

#### Examples of bit patterns that will be transmitted via standard frames

EXAMPLE 1 Transmission of data, first byte = '00', second byte = '00', CRC\_A appended.

Calculated CRC\_A = '1EA0'

First bit transmitted

↓

|   |           |   |           |   |           |   |           |   |   |
|---|-----------|---|-----------|---|-----------|---|-----------|---|---|
| S | 0000 0000 | 1 | 0000 0000 | 1 | 0000 0101 | 1 | 0111 1000 | 1 | E |
|   | '00'      | P | '00'      | P | 'A0'      | P | '1E'      | P |   |

**Figure B.1 — Example 1 for CRC\_A encoding**

**Table B.2 — Content of 16-stage shift register according to value '1EA0'**

| FF0 | FF1 | FF2 | FF3 | FF4 | FF5 | FF6 | FF7 | FF8 | FF9 | FF10 | FF11 | FF12 | FF13 | FF14 | FF15 |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|------|------|------|------|------|------|
| 0   | 0   | 0   | 1   | 1   | 1   | 1   | 0   | 1   | 0   | 1    | 0    | 0    | 0    | 0    | 0    |

EXAMPLE 2 Transmission of data block, first byte = '12', second byte = '34', CRC\_A appended.

Calculated CRC\_A = 'CF26'

First bit transmitted

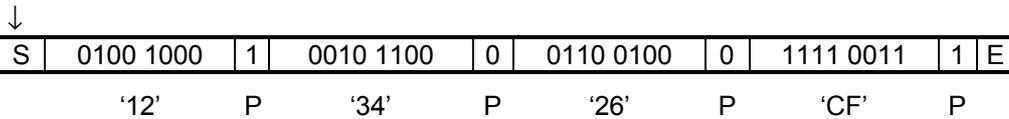


Figure B.2 — Example 2 for CRC\_A encoding

Table B.3 — Content of 16-stage shift register according to value 'CF26'

|     |     |     |     |     |     |     |     |     |     |      |      |      |      |      |      |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|------|------|------|------|------|------|
| FF0 | FF1 | FF2 | FF3 | FF4 | FF5 | FF6 | FF7 | FF8 | FF9 | FF10 | FF11 | FF12 | FF13 | FF14 | FF15 |
| 1   | 1   | 0   | 0   | 1   | 1   | 1   | 1   | 0   | 0   | 1    | 0    | 0    | 1    | 1    | 0    |

**CRC\_B encoding**

This annex is provided for explanatory purposes and indicates the bit patterns that will exist in the physical layer. It is included for the purpose of checking an ISO/IEC 14443-3 Type B implementation of CRC\_B encoding. Refer to ISO/IEC 13239 and CCITT X.25 #2.2.7 and V.42 #8.1.1.6.1 for further details.

Initial Value = 'FFFF'

**Examples of bit patterns that will be transmitted via standard frames**

EXAMPLE 1 Transmission of first byte = '00', second byte = '00', third byte = '00', CRC\_B appended.

Calculated CRC\_B = 'C6CC'

|         |     |                 |                 |                 |              |      |     |
|---------|-----|-----------------|-----------------|-----------------|--------------|------|-----|
|         |     | <b>1st byte</b> | <b>2nd byte</b> | <b>3rd byte</b> | <b>CRC_B</b> |      |     |
| Frame = | SOF | '00'            | '00'            | '00'            | 'CC'         | 'C6' | EOF |

Figure B.3 — Example 1 for CRC\_B encoding

EXAMPLE 2 Transmission of first byte = '0F', second byte = 'AA', third byte = 'FF', CRC\_B appended.

Calculated CRC\_B = 'D1FC'

|         |     |                 |                 |                 |              |      |     |
|---------|-----|-----------------|-----------------|-----------------|--------------|------|-----|
|         |     | <b>1st byte</b> | <b>2nd byte</b> | <b>3rd byte</b> | <b>CRC_B</b> |      |     |
| Frame = | SOF | '0F'            | 'AA'            | 'FF'            | 'FC'         | 'D1' | EOF |

Figure B.4 — Example 2 for CRC\_B encoding

EXAMPLE 3 Transmission of first byte = '0A', second byte = '12', third byte = '34', fourth byte = '56', CRC\_B appended.

Calculated CRC\_B = 'F62C'

|         |     |                 |                 |                 |                 |              |      |     |
|---------|-----|-----------------|-----------------|-----------------|-----------------|--------------|------|-----|
|         |     | <b>1st byte</b> | <b>2nd byte</b> | <b>3rd byte</b> | <b>4th byte</b> | <b>CRC_B</b> |      |     |
| Frame = | SOF | '0A'            | '12'            | '34'            | '56'            | '2C'         | 'F6' | EOF |

Figure B.5 — Example 3 for CRC\_B encoding

**Code sample written in C language for CRC calculation**

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <ctype.h>

#define CRC_A 1
#define CRC_B 2
#define BYTE unsigned char

unsigned short UpdateCrc(unsigned char ch, unsigned short *lpwCrc)
{
    ch = (ch^(unsigned char)((*lpwCrc) & 0x00FF));

    ch = (ch^(ch<<4));

    *lpwCrc = (*lpwCrc >> 8)^((unsigned short)ch << 8)^((unsigned short)ch<<3)^((unsigned short)ch>>4);

    return(*lpwCrc);
}

void ComputeCrc(int CRCType, char *Data, int Length,
    BYTE *TransmitFirst, BYTE *TransmitSecond)
{
    unsigned char chBlock;
    unsigned short wCrc;

    switch(CRCType) {
        case CRC_A:
            wCrc = 0x6363; /* ITU-V.41 */
            break;
        case CRC_B:
            wCrc = 0xFFFF; /* ISO/IEC 13239 (formerly ISO/IEC 3309) */
            break;
        default:
            return;
    }

    do {
        chBlock = *Data++;
        UpdateCrc(chBlock, &wCrc);
    } while (--Length);
    if (CRCType == CRC_B)
        wCrc = ~wCrc; /* ISO/IEC 13239 (formerly ISO/IEC 3309) */

    *TransmitFirst = (BYTE) (wCrc & 0xFF);
    *TransmitSecond = (BYTE) ((wCrc >> 8) & 0xFF);

    return;
}

BYTE BuffCRC_A[10] = {0x12, 0x34};
BYTE BuffCRC_B[10] = {0x0A, 0x12, 0x34, 0x56};
unsigned short Crc;
BYTE First, Second;
FILE *OutFd;
int i;

```

```
int main(void)
{
    printf("CRC-16 reference results ISO/IEC 14443-3\n");
    printf("Crc-16 G(x) = x^16 + x^12 + x^5 + 1\n\n");
    printf("CRC_A of [ ");
    for(i=0; i<2; i++) printf("%02X ",BuffCRC_A[i]);
    ComputeCrc(CRC_A, BuffCRC_A, 2, &First, &Second);
    printf("] Transmitted: %02X then %02X.\n", First, Second);
    printf("CRC_B of [ ");
    for(i=0; i<4; i++) printf("%02X ",BuffCRC_B[i]);
    ComputeCrc(CRC_B, BuffCRC_B, 4, &First, &Second);
    printf("] Transmitted: %02X then %02X.\n", First, Second);

    return(0);
}
```