INTERNATIONAL STANDARD

ISO/IEC 14443-4

Third edition
2016-06-01

Identification cards — Contactless integrated circuit cards — Proximity cards —

Part 4: Transmission protocol

Cartes d'identification — Cartes à circuit intégré sans contact — Cartes de proximité —

Partie 4: Protocole de transmission

Reference number
ISO/IEC 14443-4:2016(E)

ISO copyright office
Ch. de Blandonnet 8 • CP 401
CH-1214 Vernier, Geneva, Switzerland
Tel. +41 22 749 01 11
Fax +41 22 749 09 47
copyright@iso.org
www.iso.org

Contents

Page

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.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular the different approval criteria needed for the different types of document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

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. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation on the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the WTO principles in the Technical Barriers to Trade (TBT) see the following URL: Foreword - Supplementary information

The committee responsible for this document is ISO/IEC JTC 1, *Information technology*, SC 17, *Cards and personal identification*.

This third edition cancels and replaces the second edition (ISO/IEC 14443-4:2008), which has been technically revised. It also incorporates the Amendments ISO/IEC 14443-4:2008/Amd 1:2012, ISO/IEC 14443-4:2008/Amd 2:2012, ISO/IEC 14443-4:2008/Amd 3:2013 and ISO/IEC 14443-4:2008/Amd 4:2014.

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

— *Part 1: Physical characteristics*

— *Part 2: Radio frequency power and signal interface*

— *Part 3: Initialization and anticollision*

— *Part 4: Transmission protocol*

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.

The protocol, as defined in this part of ISO/IEC 14443, is capable of transferring the application protocol data units as defined in ISO/IEC 7816-4. Thus, application protocol data units may be mapped as defined in ISO/IEC 7816-4 and application selection may be used as defined ISO/IEC 7816-5.

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 and near field communication (NFC) devices conforming to ISO/IEC 18092 and ISO/IEC 21481.

The International Organization for Standardization (ISO) and International Electrotechnical Commission (IEC) draw attention to the fact that it is claimed that compliance with this International Standards may involve the use of patents.

Identification cards — Contactless integrated circuit cards — Proximity cards —

Part 4: Transmission protocol

1 Scope

This part of ISO/IEC 14443 specifies a half-duplex block transmission protocol featuring the special needs of a contactless environment and defines the activation and deactivation sequence of the protocol.

This part of ISO/IEC 14443 is intended to be used in conjunction with other parts of ISO/IEC 14443 and is applicable to proximity cards or objects of Type A and Type B.

2 Normative references

The following documents, in whole or in part, are normatively referenced in this document and are indispensable for its application. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 7816-3, *Identification cards — Integrated circuit cards — Part 3: Cards with contacts — Electrical interface and transmission protocols*

ISO/IEC 7816-4, *Identification cards — Integrated circuit cards — Part 4: Organization, security and commands for interchange*

ISO/IEC 14443-2, *Identification cards — Contactless integrated circuit cards — Proximity cards — Part 2: Radio frequency power and signal interface*

ISO/IEC 14443-3, *Identification cards — Contactless integrated circuit cards — Proximity cards — Part 3: Initialization and anticollision*

3 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

3.1
bit duration
one elementary time unit (etu), calculated by the following formula:

$$1 \text{ etu} = 128 / \left(D \times fc\right)$$

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

3.2
block
special type of frame, which contains a valid protocol data format

Note 1 to entry: A valid protocol data format includes I-blocks, R-blocks or S-blocks.

3.3
invalid block
type of frame, which contains an invalid protocol format

Note 1 to entry: A time-out, when no frame has been received, is not interpreted as an invalid block.

3.4
frame
sequence of bits as defined in ISO/IEC 14443-3

Note 1 to entry: The PICC independent from its type may use the frame with error correction defined in Clause 10. Alternatively, the PICC Type A can use one of the standard frames defined for Type A and the PICC Type B can use the frame defined for Type B. This Type B frame is called standard frame, too, within this part of ISO/IEC 14443.

4 Symbols and abbreviated terms

For the purposes of this part of ISO/IEC 14443, the following symbols and abbreviated terms apply.

A	Hamming control bits generation matrix (6 rows, 56 columns)
ACK	positive ACKnowledgement
ATS	Answer To Select
ATQA	Answer To reQuest, Type A
ATQB	Answer To reQuest, Type B
CID	Card IDentifier
CRC	Cyclic Redundancy Check, as defined for each PICC Type in ISO/IEC 14443-3
CRC1	most significant byte of CRC (b16 to b9)
CRC2	least significant byte of CRC (b8 to b1)
CRC_32	Cyclic Redundancy Check error detection code used within enhanced block
c_n	Hamming control bit n
$\underline{\boldsymbol{d}}$	vector containing 56 data bits
d_n	data bit n
D	Divisor
DR	Divisor Receive (PCD to PICC)
DRI	Divisor Receive Integer (PCD to PICC)
DS	Divisor Send (PICC to PCD)
DSI	Divisor Send Integer (PICC to PCD)
EDC	Error Detection Code
etu	elementary time unit
fc	carrier frequency
FSC	Frame Size for proximity Card
FSCI	Frame Size for proximity Card Integer
FSD	Frame Size for proximity coupling Device
FSDI	Frame Size for proximity coupling Device Integer
FWI	Frame Waiting time Integer
FWT	Frame Waiting Time

FWT_{TEMP}	temporary Frame Waiting Time
$\boldsymbol{H}$	matrix needed to calculate Hamming syndrome $\underline{s}$ (6 rows, 62 columns)
$h'_{m,n}$	element in row m and column n of matrix $\boldsymbol{H}'$
$\boldsymbol{H}'$	matrix needed to get matrix A (6 rows, 62 columns)
$\underline{\boldsymbol{h}'}_n$	column vector of matrix $\boldsymbol{H}'$
HLTA	HALT Command, Type A
$\boldsymbol{I}_{6\times6}$	6 by 6 Identity matrix
I-block	Information block
INF	INformation field
LEN	two bytes LENgth field used within enhanced block
m	row index
MAX	index to define a MAXimum value
MIN	index to define a MINimum value
n	column index
NAD	Node ADdress
NAK	Negative AcKnowledgement
OSI	Open Systems Interconnection
PCB	Protocol Control Byte
PCD	Proximity Coupling Device
PICC	Proximity Card or Object
PPS	Protocol and Parameter Selection
PPSS	Protocol and Parameter Selection Start
PPS0	Protocol and Parameter Selection parameter 0
PPS1	Protocol and Parameter Selection parameter 1
R-block	Receive ready block
R(ACK)	R-block containing a positive acknowledge
R(NAK)	R-block containing a negative acknowledge
RATS	Request for Answer To Select
REQA	REQuest Command, Type A
RFU	Reserved for Future Use by ISO/IEC
$\underline{\boldsymbol{s}}$	6-bit vector containing Hamming syndrome
s'	error position code
s	error position
S-block	Supervisory block
SAK	Select AcKnowledge
SFGI	Start-up Frame Guard time Integer
SFGT	Start-up Frame Guard Time
SYNC	SYNChronization sequence
WUPA	Wake-UP command, Type A
WTX	Waiting Time eXtension
WTXM	Waiting Time eXtension Multiplier
$\underline{\boldsymbol{y}}$	64-bit vector ($\underline{\boldsymbol{y}'}$ with no padding bits)
$\underline{\boldsymbol{y}'}$	64-bit vector containing received modified Hamming sub-block
y'_n	received bit n in each modified Hamming sub-block

For the purposes of this part of ISO/IEC 14443, the following notations apply.

— (xxxxx)b data bit representation;

— 'XY' hexadecimal notation, equal to XY to the base 16.

5 Protocol activation of PICC Type A

The following activation sequence shall be applied.

— PICC activation sequence as defined in ISO/IEC 14443-3 (request, anticollision loop and select).

— The SAK byte shall be checked to get information if the PICC is compliant with ISO/IEC 14443-4. The SAK byte is defined in ISO/IEC 14443-3.

— The PICC may be set to HALT state, using the HLTA Command as defined in ISO/IEC 14443-3, if e.g. no ISO/IEC 14443-4 protocol is used at the PCD.

NOTE The PCD cannot continue the activation sequence in that case.

— If the PICC is compliant to ISO/IEC 14443-4, the RATS may be sent by the PCD as next command after receiving the SAK.

— The PICC shall send its ATS as answer to the RATS. The PICC shall only answer to the RATS if the RATS is received directly after the selection.

— If the PICC supports any changeable parameters in the ATS, a PPS request may be used by the PCD as the next command after receiving the ATS to change parameters.

— The PICC shall send a PPS Response as answer to the PPS request.

A PICC does not need to implement the PPS, if it does not support any changeable parameters in the ATS.

The PCD activation sequence for a PICC Type A is shown in Figure 1.

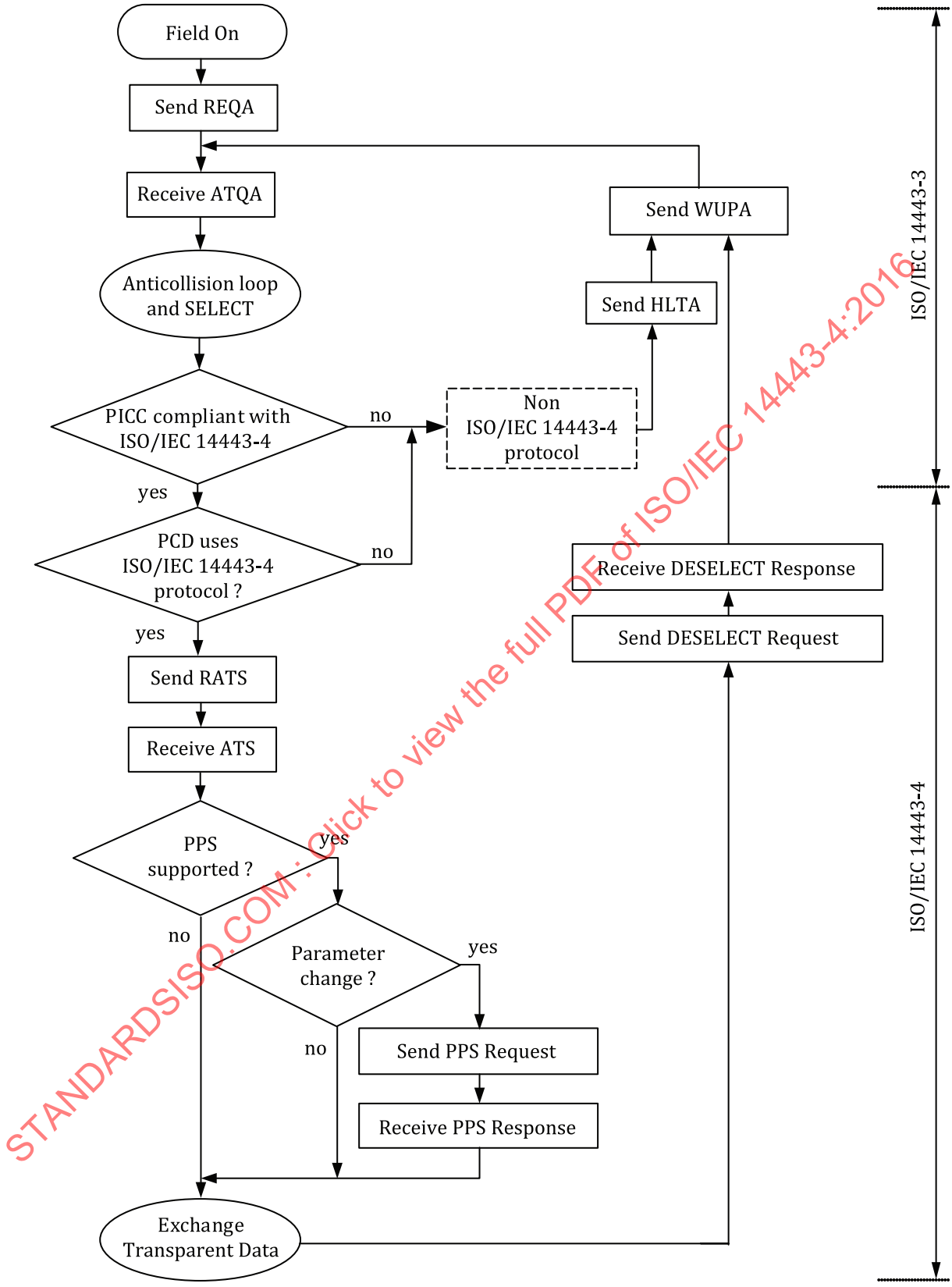


Figure 1 — Activation of a PICC Type A by a PCD

5.1 Request for answer to select

This Clause defines the RATS with all its fields (see Figure 2).

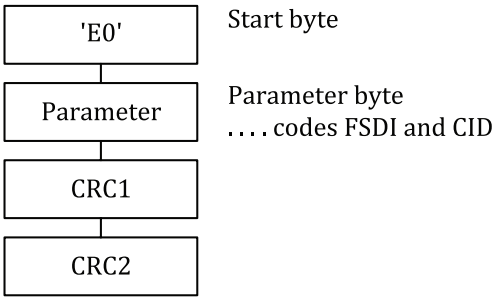


Figure 2 — Request for answer to select

The parameter byte consists of two parts (see Figure 3).

— The most significant half-byte b8 to b5 is called FSDI and codes FSD. The FSD defines the maximum size of a frame the PCD is able to receive. The coding of FSD is given in Table 1.

— A PCD setting FSDI = 'D'–'F' is not compliant with this part of ISO/IEC 14443. Until the RFU values 'D'–'F' are assigned by ISO/IEC, a PICC receiving value of FSDI = 'D'–'F' should interpret it as FSDI = 'C' (FSD = 4 096 bytes).

NOTE This PCD recommendation is added for PCD's compatibility with future PICC's when ISO/IEC defines the behaviour for the RFU values of 'D'–'F'.

— The least significant half byte b4 to b1 is named CID and it 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 all PICCs, which are in the ACTIVE state at the same time. The CID is fixed for the time the PICC is active and the PICC shall use the CID as its logical identifier, which is contained in the first error-free RATS received;

— A PCD setting CID = 15 is not compliant with this part of ISO/IEC 14443. For PICC behaviour see 5.6.1.2 c).

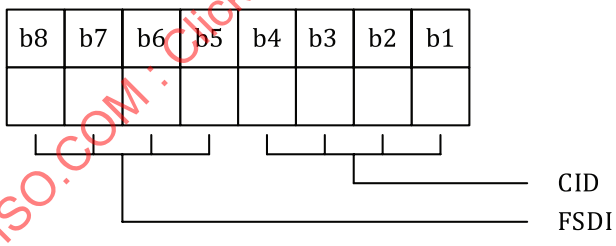


Figure 3 — Coding of RATS parameter byte

Table 1 — FSDI to FSD conversion

FSDI	'0'	'1'	'2'	'3'	'4'	'5'	'6'	'7'	'8'	'9'	'A'	'B'	'C'	'D'-'F'
FSD (bytes)	16	24	32	40	48	64	96	128	256	512	1 024	2 048	4 096	RFU

5.2 Answer to select

This Clause defines the ATS with all its available fields (see Figure 4).

In the case that one of the defined fields is not present in an ATS sent by a PICC, the default values for that field shall apply.

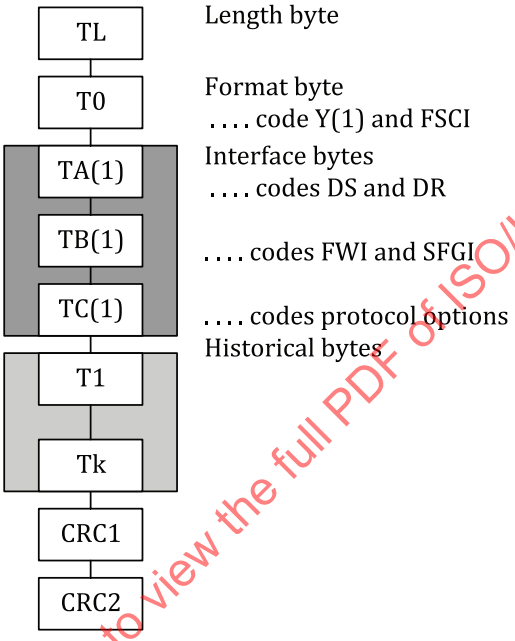


Figure 4 — Structure of the ATS

5.2.1 Structure of the bytes

The length byte TL is followed by a variable number of optional subsequent bytes in the following order:

— format byte T0;

— interface bytes TA(1), TB(1), TC(1);

— historical bytes T1 to Tk.

5.2.2 Length byte

The length byte TL is mandatory and specifies the length of the transmitted ATS including itself. The two CRC bytes are not included in TL. The maximum size of the ATS shall not exceed the indicated FSD. Therefore, the maximum value of TL shall not exceed FSD-2.

5.2.3 Format byte

The format byte T0 is optional and is present as soon as the length is greater than 1. The ATS can only contain the following optional bytes when this format byte is present.

T0 consists of three parts (see Figure 5).

— The most significant bit b8 shall be set to (0)b. The value (1)b is RFU.

— The bits b7 to b5 contain Y(1) indicating the presence of subsequent interface bytes TC(1), TB(1) and TA(1).

— The least significant half byte b4 to b1 is called FSCI and codes FSC. The FSC defines the maximum size of a frame accepted by the PICC. The default value of FSCI is 2 and leads to a FSC of 32 bytes. The coding of FSC is equal to the coding of FSD (see Table 1).

— A PICC setting FSCI = 'D'–'F' is not compliant with this standard. Until the RFU values 'D'–'F' are assigned by ISO/IEC, a PCD receiving value of FSCI = 'D'–'F' should interpret it as FSCI = 'C (FSC = 4 096 bytes).

NOTE This PICC recommendation is added for PICC's compatibility with future PCDs when ISO/IEC defines the behaviour for the RFU values 'D'–'F'.

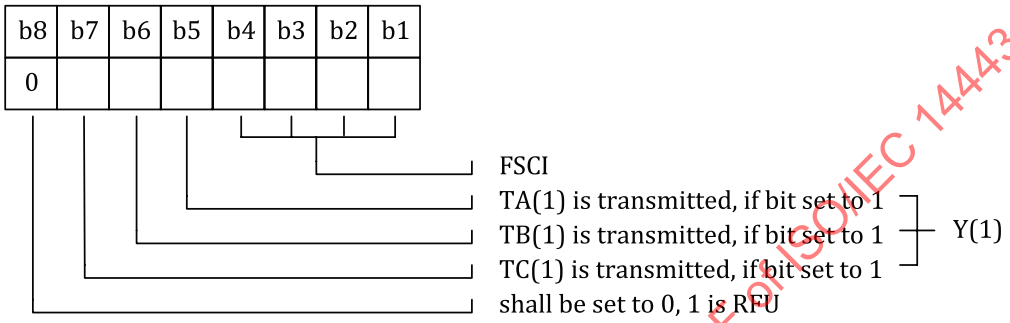


Figure 5 — Coding of format byte

5.2.4 Interface byte TA(1)

The interface byte TA(1) consists of four parts (see Figure 6).

— The most significant bit b8 codes the possibility to handle different divisors for each direction. When this bit is set to 1 the PICC is unable to handle different divisors for each direction.

— The bits b7 to b5 code the bit rate capability of the PICC for the direction from PICC to PCD, called DS. The default value shall be (000)b.

— The bit b4 shall be set to (0)b and the other value is RFU.

— The bits b3 to b1 code the bit rate capability of the PICC for the direction from PCD to PICC, called DR. The default value shall be (000)b.

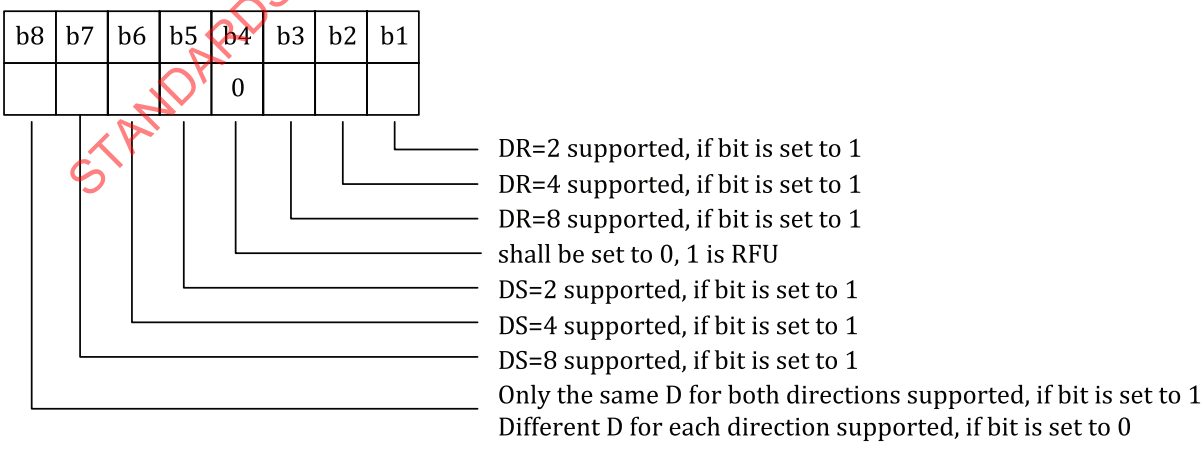


Figure 6 — Coding of interface byte TA(1)

The selection of a specific divisor D for each direction may be done by the PCD using PPS.

A PICC setting b4 = 1 is not compliant with this part of ISO/IEC 14443. A received value of TA(1) with b4 = 1 should be interpreted by the PCD as (b8 to b1) = (00000000)b (only ~106 kbit/s in both directions).

5.2.5 Interface byte TB(1)

The interface byte TB(1) conveys information to define the frame waiting time and the start-up frame guard time.

The interface byte TB(1) consists of two parts (see Figure 7).

— The most significant half-byte b8 to b5 is called FWI and codes FWT (see 7.2).

— The least significant half byte b4 to b1 is called SFGI and codes a multiplier value used to define the 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 ATS. SFGI is coded in the range from 0 to 14. The value of 15 is RFU. The value of 0 indicates no SFGT needed and the values in the range from 1 to 14 are used to calculate the SFGT with the formula given below. The default value of SFGI is 0.

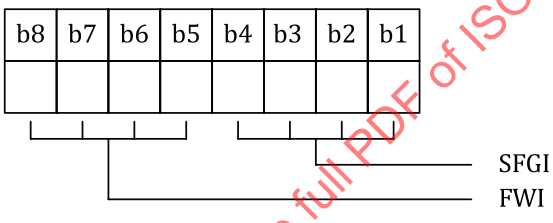


Figure 7 — Coding of interface byte TB(1)

SFGT is calculated by the following formulae:

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

$SFGT_{MIN}$ = minimum value of the frame delay time as defined in ISO/IEC 14443-3

$SFGT_{DEFAULT}$ = minimum value of the frame delay time as defined in ISO/IEC 14443-3

$SFGT_{MAX} = (256 \times 16/fc) \times 2^{14}$ (~4 949 ms)

A PICC setting SFGI = 15 is not compliant with this part of ISO/IEC 14443. Until the RFU value 15 is assigned by ISO/IEC, a PCD receiving SFGI = 15 should interpret it as SFGI = 0.

A PICC setting FWI = 15 is not compliant with this part of ISO/IEC 14443. Until the RFU value 15 is assigned by ISO/IEC, a PCD receiving FWI = 15 should interpret it as FWI = 4.

5.2.6 Interface byte TC(1)

The interface byte TC(1) specifies a parameter of the protocol.

The specific interface byte TC(1) consists of two parts (see Figure 8).

— The most significant bits b8 to b3 shall be (000000)b and all other values are RFU.

— The bits b2 and b1 define which optional fields in the prologue field a PICC does support. The PCD is allowed to skip fields, which are supported by the PICC, but a field not supported by the PICC shall never be transmitted by the PCD. The default value shall be (10)b indicating CID supported and NAD not supported.

— A PICC setting (b8 to b3) <> (000000)b is not compliant with this part of ISO/IEC 14443. The PCD should ignore (b8 to b3) and its interpretation of (b2,b1), or of any other field of the whole frame shall not change.

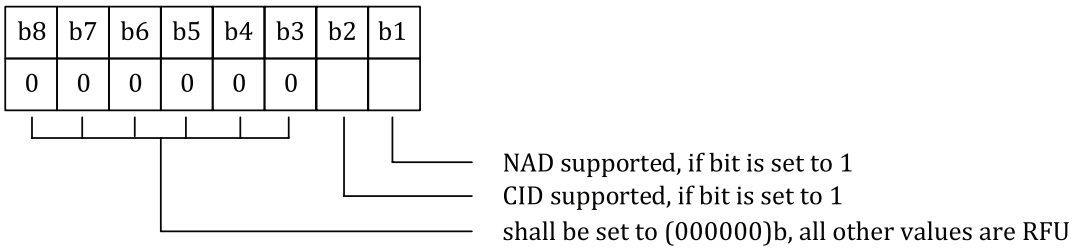


Figure 8 — Coding of interface byte TC(1)

5.2.7 Historical bytes

The historical bytes T1 to Tk are optional and designate general information. The maximum length of the ATS gives the maximum possible number of historical bytes. ISO/IEC 7816-4 specifies the content of the historical bytes.

5.3 Protocol and parameter selection request

PPS request contains the start byte that is followed by two parameter bytes (see Figure 9).

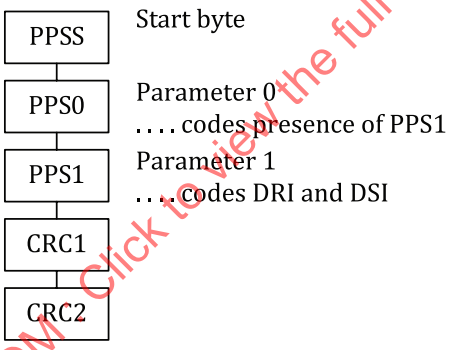


Figure 9 — Protocol and parameter selection request

5.3.1 Start byte

PPSS consists of two parts (see Figure 10).

— The most significant half byte b8 to b5 shall be set to (1101)b and identifies the PPS.

— The least significant half byte b4 to b1 is named CID and it defines the logical number of the addressed PICC.

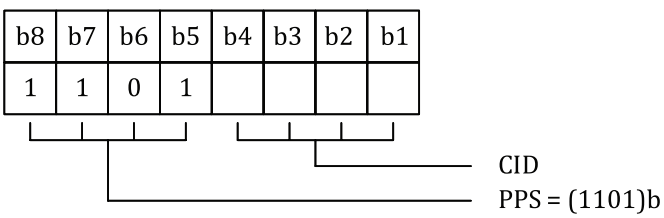


Figure 10 — Coding of PPSS

5.3.2 Parameter 0

PPS0 indicates the presence of the optional byte PPS1 (see Figure 11).

A PCD setting (b4 to b1) <> (0001)b and/or setting (b8 to b6) <> (000)b is not compliant with this part of ISO/IEC 14443. A PICC receiving (b4 to b1) <> (0001)b and/or receiving (b8 to b6) <> (000)b shall apply 5.6.2.2 b).

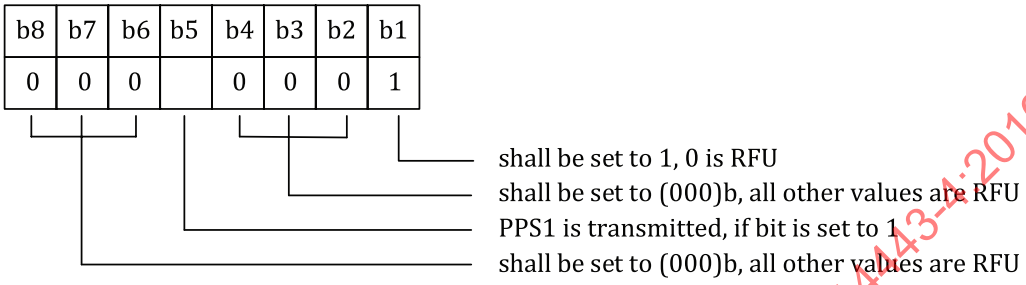


Figure 11 — Coding of PPS0

5.3.3 Parameter 1

PPS1 consists of three parts (see Figure 12).

— The most significant half byte b8 to b5 shall be (0000)b and all other values are RFU.

— The bits b4 and b3 are called DSI and code the selected divisor integer from PICC to PCD.

— The bits b2 and b1 are called DRI and code the selected divisor integer from PCD to PICC.

— A PCD setting (b8 to b5) <> (0000)b is not compliant with this part of ISO/IEC 14443. A PICC receiving (b8 to b5) <> (0000)b shall apply 5.6.2.2 b).

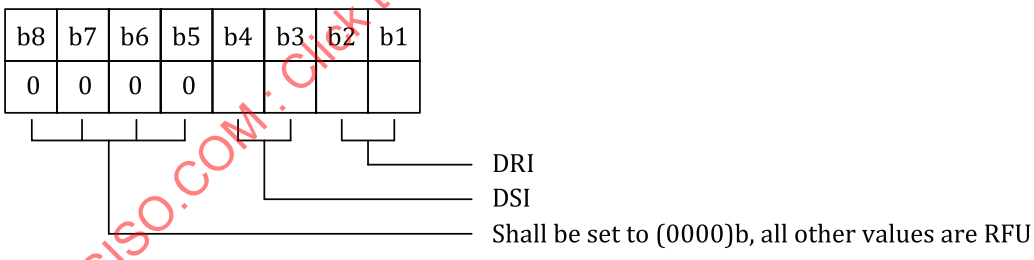


Figure 12 — Coding of PPS1

For the definition of DS and DR (see 5.2.4).

The coding of D is given in Table 2.

Table 2 — DRI, DSI to D conversion

DRI, DSI	(00)b	(01)b	(10)b	(11)b
D	1	2	4	8

5.4 Protocol and parameter selection response

The PPS response acknowledges the received PPS request (see Figure 13) and it contains only the start byte (see 5.3.1).

The new bit rates shall become effective in the PICC immediately after it has sent the PPS response. A PCD that changes the bit rate when the PPS response is missing or invalid or when the PPSS returned by the PICC is not identical with the PPSS sent by the PCD is not compliant with this part of ISO/IEC 14443.

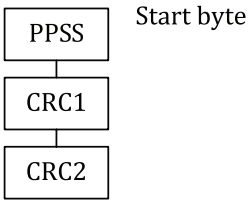


Figure 13 — Protocol and parameter selection response

5.5 Activation frame waiting time

The activation frame waiting time defines the maximum time for a PICC to start sending its response frame after the end of a frame received from the PCD and has a value of 65 536/f_c (~4 833 µs).

NOTE The minimum time between frames in any direction is defined in ISO/IEC 14443-3.

5.6 Error detection and recovery

5.6.1 Handling of RATS and ATS

5.6.1.1 PCD rules

When the PCD has sent the RATS and receives a valid ATS the PCD shall continue operation.

In any other case, the PCD may retransmit the RATS before it shall use the deactivation sequence as defined in Clause 8. In case of failure of this deactivation sequence, it may use the HLTA command as defined in ISO/IEC 14443-3.

5.6.1.2 PICC rules

When the PICC has been selected with the last command and

a) receives a valid RATS, the PICC
   - shall send back its ATS, and
   - shall disable the RATS (stop responding to received RATS);

b) receives a valid block (HLTA), the PICC
   - shall process the command and shall enter HALT state;

c) receives an invalid block or RATS with CID = 15, the PICC
   - shall not respond and shall enter IDLE state or HALT state as specified in ISO/IEC 14443-3:2016, Figure 7.

5.6.2 Handling of PPS request and PPS response

5.6.2.1 PCD rules

When the PCD has sent a PPS request and received a valid PPS response, the PCD shall activate the selected parameters and continue operation. In any other case, the PCD may retransmit a PPS request and continue operation.

5.6.2.2 PICC rules

When the PICC has received a RATS, sent its ATS and

a) received a valid PPS request, the PICC
   - shall send the PPS response,
   - shall disable the PPS request (stop responding to received PPS requests), and
   - shall activate the received parameter;

b) received an invalid block, the PICC
   - shall disable the PPS request (stop responding to received PPS requests), and
   - shall remain in receive mode;

c) received a valid block, except a PPS request, the PICC
   - shall disable the PPS request (stop responding to received PPS requests), and
   - shall continue operation.

5.6.3 Handling of the CID during activation

When the PCD has sent a RATS containing a CID = *n* not equal to 0 and

- received an ATS indicating CID is supported, the PCD
  - shall send blocks containing CID=n to this PICC, and
  - shall not use the CID=n for further RATS while this PICC is in ACTIVE state;
- received an ATS indicating CID is not supported, the PCD
  - shall send blocks containing no CID to this PICC, and
  - shall not activate any other PICC while this PICC is in ACTIVE state;

When the PCD has sent a RATS containing a CID equal to 0

- received an ATS indicating CID is supported, the PCD
  - may send blocks containing CID equal to 0 to this PICC, and
  - shall not activate any other PICC while this PICC is in ACTIVE state;
- received an ATS indicating CID is not supported, the PCD shall
  - send blocks containing no CID to this PICC, and
  - not activate any other PICC while this PICC is in ACTIVE state.

6 Protocol activation of PICC Type B

The activation sequence for a PICC Type B is described in ISO/IEC 14443-3.

7 Half-duplex block transmission protocol

The half-duplex block transmission protocol addresses the special needs of contactless card environments and uses the frame format as defined in ISO/IEC 14443-3.

Other relevant elements of the frame format are

— block format,

— maximum frame waiting time,

— power indication, and

— protocol operation.

A mechanism is provided in order to introduce additional protocol functions that may be defined from time to time in this part of ISO/IEC 14443 or in other standards that use this part of ISO/IEC 14443 as their foundation.

This protocol is designed according to the principle layering of the OSI reference model, with particular attention to the minimization of interactions across boundaries. Four layers are defined as follows:

— physical layer exchanges bytes according to ISO/IEC 14443-3;

— data link layer exchanges blocks as defined in this Clause;

— session layer combined with the data link layer for a minimum overhead;

— application layer processing commands, which involve the exchange of at least one block or chain of blocks in either direction.

Application selection may be used as defined in ISO/IEC 7816-4. Implicit application selection is not recommended to be used with multi-application PICCs.

7.1 Block format

The block format depends on the frame format used for its transmission.

The standard block format as specified in Figure 14 shall be used in standard frames as defined in ISO/IEC 14443-3 and consists of the following:

— a prologue field (mandatory);

— an information field (optional);

— a two-byte epilogue field (mandatory).

The enhanced block format specified in Figure 15 shall be used in frames with error correction as defined in Clause 10 and consists of the following:

— a length field (mandatory);

— a prologue field (mandatory);

— an information field (optional);

— a four-byte epilogue field (mandatory).

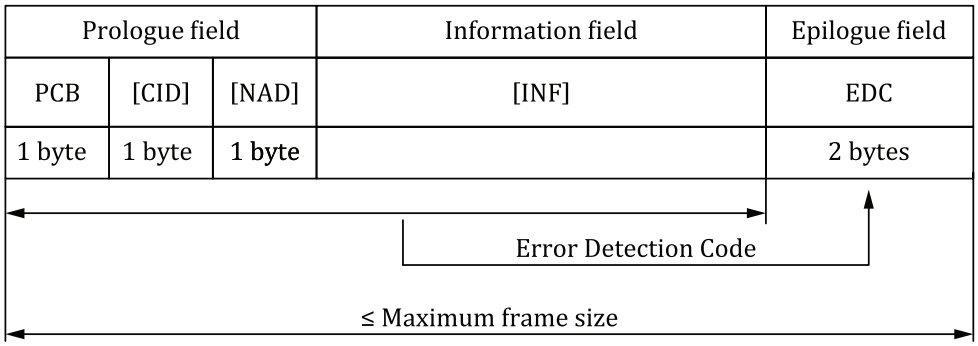


Figure 14 — Standard block format

NOTE 1 The items in brackets indicate optional requirements.

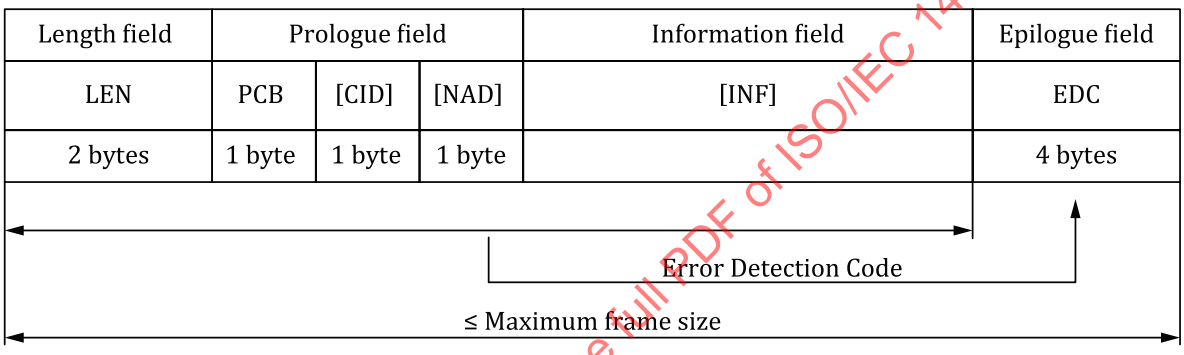


Figure 15 — Enhanced block format

NOTE 2 The items in brackets indicate optional requirements.

7.1.1 Length field

The two-byte length field shall contain the total number of bytes contained in the following fields:

— length field;

— prologue field;

— information field.

Least significant byte is transmitted first, then most significant byte.

7.1.2 Prologue field

The prologue field is mandatory and may be 1, 2, or 3 bytes with PCB mandatory and CID and NAD optional.

7.1.2.1 Protocol control byte field

The PCB is used to convey the information required to control the data transmission.

The protocol defines three fundamental types of blocks.

— I-block used to convey information for use by the application layer.

— R-block used to convey positive or negative acknowledgements. An R-block never contains an INF field. The acknowledgement relates to the last received block.

— S-block used to exchange control information between the PCD and the PICC. The support of the S(PARAMETERS) block is optional for PCDs and PICCs. Three different types of S-blocks are defined.

1) "Waiting time extension" containing a 1 byte long INF field,

2) "DESELECT" containing no INF field,

3) "PARAMETERS" containing a n-byte long INF field with $n \geq 0$.

FSD and FSC should be large enough to contain the expected S(PARAMETERS) blocks.

The coding of the PCB depends on its type and is defined by the following figures. PCB coding not defined here are either used in other clauses of ISO/IEC 14443 or are RFU. The coding of I-blocks, R-blocks and S-blocks are shown in Figures 16, 17 and 18.

A PICC or PCD setting b6 <> 0(b) of an I-block is not compliant with this part of ISO/IEC 14443. A PICC or PCD setting b2 <> 1(b) of an R-block is not compliant with this part of ISO/IEC 14443. A PICC or PCD setting (b1) <> (0)b of an S-block is not compliant with this part of ISO/IEC 14443.

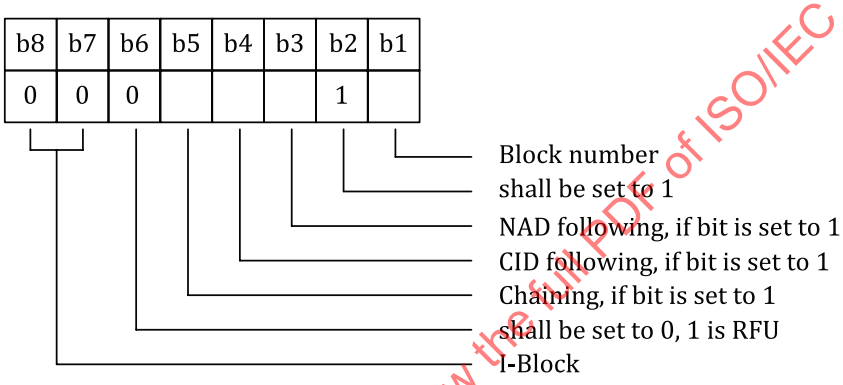


Figure 16 — Coding of I-block PCB

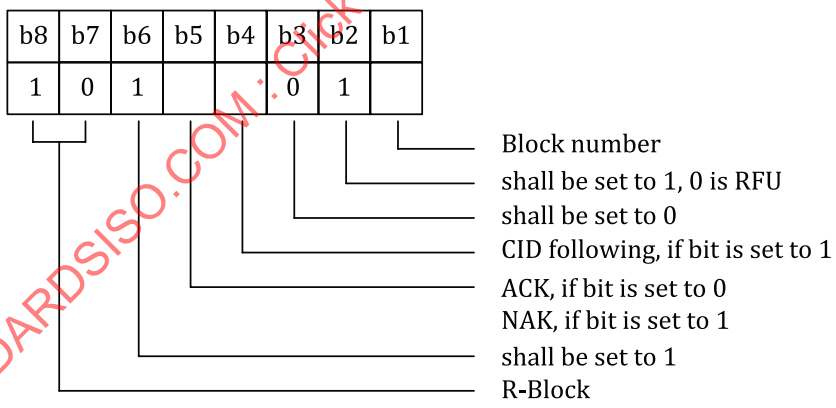


Figure 17 — Coding of R-block PCB

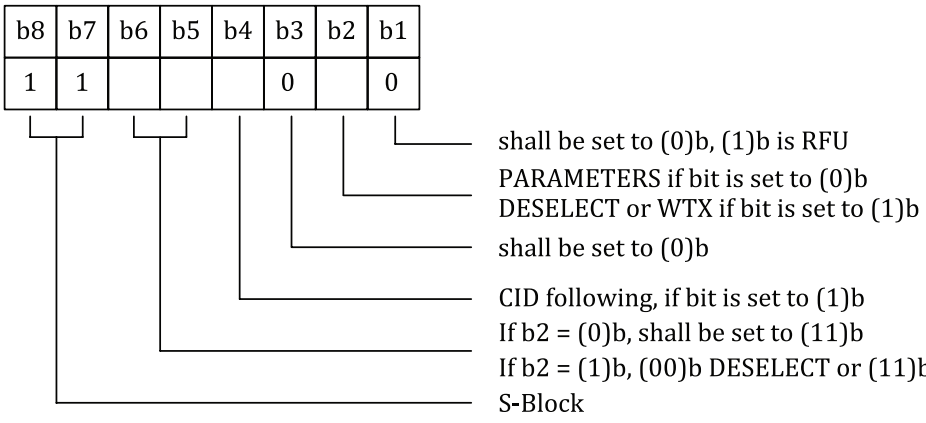


Figure 18 — Coding of S-block PCB

7.1.2.2 Card identifier field

The CID field is used to identify a specific PICC and consists of three parts (see Figure 19).

— The two most significant bits b8 and b7 are used to indicate the power level indication received by a PICC from a PCD. These two bits shall be set to (00)b for PCD to PICC communication. For a definition of power level indication (see 7.4).

— The bits b6 and b5 are used to convey additional information, which are not defined and shall be set to (00)b and all other values are RFU.

— A PICC or PCD setting (b6,b5) <> (00)b is not compliant with this part of ISO/IEC 14443. (b6,b5) <> (00)b shall be treated as a protocol error.

— The bits b4 to b1 code the CID.

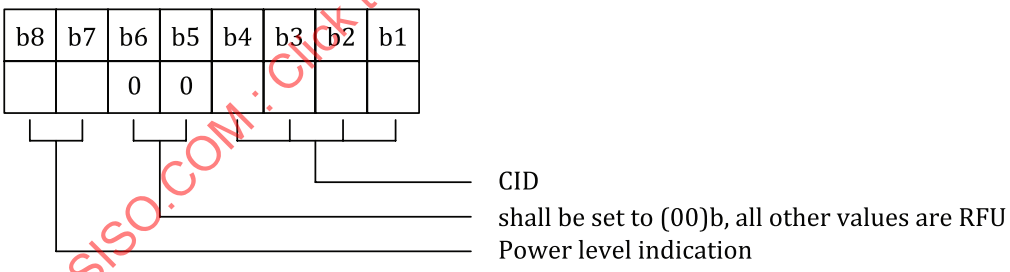


Figure 19 — Coding of card identifier

The coding of CID is given in 5.1 for Type A and ISO/IEC 14443-3 for Type B.

The handling of the CID by a PICC is described below:

A PICC, which does not support a CID

— shall ignore any block containing a CID.

A PICC, which does support a CID

— shall respond to blocks containing its CID by using its CID,

— shall ignore blocks containing other CIDs, and

— shall, in case its CID is 0, respond also to blocks containing no CID by using no CID.

7.1.2.3 Node address field

The NAD in the prologue field is reserved to build up and address different logical connections. The usage of the NAD shall be compliant with the definition from ISO/IEC 7816-3, when the bits b8 and b4 are both set to 0. All other values are RFU.

A PICC or PCD setting b8 <> 0 and/or b4 <> 0 is not compliant with this standard. b8 <> 0 and/or b4 <> 0 shall be treated as a protocol error.

The following definitions shall apply for the usage of the NAD:

a) the NAD field shall only be used for I-blocks;

b) when the PCD uses the NAD, the PICC shall also use the NAD;

c) during chaining the NAD shall only be transmitted in the first block of chain;

d) the PCD shall never use the NAD to address different PICCs (The CID shall be used to address different PICCs);

e) when the PICC does not support the NAD, it shall ignore any block containing the NAD.

7.1.3 Information field

The INF field is optional. When present, the INF field conveys either application data in I-blocks or non-application data and status information in S-blocks. The length of the information field is calculated by counting the number of bytes of the whole block minus length of prologue and epilogue field.

7.1.4 Epilogue field

The epilogue field contains the EDC of the transmitted block. A transmitted block shall be considered correct if it is received with a valid EDC value.

The EDC of standard blocks shall be the CRC defined in ISO/IEC 14443-3. Type A PICCs shall use CRC_A and Type B PICCs shall use CRC_B in both directions.

The EDC of enhanced blocks shall be CRC_32 as defined below.

The CRC_32 uses polynomial = '04C11DB7' with initial value = 'FFFFFFFF' and reflected bit order (LSB first). The final CRC value is bit-inverted before transmission and the least significant byte is transmitted first. Refer to ISO/IEC 13239 for further details. A code sample and an example are given in Annex E.

7.2 Frame waiting time

The FWT defines the time within which a PICC shall start its response frame after the end of a PCD frame (see Figure 20).

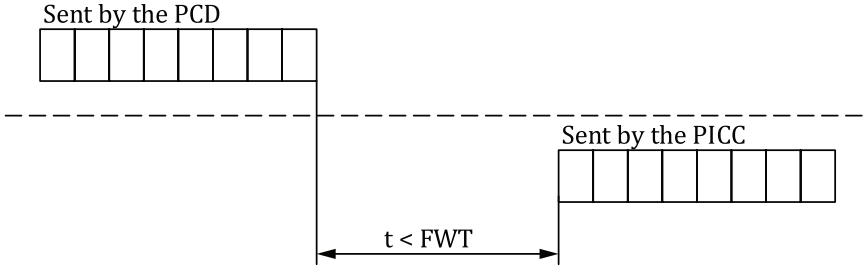


Figure 20 — Frame waiting time

NOTE 1 The minimum time between frames in any direction is defined in ISO/IEC 14443-3.

FWT is calculated by the following formula:

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

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

The default value of FWI is 4 (which gives a FWT value of ~4,8 ms) in the following cases:

— for Type A, if TB(1) is omitted, and

— for S(PARAMETERS) and S(DESELECT) blocks.

For FWI = 0, FWT = $\mathrm{FWT}_{\mathbf{MIN}}$ (~302 µs).

For FWI = 14, FWT = $\mathrm{FWT}_{\mathbf{MAX}}$ (~4 949 ms).

The FWT value shall be used by the PCD to detect a protocol error or an unresponsive PICC. The PCD obtains the right to re-transmit if the start of a response from the PICC is not received within FWT.

The FWI field for Type B is located in ATQB as defined in ISO/IEC 14443-3. The FWI field for Type A is located in the ATS (see 5.2.5).

The PICC shall not set FWI to the RFU value of 15. Until the RFU value 15 is assigned by ISO/IEC, a PCD receiving FWI = 15 should interpret it as FWI = 4.

NOTE 2 This Clause is added for PCD's compatibility with future PICCs when ISO/IEC defines the RFU value 15.

7.3 Frame waiting time extension

When the PICC needs more time than the defined FWT to process the received block, it shall use an S(WTX) request for a waiting time extension. An S(WTX) request contains a 1 byte long INF field that consists of two parts (see Figure 21).

— The two most significant bits b8 and b7 code the power level indication (see 7.4).

— A PCD not setting (b8,b7) = (00)b is not compliant with this part of ISO/IEC 14443. The PICC shall treat (b8,b7) <> (00)b as protocol error.

— The least significant bits b6 to b1 code the WTXM. The WTXM is coded in the range from 1 to 59. The values 0 and 60 to 63 are RFU.

— A PICC setting WTXM = 0 or WTXM = 60-63 is not compliant with this part of ISO/IEC 14443. When receiving WTXM = 0 or WTXM = 60-63 the PCD shall treat it as a protocol error.

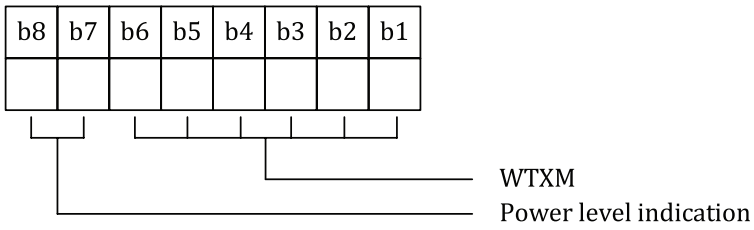


Figure 21 — Coding of INF field of an S(WTX) request

The PCD shall acknowledge by sending an S(WTX) response containing also a 1 byte long INF field that consists of two parts (see Figure 22) and contains the same WTXM as received in the request.

— The most significant bits b8 and b7 shall be (00)b and all other values are RFU.

— The least significant bits b6 to b1 codes the acknowledged WTXM value used to define a temporary FWT.

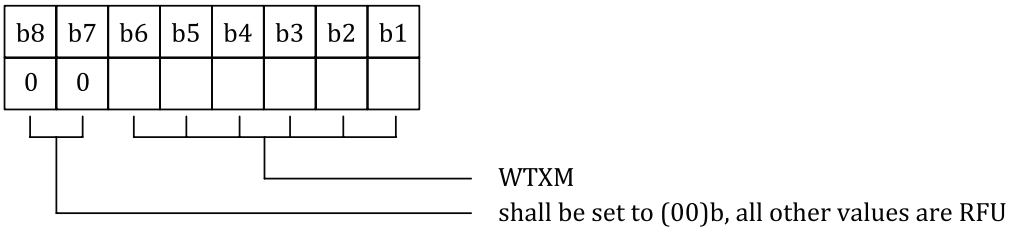


Figure 22 — Coding of INF field of an S(WTX) response

The corresponding temporary value of FWT is calculated by the following formula:

$$FWT_{TEMP} = FWT \times WTXM$$

The time FWT_{TEMP} requested by the PICC, starts after the PCD has sent the S(WTX) response.

FWT_{MAX} shall be used, when the formula results in a value higher than FWT_{MAX}.

The temporary FWT applies only until the next block has been received by the PCD.

7.4 Power level indication

The power level indication is coded as shown in Table 3 using two bits embedded in the CID field (when present) and in the S-block sent by the PICC (see 7.1.2.2 and 7.3).

Table 3 — Coding of power level indication

(00)b	PICC does not support the power level indication
(01)b	Insufficient power for full functionality
(10)b	Sufficient power for full functionality
(11)b	More than sufficient power for full functionality

NOTE Interpretation of the power level indication by the PCD is optional.

7.5 Protocol operation

After the activation sequence the PICC shall wait for a block as only the PCD has the right to send. After sending a block, the PCD shall switch to receive mode and wait for a block before switching back to transmit mode. The PICC may transmit blocks only in response to received blocks (it is insensitive to time delays). After responding, the PICC shall return to the receive mode.

The PCD shall not initiate a new pair of command/response until the current pair of command/response has been completed or if the frame waiting time is exceeded with no response.

7.5.1 S(PARAMETERS) blocks

After the activation sequence, the PCD may send at any time a first S(PARAMETERS) block with or without INF field to check if S(PARAMETERS) blocks are supported by the PICC.

This first PCD S(PARAMETERS) block and the PICC answer (if the PICC supports S(PARAMETERS) blocks) may contain information indicating the support of different application protocol types and/or other communication parameters.

The content of the S(PARAMETERS) INF field is defined in the relevant part of ISO/IEC 14443 and shall comply with the BER-TLV encoding rules for the context-specific class according to ISO/IEC 7816-4.

7.5.2 Multi-Activation

The Multi-Activation feature allows the PCD to hold several PICCs in the ACTIVE state simultaneously. It allows switching directly between several PICCs without needing additional time for deactivation of a PICC and activation of another PICC.

For an example of Multi-Activation, see Annex A.

NOTE The PCD needs to handle a separate block number for each activated PICC.

7.5.3 Chaining

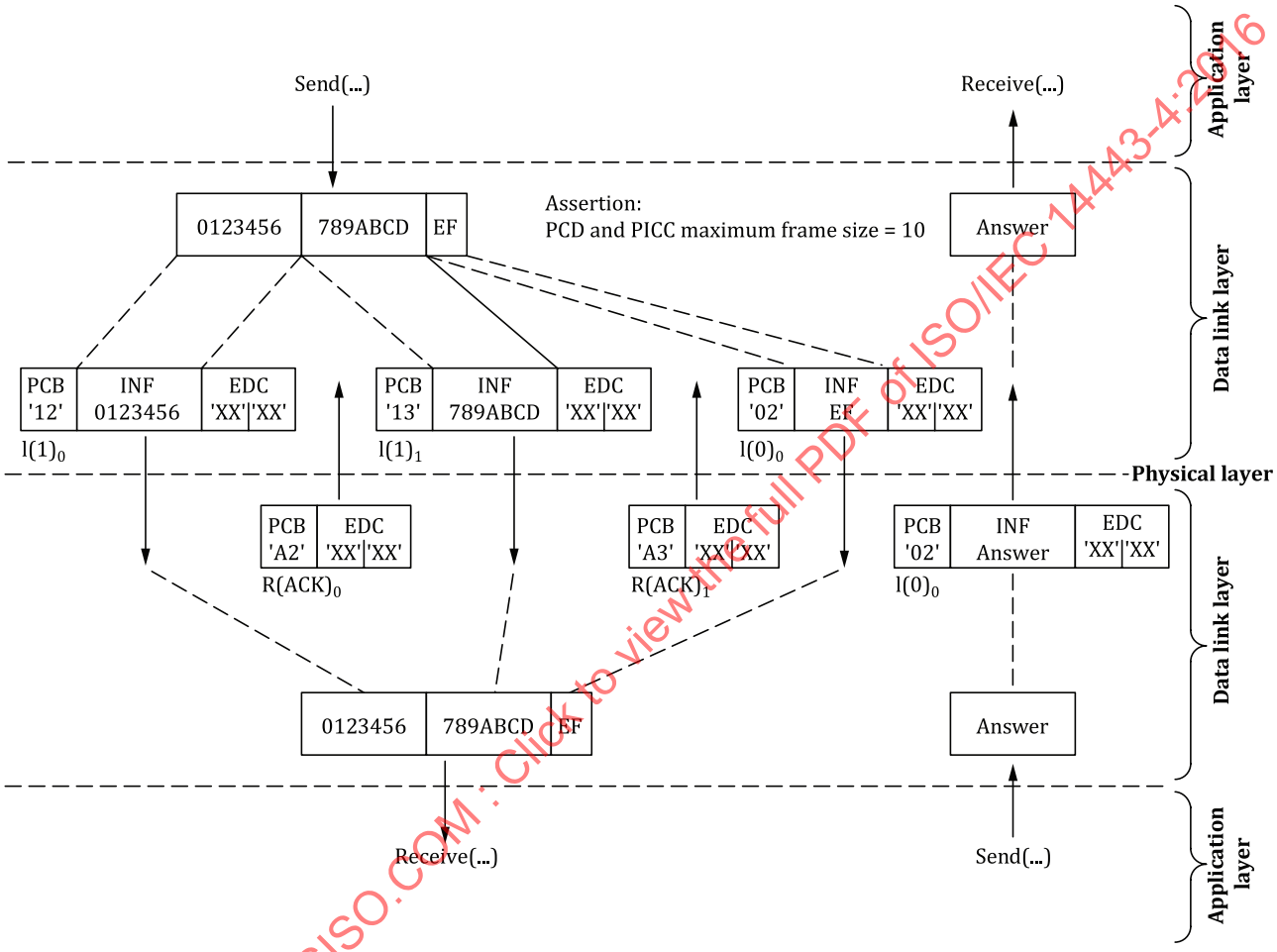
The chaining feature allows the PCD or PICC to transmit information that does not fit in a single block, as defined by FSC or FSD respectively, by dividing the information into several blocks. Each of those blocks shall have a length less than or equal to FSC or FSD respectively.

The chaining bit in the PCB of an I-block controls the chaining of blocks. Each I-block with the chaining bit set shall be acknowledged by an R-block.

The chaining feature is shown in Figure 23, using a 16 byte long string transmitted in three blocks.

Notation:

$I(1)_X$	I-block with chaining bit set and block number x;
$I(0)_X$	I-block with chaining bit not set (last block of chain) and block number x;
$R(ACK)_X$	R-block that indicates a positive acknowledge.



NOTE This example does not use the optional fields NAD and CID.

Figure 23 — Chaining

7.5.4 Block numbering rules

7.5.4.1 PCD rules

Rule A. The PCD block number shall be initialized to 0 for each activated PICC.

Rule B. When an I-block or an R(ACK) block with a block number equal to the current block number is received, the PCD shall toggle the current block number for that PICC before optionally sending a block.

7.5.4.2 PICC rules

Rule C. The PICC block number shall be initialized to 1 at activation.

Rule D. When an I-block is received, the PICC shall toggle its block number before sending a block.

NOTE 1 The PICC can check if the received block number is not in compliance with PCD rules to decide neither to toggle its internal block number nor to send a response block.

Rule E. When an R(ACK) block with a block number not equal to the current PICC's block number is received, the PICC shall toggle its block number before sending a block.

NOTE 2 There is no block number toggling when an R(NAK) block is received.

7.5.5 Block handling rules

7.5.5.1 General rules

Rule 1. The first block shall be sent by the PCD.

Rule 2. When an I-block indicating chaining is received, the block shall be acknowledged by an R(ACK) block.

Rule 3. S-blocks are only used in pairs. An S(...) request block shall always be followed by an S(...) response block (see 7.3 and Clause 8).

7.5.5.2 PCD rules

Rule 4. When an invalid block is received or a FWT time-out occurs, an R(NAK) block shall be sent [except in the case of PICC chaining or S(DESELECT) or S(PARAMETERS)].

Rule 5. In the case of PICC chaining, when an invalid block is received or a FWT time-out occurs, an R(ACK) block shall be sent.

NOTE 1 An R(ACK) block can be sent by the PCD only in case of PICC chaining, as the PICC response when receiving an R(ACK) block in other cases is not defined.

Rule 6. When an R(ACK) block is received, if its block number is not equal to the PCD's current block number, the last I-block shall be re-transmitted.

NOTE 2 The last I-block re-transmission is not required out of PCD chaining. The PCD can determine the presence of a PICC by sending R(NAK) blocks at any time out of chaining (including before sending any I-block) and receiving R(ACK) from the PICC if present.

Rule 7. When an R(ACK) block is received, if its block number is equal to the PCD's current block number, chaining shall be continued.

Rule 8. If the S(DESELECT)/S(PARAMETERS) request is not answered by an error-free S(DESELECT)/S(PARAMETERS) response the S(DESELECT)/S(PARAMETERS) request may be re-transmitted.

In case of not receiving an S(DESELECT) response after an S(DESELECT) request, the PICC may be ignored.

7.5.5.3 PICC rules

Rule 9. The PICC is allowed to send an S(WTX) block instead of an I-block or an R(ACK) block.

Rule 10. When an I-block not indicating chaining is received, the block shall be acknowledged by an I-block.

NOTE If the I-block received is empty then the mandatory I-block sent can either be empty or contain any applicative information (e.g. error code).

Rule 11. When an R(ACK) or an R(NAK) block is received, if its block number is equal to the PICC's current block number, the last block shall be re-transmitted.

Rule 12. When an R(NAK) block is received, if its block number is not equal to the PICC's current block number, an R(ACK) block shall be sent.

Rule 13. When an R(ACK) block is received, if its block number is not equal to the PICC's current block number, and the PICC is in chaining, chaining shall be continued.

7.5.6 PICC presence check

The following methods may be used to check the presence of a PICC at any time including before any I-block exchange.

The PCD shall not check the presence of a PICC until the current pair of command/response has been completed or when the frame waiting time is exceeded with no response.

7.5.6.1 Method 1

The PCD may send an empty I-block and expect to receive an I-block from the PICC.

7.5.6.2 Method 2

Before the first I-block exchange, the PCD may send an R(NAK) block (with block number 0) and expect to receive an R(ACK) (with block number 1) block from the PICC (rule 12).

After the first I-block exchange, the PCD may either

a) send an R(NAK) block (with current block number) and expect to receive an R(ACK) block from the PICC (rule 12), in which case the PCD should not retransmit its last I-block as mentioned in the note in rule 6, or

b) toggle its block number then send an R(NAK) block and expect to receive the last I-block from the PICC (rule 11).

7.5.7 Error detection and recovery

When at least one error is detected (after the optional error recovery mechanism, see 10.4.7) the following recovery rules shall be applied. The definitions made in this Clause overrule the block handling rules (see 7.5.4).

7.5.7.1 Errors detected by the PCD

The following errors shall be detected by the PCD.

a) Transmission error (Frame error or EDC error) or FWT time-out

The PCD shall attempt error recovery by the following techniques in the order shown:

— application of PCD rules (see 7.5.5.2);

— optionally apply PCD rules (see 7.5.5.2) once more;

— use of S(DESELECT) request;

— optionally use of S(DESELECT) request once more (as specified in 8.2);

— ignore the PICC.

b) Protocol error (infringement of PCB coding or infringement of protocol rules)

The PCD shall attempt error recovery by the following techniques in the order shown:

— use of S(DESELECT) request;

— ignore the PICC.

7.5.7.2 Errors detected by the PICC

The following errors shall be detected by the PICC:

a) transmission error (Frame error or EDC error);

b) protocol error (infringement of the protocol rules).

The PICC shall attempt no error recovery. The PICC shall always return to receive mode, when a transmission error or a protocol error occurs and it shall accept an S(DESELECT) request at any time.

NOTE An R(NAK) block is never sent by the PICC.

8 Protocol deactivation of PICC Type A and Type B

The PICC shall be set to the HALT state, after the transactions between PCD and PICC have been completed.

The deactivation of a PICC is done by using a DESELECT Command.

The DESELECT Command is coded as an S-block of the protocol and consists of an S(DESELECT) request block sent by the PCD and an S(DESELECT) response sent as acknowledge by the PICC.

8.1 Deactivation frame waiting time

The deactivation frame waiting time defines the maximum time for a PICC to start sending its S(DESELECT) response frame after the end of the S(DESELECT) request frame received from the PCD and has a value of 65 536/*fc* (~4,8 ms).

NOTE The minimum time between frames in any direction is defined in ISO/IEC 14443-3.

8.2 Error detection and recovery

When the PCD has sent an S(DESELECT) request and has received an S(DESELECT) response, the PICC has been set successfully to the HALT state and the CID assigned to it is released.

When the PCD fails to receive an S(DESELECT) response the PCD may retry the deactivation sequence.

9 Activation of bit rates and framing options in the PROTOCOL state

S(PARAMETERS) blocks shall be used to negotiate bit rates and communication parameters when the PICC is in PROTOCOL state. The information field shall contain tags and values as defined in Table 4, Table 5, Figure 24 and Figure 25.

The following rules shall be applied to negotiate those parameters:

— the PCD shall send an S(PARAMETERS) block to request parameters;

— if the PICC supports S(PARAMETERS) blocks, the PICC shall respond with an S(PARAMETERS) block containing values for all supported parameters. If the PICC does not support S(PARAMETERS) it shall stay mute.

After the PICC has sent its response and has indicated its parameters the PCD may activate one bit rate for each communication direction with the following rules:

— the PCD shall send an S(PARAMETERS) block to activate selected communication parameters;

— the PICC shall acknowledge the activated parameters with an S(PARAMETERS) block and then shall activate the negotiated parameters;

— the PCD shall activate the negotiated parameters.

Table 4 — S(PARAMETERS) tag definition

Tag (Hex)	Description	Length (Hex)	Value
'A0'	S(PARAMETERS) block information	L	Function Tags Identifier (see Table 5)

NOTE The length field is in accordance with the full range of BER-TLV (see ISO/IEC 7816-4).

Table 5 — Bit rates function tags identifier definition

Tag (Hex)	Description	Length (Hex)	Value		
'A1'	Bit rates Request	'0'	—		
'A2'	Bit rates Indication	L	Tag (Hex)	Length (Hex)	Value
			'80'	'02'	Supported bit rates from PCD to PICC 1st byte is specified in Figure 24 2nd byte set to '00', other values are RFU
			'81'	'02'	Supported bit rates from PICC to PCD 1st byte is specified in Figure 24 2nd byte set to '00', other values are RFU
			'82'	'01'	Supported framing options PICC to PCD (see Figure 25)
'A3'	Bit rates Activation	L	Tag (Hex)	Length (Hex)	Value
			'83'	'02'	Selected bit rate from PCD to PICC[a] 1st byte is specified in Figure 24 2nd byte set to '00', other values are RFU
			'84'	'02'	Selected bit rate from PICC to PCD[a] 1st byte is specified in Figure 24 2nd byte set to '00', other values are RFU
			'85'	'01'	Selected framing options PICC to PCD (see Figure 24)[b]
'A4'	Bit rates Acknowledgement	'0'	—		

[a] The PCD shall set only one bit. The PCD shall not activate simultaneously a bit rate higher than *fc*/16 for PCD to PICC communication and a bit rate of *fc*/128 for PICC to PCD communication in Type A.

[b] The PCD shall not set both b1 (start bit and stop bit suppression) and b2 (SOF and EOF suppression). When the PCD sets b1 (start bit and stop bit suppression):

— The PICC shall use a SOF low time of 10 etu and a SOF high time of 2 etu.

— The PICC shall use an EOF low time of 10 etu.

— The PICC shall apply no character separation.

Only relevant objects should be sent. It is even possible to send an empty parent object with no children objects (i.e. 'A0 00'), although it is also possible to send an empty S(PARAMETERS) block (i.e. not even the parent object sent).

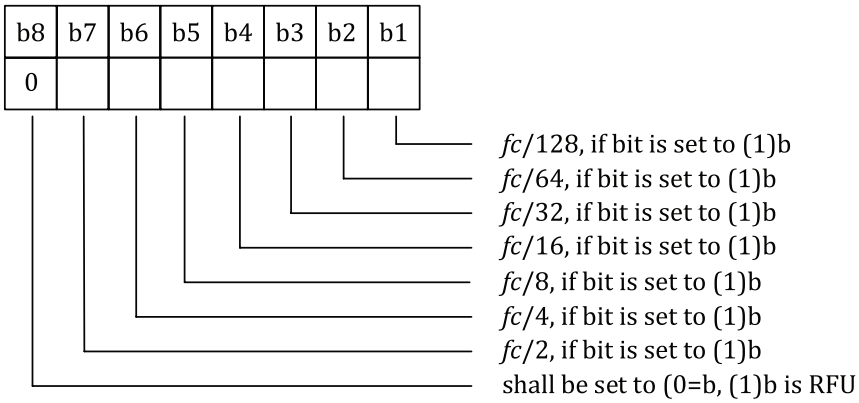


Figure 24 — Coding of bit rates

For bit rates of 3*fc*/4, *fc*, 3*fc*/2 and 2*fc*, see Annex D.

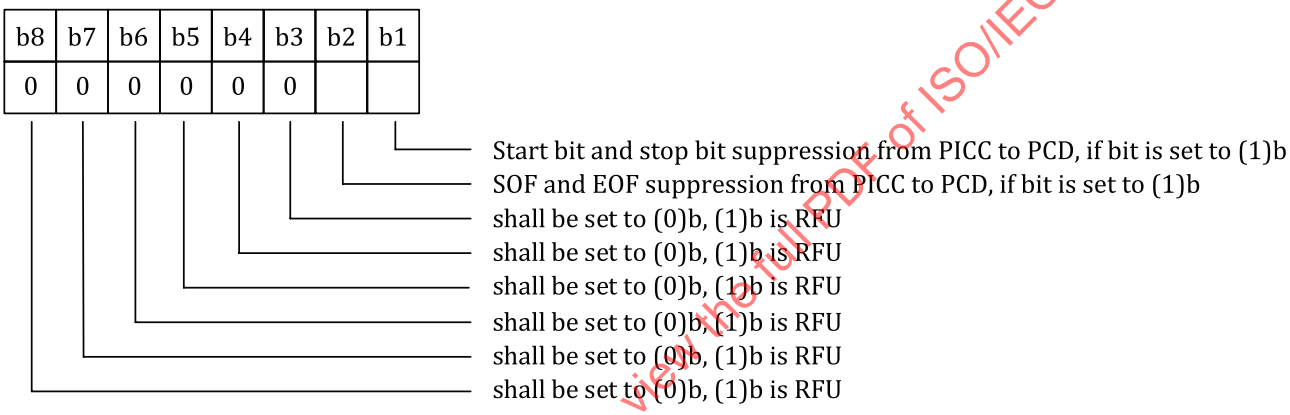


Figure 25 — Framing options

As an example, the sequence for an activation of the bit rate

— *fc*/8 from PCD to PICC, and

— *fc*/2 from PICC to PCD

with a PICC indicating to support

— bit rates *fc*/128, *fc*/16 and *fc*/8 for PCD to PICC communication,

— bit rates *fc*/128, *fc*/16 and *fc*/2 for PICC to PCD communication, and

— no framing options

is illustrated in Figure 26.

Step	PCD		PICC
1	S(PARAMETERS)('A0 02 A1 00' CRC)	⟶	
2		⟵	S(PARAMETERS) ('A0 0A' 'A2 08' '80 02 19 00' '81 02 49 00' CRC)
3	S(PARAMETERS) ('A0 0A' 'A3 08' '83 02 10 00' '84 02 40 00' CRC)	⟶	
4		⟵	S(PARAMETERS)('A0 02 A4 00' CRC)

Figure 26 — Bit rates activation example

10 Frame with error correction

10.1 General

Frames with error correction as specified in 10.2 and 10.3 shall be used after their activation as specified in 10.5. An example is given in Annex F.

10.2 Type A PCD frame format for bit rates up to *fc*/16 and higher than *fc*/2 and Type A PICC frame format for all bit rates

Frames with error correction, as defined in Figure 27, shall be used for data exchange and consist of, in the following order:

— start of communication;

— SYNC;

— enhanced block with error correction (see 10.4);

— end of communication.

SYNC consists of six dedicated bytes with the values ‘55’, ‘55’, ‘74’, ‘74’, ‘74’ and ‘74’ transmitted in this order.

SYNC and enhanced blocks with error correction shall be transmitted as bytes consisting of 8 bits.

NOTE Parity bits (see ISO/IEC 14443-3:2016, 6.2.3.2) are not used.

S	SYNC	Enhanced block with error correction	E

Figure 27 — Frame with error correction

10.3 Type A PCD frame format for bit rates of *fc*/8, *fc*/4 and *fc*/2 and Type B PCD and PICC frame format for all bit rates

Frames with error correction, as defined in Figure 28, shall be used for data exchange and consist of, in the following order:

— SOF as defined in ISO/IEC 14443-3:2016, 7.1.4;

— SYNC as defined in 10.2, transmitted as characters as defined in ISO/IEC 14443-3:2016, 7.1.1;

— enhanced block with error correction (see 10.4) transmitted as characters as defined in ISO/IEC 14443-3:2016, 7.1.1;

— EOF as defined in ISO/IEC 14443-3:2016, 7.1.5.

No character separation shall be applied in frames with error correction.

SOF, EOF, start bit, stop bit and SYNC may be suppressed in accordance with 10.5.

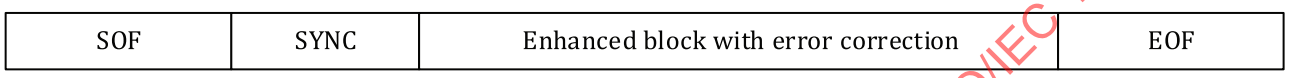


Figure 28 — Frame with error correction

10.4 Enhanced block with error correction

10.4.1 General

Enhanced block with error correction shall be composed of one or several 8-byte modified Hamming sub-blocks, each of them being calculated from 7-byte sub-blocks from enhanced block (see Figure 29).

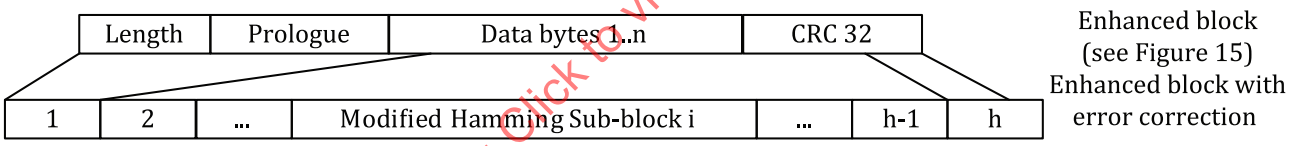


Figure 29 — Enhanced block with error correction

10.4.2 Modified Hamming sub-block format

Each modified Hamming sub-block shall consist of 7 bytes from enhanced block, followed by one Hamming control byte used to correct one single-bit error on the Hamming sub-block.

Modified Hamming sub-blocks shall always be complete. If necessary, 'FF' bytes shall be added to complete the last bytes from enhanced block to get 7 bytes.

10.4.3 Hamming control byte

The Hamming control byte shall contain the Hamming control bits c_n and logical "1" padding bits in the following order:

— one logical "1" padding bit;

— six Hamming control bits c_n in the order c_1, c_2, c_3, c_4, c_5, c_6;

— one logical "1" padding bit.

An example for Hamming control byte calculation in ANSI C language is given in F.2.

10.4.4 Hamming control generation matrix *A*

Hamming control bits generation matrix ***A*** (see Figure 32) shall be generated by following steps:

— generate Matrix ***H′*** (see Figure 31) using formula in Figure 30;

— remove column vectors $\underline{h'}_n$, with n = 1, 2, 4, 8, 16 and 32, of ***H′***.

$$h'_{m,n} = \begin{cases} 1 \text{ for } (n \wedge 2^{m-1}) \neq 0 \\ 0 \text{ otherwise} \end{cases} \quad \text{with } 1 \leq m \leq 6 \text{ and } 1 \leq n \leq 62$$

Figure 30 — Matrix *H′* generation

NOTE $\wedge$ stands for a bitwise AND operation.

$$\boldsymbol{H'} = \begin{pmatrix} 1 & 0 & 1 & 0 & \cdots & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & \cdots & 1 & 1 & 1 \end{pmatrix}$$

Figure 31 — Matrix *H′*

$$\boldsymbol{A} = \begin{pmatrix} 1 & 1 & 0 & 1 & \cdots & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & \cdots & 1 & 1 & 1 \end{pmatrix}$$

Figure 32 — Hamming control generation matrix *A*

10.4.5 Hamming control bits calculation

Hamming control bits c_m (m = 1..6) shall be calculated over data d_n (n = 1..56) using the formula in Figure 33. d_1 is bit b1 of the first byte and d_{56} is bit b8 of the seventh byte of any 7-byte sub-block from enhanced block.

$$\underline{\boldsymbol{c}} = \boldsymbol{A} \times \underline{\boldsymbol{d}}$$

Figure 33 — Hamming control bits generation

10.4.6 Hamming control check matrix *H*

The Hamming control check matrix ***H*** (illustrated in Figure 34) is a concatenation of matrix ***A*** and matrix $\boldsymbol{I}_{6 \times 6}$.

$$\boldsymbol{H} = \boldsymbol{A}|\boldsymbol{I}_{6,6} = \begin{pmatrix} 1 & 1 & 0 & 1 & \cdots & 0 & 1 & 0 & 1 & 0 & \cdots & 0 & 0 \\ 1 & 0 & 1 & 1 & & 0 & 0 & 1 & 0 & 1 & & 0 & 0 \\ 0 & 1 & 1 & 1 & & 1 & 1 & 1 & 0 & 0 & & 0 & 0 \\ 0 & 0 & 0 & 0 & & 1 & 1 & 1 & 0 & 0 & & 0 & 0 \\ 0 & 0 & 0 & 0 & & 1 & 1 & 1 & 0 & 0 & & 1 & 0 \\ 0 & 0 & 0 & 0 & \cdots & 1 & 1 & 1 & 0 & 0 & \cdots & 0 & 1 \end{pmatrix}$$

Figure 34 — Hamming control check matrix *H*

10.4.7 Error correction

Hamming control bits shall be used to detect and correct any single bit error in modified Hamming sub-blocks.

The so called syndrome $\underline{\boldsymbol{s}}$ shall be calculated using formula in Figure 35. To get $\underline{\boldsymbol{y}}$ from $\underline{\boldsymbol{y}}'$, the padding bits of the received data $\underline{\boldsymbol{y}}'_n$ on position 57 and 64 shall be removed. $\underline{\boldsymbol{y}}'_1$ is bit b1 of the first received byte and y'_{64} is bit b8 of the eighth received byte of any 8-byte sub-block from enhanced block with error correction.

$$\underline{\boldsymbol{s}} = \boldsymbol{H} \times \underline{\boldsymbol{y}}$$

Figure 35 — Syndrome calculation

The numerical interpretation s' of the syndrome $\underline{\boldsymbol{s}}$ shall be used for error correction:

— if s′ = 0, 1, 2, 4, 8, 16, 32 or 63 no change in received bits y'_1 to y'_{56};

— else

  — calculate error position s by reducing s' by the amount of powers of 2 (1, 2, 4, 8, 16, 32) which are smaller than s′;

  — invert the received bit y'_s.

NOTE More than one bit error cannot be corrected by this method. EDC will detect these multiple errors with very high probability.

10.5 Activation of frame with error correction in the PROTOCOL state

S(PARAMETERS) blocks shall be used to negotiate the used frame and communication parameters in PROTOCOL state. The information field shall contain tags and values as defined in Table 4, Table 6, Figure 36 and Figure 37.

The following rules shall be applied to negotiate those parameters.

— The PCD shall send an S(PARAMETERS) block to request frame format parameters.

— If the PICC supports S(PARAMETERS) blocks, the PICC shall respond with an S(PARAMETERS) block containing values for all supported frame format parameters. If the PICC does not support S(PARAMETERS) it shall stay mute. The PICC shall always indicate the same framing options independent of which tag is used.

After the PICC has sent its response and has indicated its parameters, the PCD may activate the desired options for each communication direction with the following rules:

— the PCD shall send an S(PARAMETERS) block to activate selected frame format parameters;

— the PICC shall acknowledge the activated frame format parameters with an S(PARAMETERS) block and then shall activate the negotiated frame format parameters;

— the PCD shall activate the negotiated frame format parameters.

Table 6 — Frame format function tags identifier definition

Tags (Hex)	Description	Length (Hex)	Value		
'A5'	Frame Format Request	0	—		
'A6'	Frame Format Indication	L	Tags (Hex)	Length (Hex)	Value
			'80'	'01'	Supported frames PCD to PICC (see Figure 36)[a]
			'81'	'01'	Supported frames PICC to PCD (see Figure 36)[a]
			'82'	'01'	Supported framing options PCD to PICC (see Figure 37)[b]
			'83'	'01'	Supported framing options PICC to PCD (see Figure 37)[c]
'A7'	Frame Format Activation	L	Tags (Hex)	Length (Hex)	Value
			'84'	'01'	Selected frame PCD to PICC (see Figure 36)[d]
			'85'	'01'	Selected frame PICC to PCD (see Figure 36)[d]
			'86'	'01'	Selected framing options from PCD to PICC (see Figure 37)[be]
			'87'	'01'	Selected framing options from PICC to PCD (see Figure 37)[cef]
'A8'	Frame Format Acknowledge-ment	0	—		

a Bit 8 shall be set to the same value in both communication directions.

b Shall be omitted for Type A PICCs for bit rates up to *fc*/16.

c Shall be omitted for Type A PICCs.

d Bit 8 shall be set to (0)b and only one bit out of b1 and b2 shall be set to (1)b.

e When SYNC is suppressed the PCD shall not select both start bit and stop bit suppression and SOF and EOF suppression. If start bit and stop bit suppression is selected, SOF and EOF low time of 10 etu and SOF high time of 2 etu shall be used.

f When tag '87' is used, the corresponding tag of bit rates > *fc*/16 shall not be used and vice versa.

NOTE 1 The length field is in accordance with the full range of BER-TLV (see ISO/IEC 7816-4).

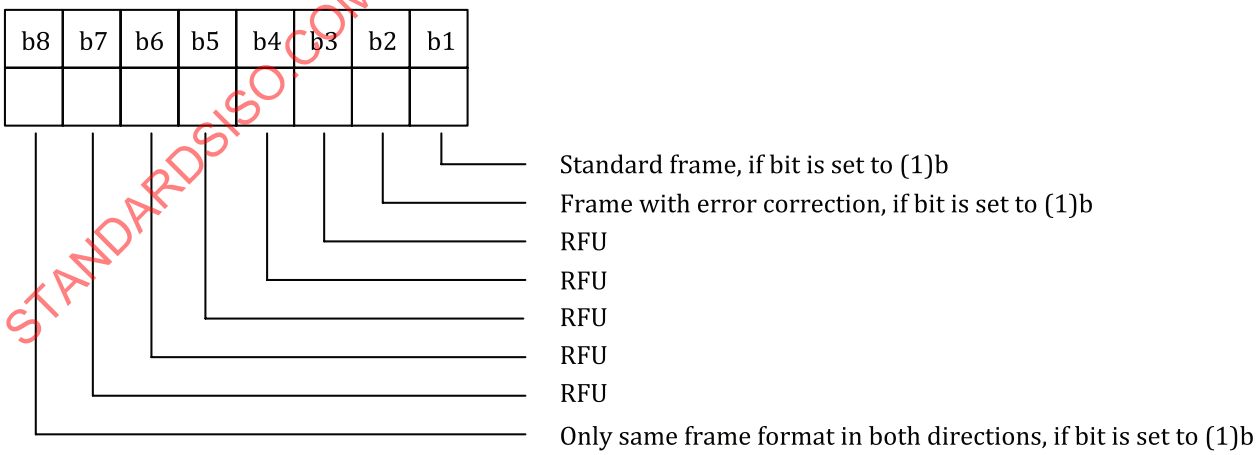


Figure 36 — Frame Formats

NOTE 2 Standard frame support is mandatory.

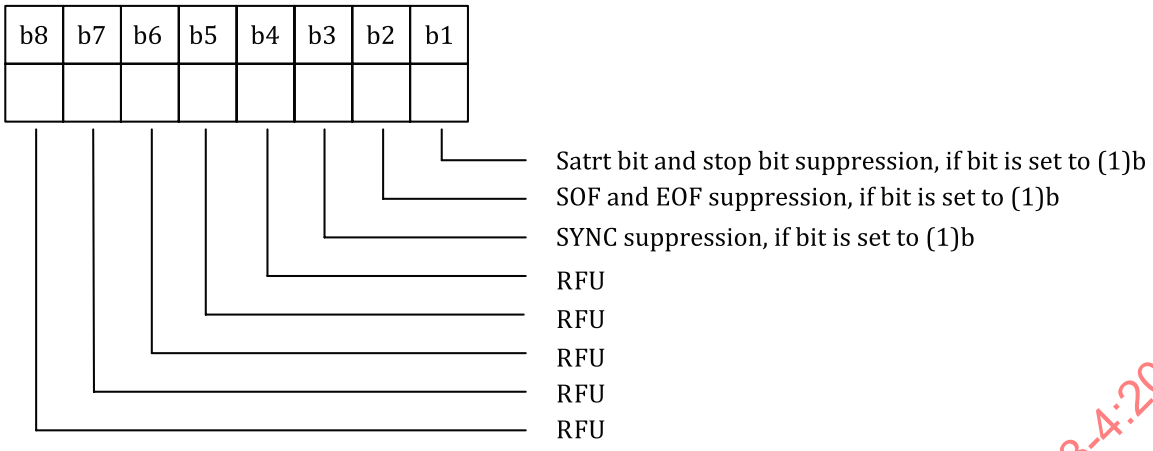


Figure 37 — Framing options

NOTE 3 Framing options depend on PICC Type, frame format, bit rate and communication direction. An overview is given in Annex G.

As an example the sequence for an activation of frame with error correction in both directions with

— no SYNC suppression in both directions,

— no SOF and EOF suppression in both directions,

— start bit and stop bit suppression in both directions, and

with a PICC indicating to support

— standard and frame with error correction in both directions independent of each direction,

— SYNC suppression in both directions,

— SOF and EOF suppression in both directions, and

— start bit and stop bit suppression in both directions is illustrated in Figure 38.

An overview on allowed suppressions of framing options is given in Annex G.

Step	PCD		PICC
1	S(PARAMETERS)('A0 02 A5 00' CRC)	⟶	
2		⟵	S(PARAMETERS) ('A0 0E' 'A6 0C' '80 01 03' '81 01 03' '82 01 07' '83 01 07' CRC)
3	S(PARAMETERS) ('A0 0E' 'A7 0C' '84 01 02' '85 01 02' '86 01 01' '87 01 01' CRC)	⟶	
4		⟵	S(PARAMETERS)('A0 02 A8 00' CRC)

Figure 38 — Frame activation example

NOTE 4 For Type A PICCs, tags '82' and '86' are omitted in certain cases (see table footnote b in Table 6) and tags '83' and '87' are always omitted (see table footnote c in Table 6).

Annex A
(informative)

Multi-Activation example

The following table describes an example of the usage of Multi-Activation for three PICCs.

Table A.1 — Multi-Activation

PCD Action	Status PICC 1	Status PICC 2	Status PICC 3
Power On field			
Three PICCs enter the field.	IDLE	IDLE	IDLE
Activate PICC with CID = 1	ACTIVE(1)	IDLE	IDLE
Any data transmission with CID = 1	ACTIVE(1)	IDLE	IDLE
…			
Activate PICC with CID = 2	ACTIVE(1)	ACTIVE(2)	IDLE
Any data transmission with CID = 1,2	ACTIVE(1)	ACTIVE(2)	IDLE
…			
Activate PICC with CID = 3	ACTIVE(1)	ACTIVE(2)	ACTIVE(3)
Any data transmission with CID = 1,2,3	ACTIVE(1)	ACTIVE(2)	ACTIVE(3)
…			
S(DESELECT) Command with CID = 3	ACTIVE(1)	ACTIVE(2)	HALT
S(DESELECT) Command with CID = 2	ACTIVE(1)	HALT	HALT
S(DESELECT) Command with CID = 1	HALT	HALT	HALT
…			

NOTE The number n in ACTIVE(n) represents the CID.

Annex B
(informative)

Protocol scenarios

B.1 General

This Annex gives some scenarios for an error-free operation, as well as for error handling. These scenarios may be used to build test cases for compliance tests.

B.2 Notation

Any block	===>	correctly received
Any block	=≠=>	erroneously received
Any block	= =>	nothing received (FWT time-out)
Separator line	—— ——	end of the smallest protocol operation
$I(1)_x$	I-block with chaining bit set and block number x	
$I(0)_x$	I-block with chaining bit not set (last block of chain) and block number x	
$R(ACK)_x$	R-block indicating a positive acknowledge	
$R(NAK)_x$	R-block indicating a negative acknowledge	
S(...)	S-block	

The block numbering in a scenario always starts with the PCD's current block number for the destination PICC. For ease of presentation, scenarios start after the PICC activation sequence and hence, the current block numbers start with 0 for the PCD and with one for the PICC.

B.3 Error-free operation

B.3.1 Exchange of I-blocks

Scenario 1 Exchange of I-blocks

	Comment	Block No. (0)	PCD		PICC	Block No. (1)	Comment
1.	rule 1		$I(0)_0$	===>		0	rule D
2.	rule B	1		<===	$I(0)_0$		rule 10
3.			$I(0)_1$	===>		1	rule D
4.	rule B	0		<===	$I(0)_1$		rule 10

B.3.2 Request for waiting time extension

Scenario 2 Waiting time extension

	Comment	Block No. (0)	PCD		PICC	Block No. (1)	Comment
1.	rule 1		$I(0)_0$	===>		0	rule D
2.				<===	S(WTX) request		rule 9
3.	rule 3		S(WTX) response	===>			
4.	rule B	1		<===	$I(0)_0$		rule 10
5.			$I(0)_1$	===>		1	rule D
6.	rule B	0		<===	$I(0)_1$		rule 10

B.3.3 DESELECT

Scenario 3 DESELECT

	Comment	Block No. (0)	PCD		PICC	Block No. (1)	Comment
1.	rule 1		$I(0)_0$	===>		0	rule D
2.	rule B	1		<===	$I(0)_0$		rule 10
3.			S(DESELECT) request	===>			
4.				<===	S(DESELECT) response		rule 3

B.3.4 Chaining

Scenario 4 PCD uses chaining

	Comment	Block No. (0)	PCD		PICC	Block No. (1)	Comment
1.	rule 1		$I(1)_0$	===>		0	rule D
2.	rule B	1		<===	$R(ACK)_0$		rule 2
3.	rule 7		$I(0)_1$	===>		1	rule D
4.	rule B	0		<===	$I(0)_1$		rule 10
5.			$I(0)_0$	===>		0	rule D
6.	rule B	1		<===	$I(0)_0$		rule 10

Scenario 5 PICC uses chaining

	Comment	Block No. (0)	PCD		PICC	Block No. (1)	Comment
1.	rule 1		$I(0)_0$	===>		0	rule D
2.	rule B	1		<===	$I(1)_0$		rule 10
3.	rule 2		$R(ACK)_1$	===>		1	rule E
4.	rule B	0		<===	$I(0)_1$		rule 13
5.			$I(0)_0$	===>		0	rule D
6.	rule B	1		<===	$I(0)_0$		rule 10

B.3.5 PICC Presence check

Scenario 6 PICC presence check using method 1

	Comment	Block No. (0)	PCD		PICC	Block No. (1)	Comment
1.	rule 1 and method 1		$I(0)_0$	===>		0	rule D
2.	rule B	1		<===	$I(0)_0$		rule 10 note

Scenario 7 PICC presence check using method 2 (before the first I-block exchange)

	Comment	Block No. (0)	PCD		PICC	Block No. (1)	Comment
1.	rule 1 and method 2		$R(NAK)_0$	===>			rule E note
2.		no change		<===	$R(ACK)_1$		rule 12
3.	rule 6 note and method 2	$R(NAK)_0$	===>				rule E note
4.	rule 6 note	no change		<===	$R(ACK)_1$		rule 12
5.			$I(0)_0$	===>		0	rule D
6.	rule B	1		<===	$I(0)_0$		rule 10

Scenario 8 PICC presence check using method 2-a (after the first I-block exchange)

	Comment	Block No. (0)	PCD		PICC	Block No. (1)	Comment
1.	rule 1		$I(0)_0$	===>		0	rule D
2.	rule B	1		<===	$I(0)_0$		rule 10
3.	method 2-a		$R(NAK)_1$	===>			rule E note
4.	rule 6 note	no change		<===	$R(ACK)_0$		rule 12
5.			$I(0)_1$	===>		1	rule D
6.	rule B	0		<===	$I(0)_1$		rule 10

Scenario 9 PICC presence check using method 2-b (after the first I-block exchange)

	Comment	Block No. (0)	PCD		PICC	Block No. (1)	Comment
1.	rule 1		$I(0)_0$	===>		0	rule D
2.	rule B	1		<===	$I(0)_0$		rule 10
3.	method 2-b	0	$R(NAK)_0$	===>			
4.	rule B	1		<===	$I(0)_0$		rule 11
5.			$I(0)_1$	===>		1	rule D
6.	rule B	0		<===	$I(0)_1$		rule 10

B.3.6 Exchange of additional parameters

Scenario 25 Exchange of additional parameters

	Comment	Block No. (0)	PCD		PICC	Block No. (1)	Comment
1.	rule 1		$I(0)_0$	===>		0	rule D
2.	rule B	1		<===	$I(0)_0$		rule 10
3.		S(PARAMETERS) request		===>			
4.		1		<===	S(PARAMETERS) response		rule 3
5.			$I(0)_1$	===>		1	rule D
6.	rule B	0		<===	$I(0)_1$		rule 10

B.4 Error handling

B.4.1 Exchange of I-blocks

Scenario 10 Start of protocol

	Comment	Block No. (0)	PCD		PICC	Block No. (1)	Comment
1.	rule 1		$I(0)_0$	===>		0	rule D
2.	rule B	1		<===	$I(0)_0$		rule 10
3.		S(PARAMETERS) request		===>			
4.		1		<===	S(PARAMETERS) response		rule 3
5.			$I(0)_1$	===>		1	rule D
6.	rule B	0		<===	$I(0)_1$		rule 10

Scenario 11 Exchange of I-blocks

	Comment	Block No. (0)	PCD		PICC	Block No. (1)	Comment
1.	rule 1		$I(0)_0$	===>		0	rule D
2.	rule B	1		<===	$I(0)_0$		rule 10
3.			$I(0)_1$	=≠=>			
4.	time-out			<= =	-		
5.	rule 4		$R(NAK)_1$	===>			
6.		no change		<===	$R(ACK)_0$		rule 12
7.	rule 6		$I(0)_1$	===>		1	rule D
8.	rule B	0		<===	$I(0)_1$		rule 10
9.			$I(0)_0$	===>		0	rule D
10.	rule B	1		<===	$I(0)_0$		rule 10

Scenario 12 Exchange of I-blocks

	Comment	Block No. (0)	PCD		PICC	Block No. (1)	Comment
1.	rule 1		$I(0)_0$	===>		0	rule D
2.				<=≠=	$I(0)_0$		rule 10
3.	rule 4		$R(NAK)_0$	===>			
4.	rule B	1		<===	$I(0)_0$		rule 11
5.			$I(0)_1$	===>		1	rule D
6.	rule B	0		<===	$I(0)_1$		rule 10

Scenario 13 Exchange of I-blocks

	Comment	Block No. (0)	PCD		PICC	Block No. (1)	Comment
1.	rule 1		$I(0)_0$	===>		0	rule D
2.				<=≠=	$I(0)_0$		rule 10
3.	rule 4		$R(NAK)_0$	=≠=>			
4.	time-out			<= =	-		
5.	rule 4		$R(NAK)_0$	===>			
6.	rule B	1		<===	$I(0)_0$		rule 11
7.			$I(0)_1$	===>		1	rule D
8.	rule B	0		<===	$I(0)_1$		rule 10

Scenario 26 Exchange of I-blocks

	Comment	Block No. (0)	PCD		PICC	Block No. (1)	Comment
1.	rule 1		$I(0)_0$	===>		0	rule D
2.	rule B	1		<===	$I(0)_0$		rule 10
3.		S(PARAMETERS) request		=≠=>			
4.	time-out			<===	-		
5.	rule 8	S(PARAMETERS) request		===>			
6.				<===	S(PARAMETERS) response		rule 3
7.			$I(0)_1$	===>		1	rule D
8.	rule B	0		<===	$I(0)_1$		rule 10

B.4.2 Request for waiting time extension

Scenario 14 Request for waiting time extension

	Comment	Block No. (0)	PCD		PICC	Block No. (1)	Comment
1.	rule 1		$I(0)_0$	===>		0	rule D
2.				<=≠=	S(WTX) request		rule 9
3.	rule 4		$R(NAK)_0$	===>			
4.				<===	S(WTX) request		rule 11
5.	rule 3		S(WTX) response	===>			
6.	rule B	1		<===	$I(0)_0$		rule 10
7.			$I(0)_1$	===>		1	rule D
8.	rule B	0		<===	$I(0)_1$		rule 10

Scenario 15 Request for waiting time extension

	Comment	Block No. (0)	PCD		PICC	Block No. (1)	Comment
1.	rule 1		$I(0)_0$	===>		0	rule D
2.				<=≠=	S(WTX) request		rule 9
3.	rule 4		$R(NAK)_0$	=≠=>			
4.	time-out			<= =	-		
5.	rule 4		$R(NAK)_0$	===>			
6.				<===	S(WTX) request		rule 11
7.	rule 3		S(WTX) response	===>			
8.	rule B	1		<===	$I(0)_0$		rule 10
9.			$I(0)_1$	===>		1	rule D
10.	rule B	0		<===	$I(0)_1$		rule 10

Scenario 16 Request for waiting time extension

	Comment	Block No. (0)	PCD		PICC	Block No. (1)	Comment
1.	rule 1		$I(0)_0$	===>		0	rule D
2.				<===	S(WTX) request		rule 9
3.	rule 3		S(WTX) response	=≠=>			
4.	time-out			<= =	-		
5.	rule 4		$R(NAK)_0$	===>			
6.				<===	S(WTX) request		rule 11
7.	rule 3		S(WTX) response	===>			
8.	rule B	1		<===	$I(0)_0$		rule 10
9.			$I(0)_1$	===>		1	rule D
10.	rule B	0		<===	$I(0)_1$		rule 10

Scenario 17 Request for waiting time extension

	Comment	Block No. (0)	PCD		PICC	Block No. (1)	Comment
1.	rule 1		$I(0)_0$	===>		0	rule D
2.				<===	S(WTX) request		rule 9
3.	rule 3		S(WTX) response	===>			
4.				<=≠=	$I(0)_0$		rule 10
5.	rule 4		$R(NAK)_0$	===>			
6.	rule B	1		<===	$I(0)_0$		rule 11
7.			$I(0)_1$	===>		1	rule D
8.	rule B	0		<===	$I(0)_1$		rule 10

Scenario 18 Request for waiting time extension

	Comment	Block No. (0)	PCD		PICC	Block No. (1)	Comment
1.	rule 1		$I(0)_0$	===>		0	rule D
2.				<===	S(WTX) request		rule 9
3.	rule 3		S(WTX) response	===>			
4.				<=≠=	$I(0)_0$		rule 10
5.	rule 4		$R(NAK)_0$	=≠=>			
6.	time-out			<= =	-		
7.	rule 4		$R(NAK)_0$	===>			
8.	rule B	1		<===	$I(0)_0$		rule 11
9.			$I(0)_1$	===>		1	rule D
10.	rule B	0		<===	$I(0)_1$		rule 10

B.4.3 DESELECT

Scenario 19 DESELECT

	Comment	Block No. (0)	PCD		PICC	Block No. (1)	Comment
1.	rule 1		$I(0)_0$	===>		0	rule D
2.	rule B			<===	$I(0)_0$		rule 10
3.			S(DESELECT) request	=≠=>			
4.	time-out			<= =	-		
5.	rule 8		S(DESELECT) request	===>			
6.				<===	S(DESELECT) response		rule 3