ISO/IEC JTC 1/SC 17 N 1689

Date: 2000-03-10

Draft ISO/IEC FCD 14443-4

ISO/IEC JTC 1/SC 17/WG 8

Editor: Hauke Meyn
hauke.meyn@philips.com

Secretariat: DIN

Identification cards — Contactless integrated circuit(s) cards — Proximity cards — Part 4: Transmission protocol

Cartes d'identification — Cartes à circuit(s) intégrés sans contacts — Cartes de proximité — Partie 4: Protocole de transmission

Document type: International Standard
Document subtype: Not applicable
Document stage: (30) Final Committee Draft
Document language: E

G:\ISO_IEC\CD14443-4V4\FCD14443-4.doc ISOSTD ISO Template Version 3.3 1998-01-12

Contents

Foreword

ISO (the International Organisation for Standardisation) and IEC (the International Electrotechnical Commission) form the specialised system for worldwide standardisation. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organisation to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organisations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 3.

In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1. 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.

International Standard ISO/IEC 14443-4 was prepared by Joint Technical Committee ISO/IEC JTC1, Information technology, Subcommittee SC17, *Identification cards and related devices.*

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

— *Part 1: Physical characteristics*

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

— *Part 3: Initialization and anticollision*

— *Part 4: Transmission protocol*

The annexes A, B and C of this part of ISO/IEC 14443 are for information only.

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 the mapping of the application protocol data units can be the same as described in ISO/IEC 7816-4 for the protocol T=1.

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 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 shall be used in conjunction with other parts of ISO/IEC 14443 and is applicable to proximity cards of Type A and Type B.

2 Normative reference(s)

The following standards contain provisions, which, through reference in this text, constitute provisions of this part of ISO/IEC 14443. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this part of ISO/IEC 14443 are encouraged to investigate the possibility of applying the most recent valid International Standards.

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

ISO/IEC 7816-4, Identification cards – Integrated circuit(s) cards with contacts – Part 4: Interindustry commands for interchange.

3 Term(s) and definition(s)

3.1
bit duration
The bit duration is defined as one elementary time unit (etu). The etu is calculated by the following formula :

1 etu = 128 / (*D* x *fc*)

The initial value of the divisor *D* shall be 1. Therefore the resulting initial etu shall be :

1 etu = 128 / *fc*

The carrier frequency *fc* is defined in ISO/IEC 14443-2.

3.2
block
A special type of frame, which contains a valid protocol data format. A valid protocol data format includes I-blocks, R-blocks or S-blocks.

3.3
invalid block
A type of frame, which contains an invalid protocol format. A time-out, when no frame has been received, is not interpreted as an invalid block.

3.4
frame
As defined in ISO/IEC 14443-3. The PICC type A uses the standard frame defined for type A and the PICC type B uses the frame defined for type B.

4 Symbols (and abbreviated terms)

ACK	positive ACKnowledgement
ATS	Answer To Select
ATQB	Answer To reQuest for PICC type B
CID	Card IDentifier
CRC	Cyclic Redundancy Check, as defined for each PICC type in ISO/IEC 14443-3
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
HLTA	HALT command for PICC Type A
I-block	Information block
INF	INformation Field
NAD	Node ADdress
NAK	Negative AcKnowledgement
OSI	Open Systems Interconnection
PCB	Protocol Control Byte

PCD	Proximity Coupling Device
PICC	Proximity Card
PPS	Protocol and Parameter Selection
PPSS	Protocol and Parameter Selection Start
PPS0	Protocol and Parameter Selection 0
PPS1	Protocol and Parameter Selection 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
RFU	Reserved for Future Use
S-block	Supervisory block
SAK	Select AcKnowledge
SFGI	Start-up Frame Guard time Integer
SFGT	Start-up Frame Guard Time
WUPA	Wake-Up command for PICC Type A
WTX	Waiting Time eXtension
WTXM	Waiting Time eXtension Multiplier

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).
- At the beginning the SAK byte shall be checked for availability of an ATS. The SAK 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 no ATS is available.
- The RATS may be sent by the PCD as next command after receiving the SAK if an ATS is available.
- 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.

Figure 1 shows the PCD activation sequence for a PICC Type A .

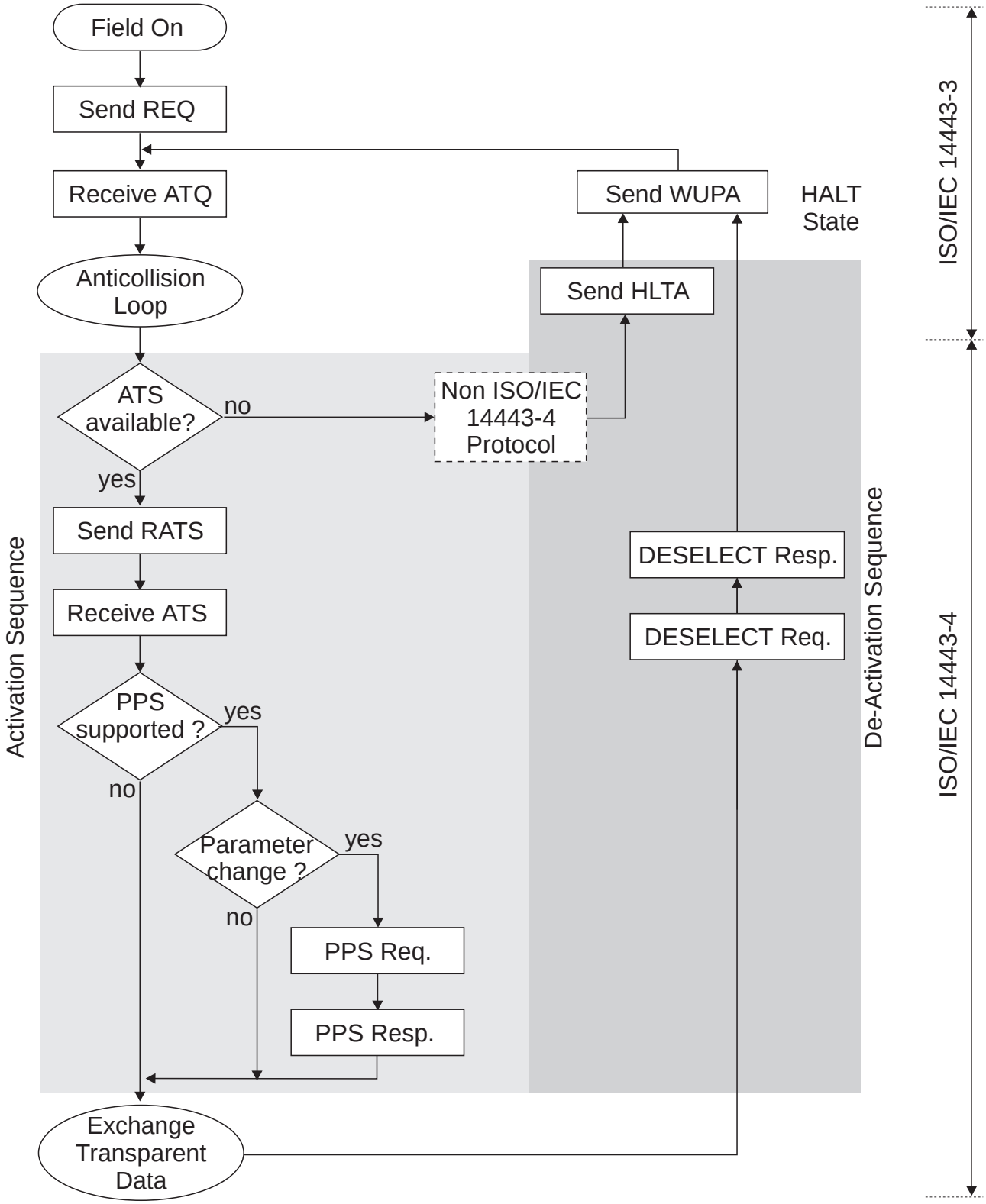

Figure 1 — Activation of a PICC Type A by a PCD

5.1 Request for answer to select

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

'E0'	Parameter	CRC
1 byte	1 byte	2 bytes

Figure 2 — Request for answer to select

The parameter byte consists of two parts (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.
- 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.

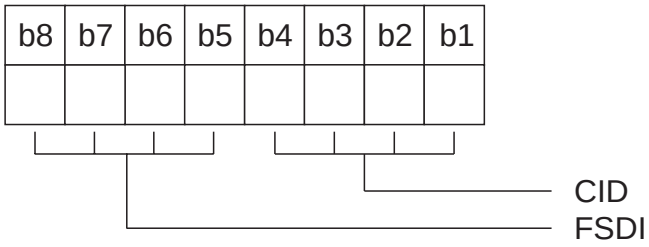

Figure 3 — Coding of RATS parameter byte

Table 1 — FSDI to FSD conversion

FSDI	'0'	'1'	'2'	'3'	'4'	'5'	'6'	'7'	'8'	'9'-'F'
FSD	16	24	32	40	48	64	96	128	256	RFU >256

5.2 Answer to select

This clause defines the ATS with all its available fields (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.

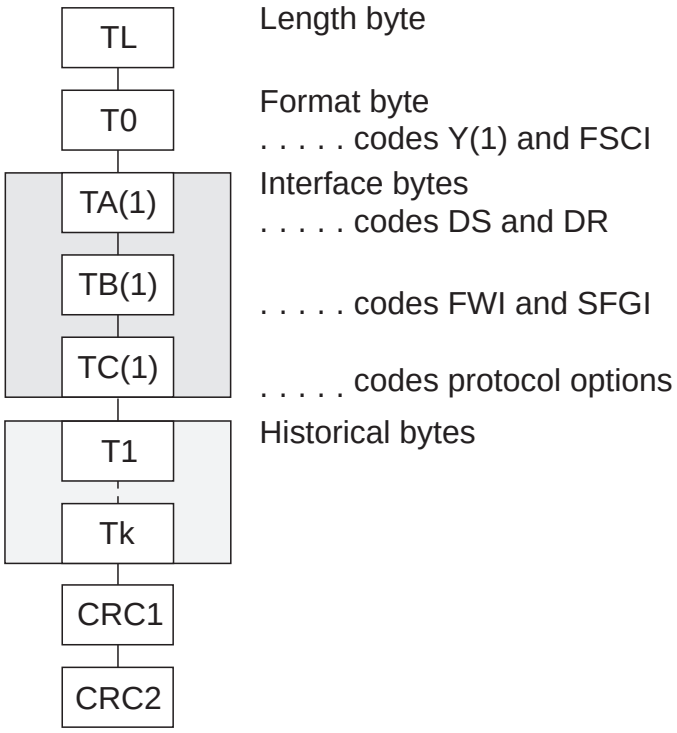

Figure 4 — Structure of the ATS

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) and
- 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 (Figure 5) :

- The most significant bit b8 shall be set to 0. The value 1 is RFU.
- The bits b7 to b5 indicate the presence of subsequent interface bytes TA(1), TB(1) and TC(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.

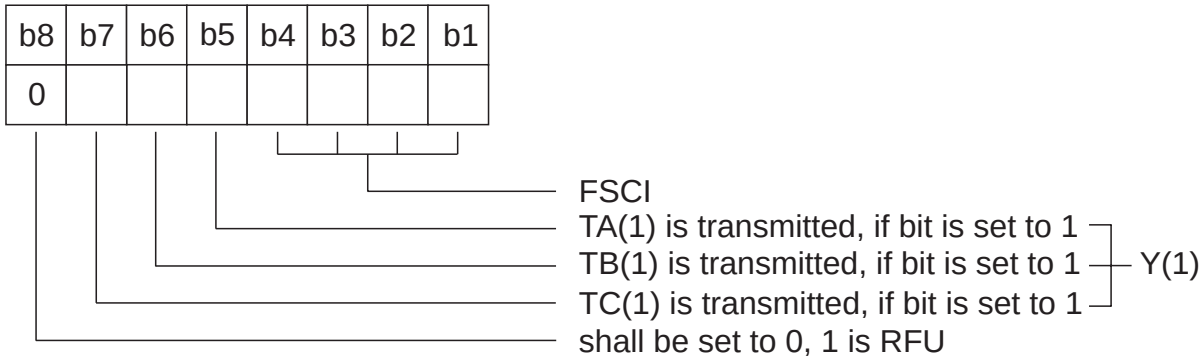

Figure 5 — Coding of format byte

5.2.4 Interface byte TA(1)

The interface byte TA(1) consists of four parts (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 is 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.

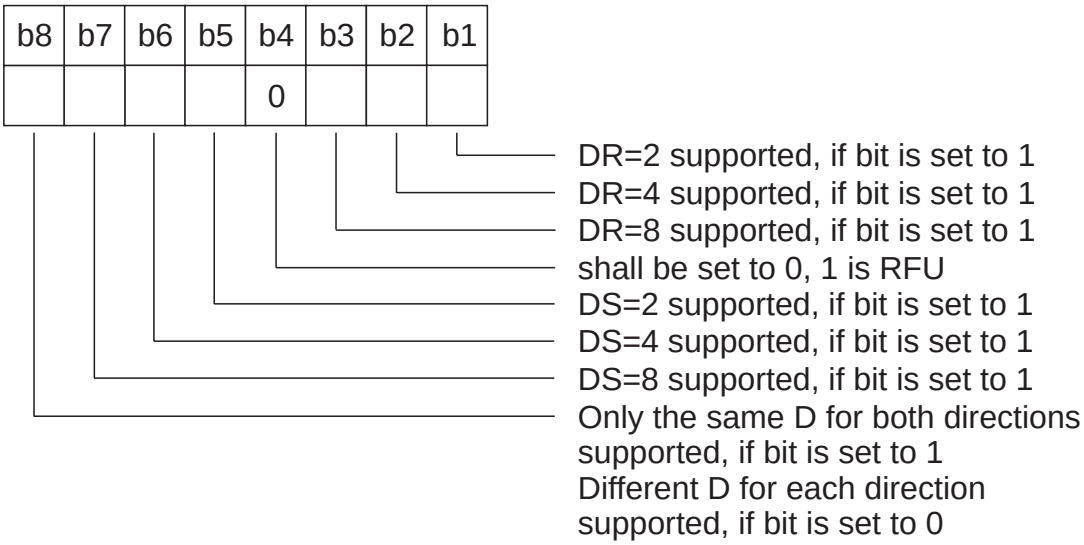

Figure 6 — Coding of interface byte TA(1)

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

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 (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.

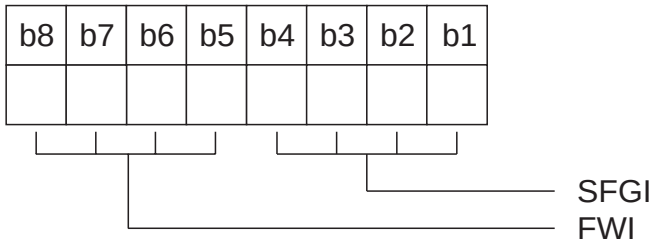

Figure 7 — Coding of interface byte TA(1)

SFGT is calculated by the following formula :

SFGT = (256 x 16 / *fc*) x 2^{SFGI}

$SFGT_{MIN}$ = minimum value as defined in ISO/IEC 14443-3

$SFGT_{DEFAULT}$ = minimum value as defined in ISO/IEC 14443-3

$SFGT_{MAX}$ = ~4949 ms

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 (Figure 8) :

- The most significant bits b8 to b3 are (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.

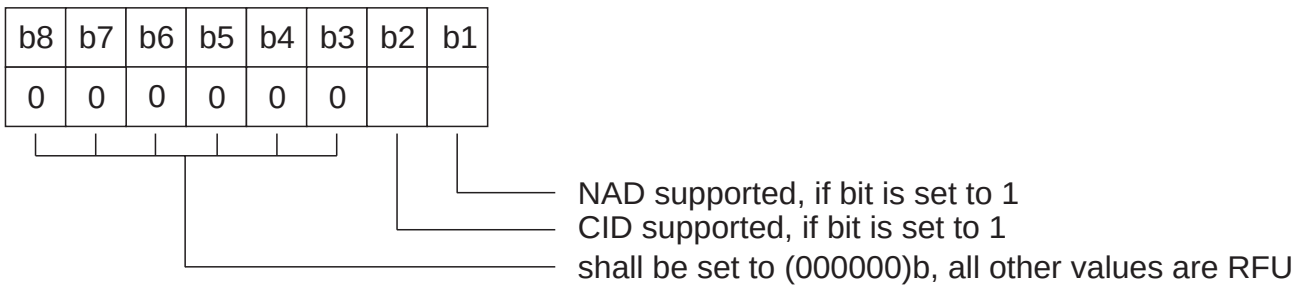

Figure 8 — Coding of interface byte TC(1)

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

The PPS request contains the start byte that is followed by a format byte and one parameter byte (Figure 9).

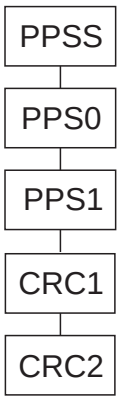

Figure 9 — Protocol and parameter selection request

5.3.1 Protocol and parameter selection start

The PPSS consists of two parts (Figure 10) :

- The most significant half byte b8 to b5 is equal to ‘D’ 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.

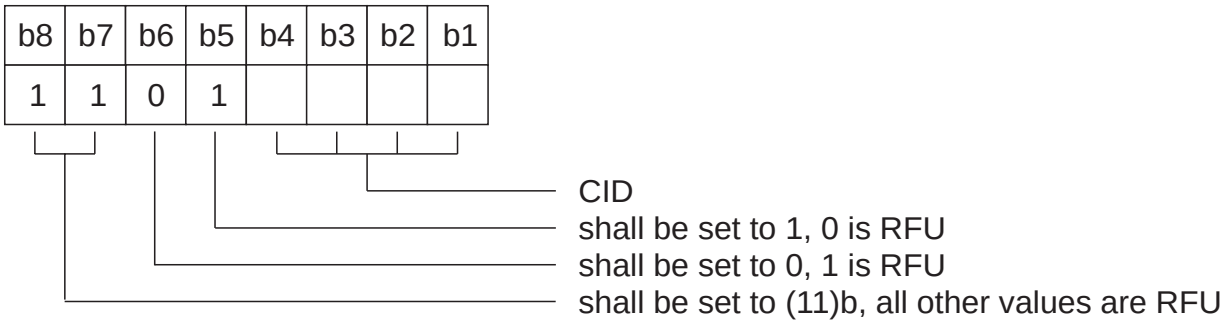

Figure 10 Coding of PPSS

5.3.2 Protocol and parameter selection 0

The PPS0 indicates the presence of the optional byte PPS1 as defined below.

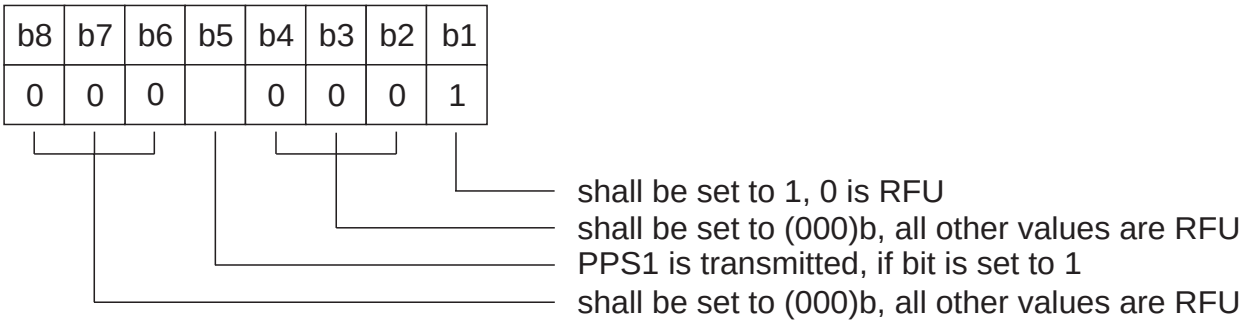

Figure 11 — Coding of PPS0

5.3.3 Protocol and parameter selection 1

The PPS1 consists of three parts (Figure 12) :

- The most significant half byte b8 to b5 is (0000)b and all other values are RFU.
- The bits b4, b3 are called DSI and code the selected divisor Integer from PICC to PCD.
- The bits b2, b1 are called DRI and code the selected divisor Integer from PCD to PICC.

Figure 12 — Coding of PPS1

For the definition of possible DS and DR, see 5.2.4.

The coding of *D* is given in Table 2.

Table 2 — DI to *D* conversion

DI	(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 (Figure 13). It contains only the PPSS as defined in **Error! Reference source not found.**.

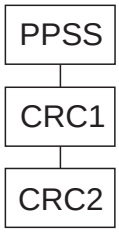

Figure 13 — Protocol and parameter selection response

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 65536/*fc* (~4833 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.

5.6.1.2 PICC rules

When the PICC has been selected with the last command and

1. receives a valid RATS, the PICC shall

- send back its ATS and
- disable the RATS (do not longer respond to received RATS).

2. receives any other block valid or invalid, except a HLTA command, the PICC shall

- ignore the block and
- remain in receive mode.

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

1. received a valid PPS request, the PICC shall

- send the PPS response,
- disable the PPS request (do not longer respond to received PPS requests) and
- activate the received parameter.

2. received an invalid block, the PICC shall

- disable the PPS request (do not longer respond to received PPS requests) and
- remain in receive mode.

3. received a valid block, except a PPS request, the PICC shall

- disable the PPS request (do not longer respond to received PPS requests) and
- 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

1. received an ATS indicating CID is supported, the PCD shall

- send blocks containing CID=*n* to this PICC and
- not use the CID=*n* for further RATS while this PICC is in ACTIVE state.

2. 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.

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

1. received an ATS indicating CID is supported, the PCD may

- send blocks containing CID equal to 0 to this PICC and
- not activate any other PICC while this PICC is in ACTIVE state.

2. 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 of Type B is described in ISO/IEC 14443-3.

7 Half-duplex block transmission protocol

This clause defines the structure of the half-duplex block transmission protocol featuring the special needs of a contactless environment.

This protocol uses the frame format as defined in ISO/IEC 14443-3. This clause covers the frame structure of

- data blocks

and the organization of

- data transmission control such as flow control, block chaining and error recovery and
- specific interface control.

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 :

- 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.

7.1 Block format

The next figure describes the structure of a block, which consists of a prologue field (mandatory), an information field (optional) and an epilogue field (mandatory).

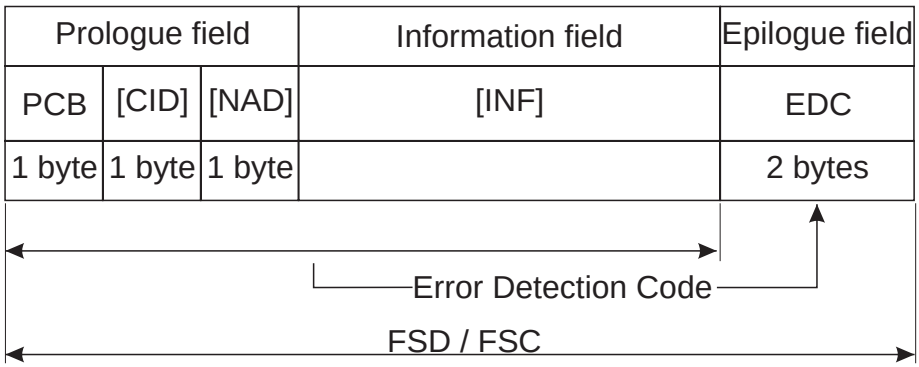

Figure 14 — Block format

7.1.1 Prologue field

This field is mandatory and consists of up to three bytes :

- Protocol Control Byte (mandatory),
- Card IDentifier (optional),

- Node ADdress (optional).

7.1.1.1 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. Two different types of S-blocks are defined :

    1. Waiting time extension containing a 1 byte long INF field and
    2. DESELECT containing no INF field.

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.

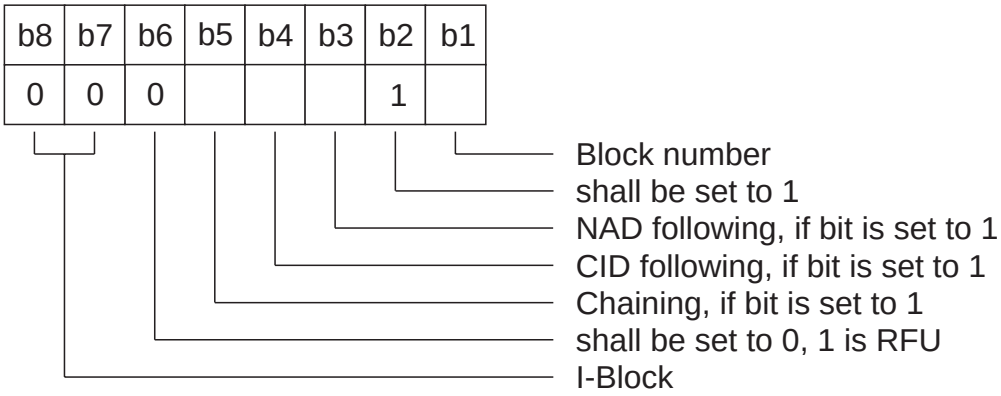

Figure 15 — Coding of I-block PCB

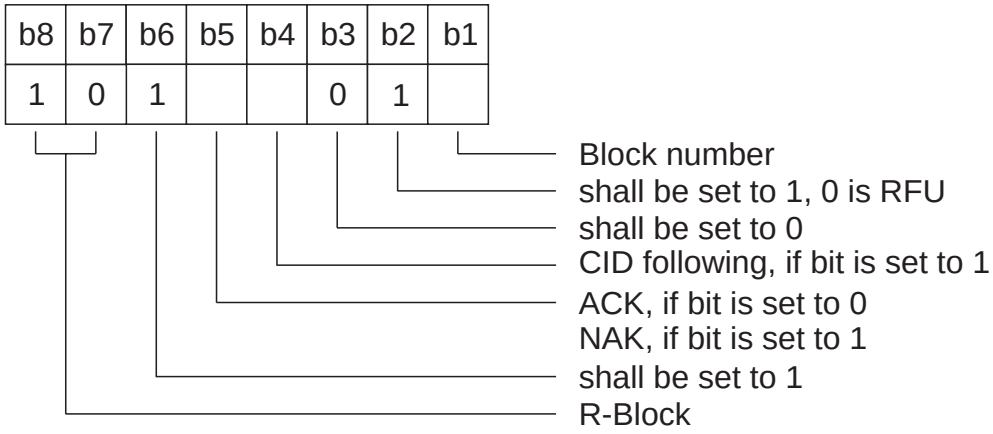

Figure 16 — Coding of R-block PCB

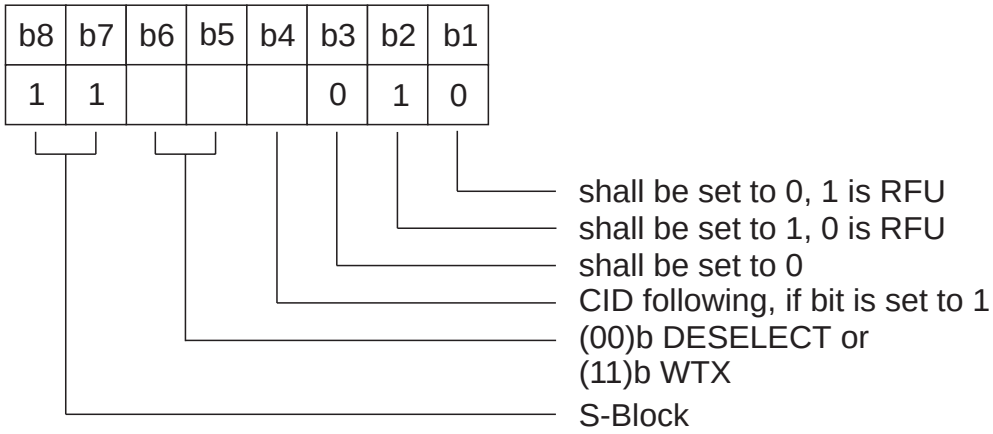

Figure 17 — Coding of S-block PCB

7.1.1.2 Card identifier field

The CID field is used to identify a specific PICC.

The CID field consists of three parts (Figure 18) :

- The most significant bits b8, b7 are used for a power level indication from PICC to PCD. These bits shall be set to 0 for PCD to PICC communication.

- The bits b6 and b5 are set to (00)b and all other values are RFU.
- The bits b4 to b1 code the CID.

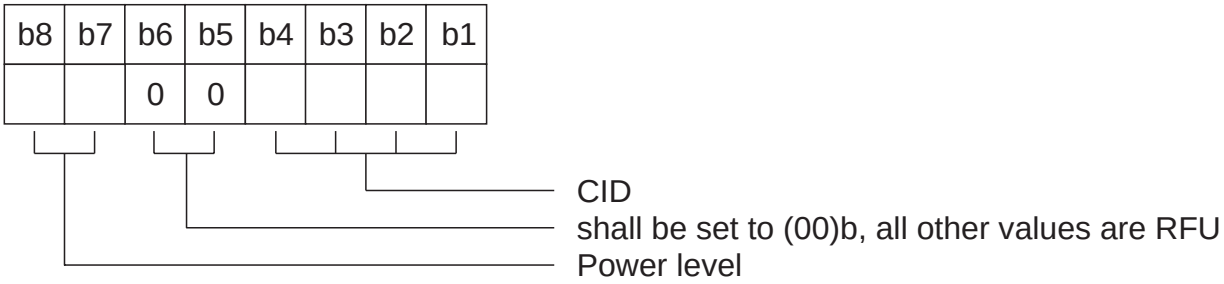

Figure 18 — Coding of card identifier

For a definition of the power level indication, see 7.3.

For the coding of the CID see 5.1 for Type A and see 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,
- ignore blocks containing other CIDs and
- in case its CID is 0, respond also to blocks containing no CID by using no CID.

7.1.1.3 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 set to 0. All other values are RFU.

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

1. The NAD field shall only be used for I-blocks.
2. When the PCD uses the NAD, the PICC shall also use the NAD.
3. During chaining the NAD is only transmitted in the first block of chain.
4. The PCD shall never use the NAD to address different PICCs (The CID shall be used to address different PICCs).
5. When the PICC does not support the NAD, it shall ignore any block containing the NAD.

7.1.2 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.

7.1.3 Epilogue field

This field contains the EDC of the transmitted block. The EDC is a CRC as defined in ISO/IEC 14443-3.

7.2 Frame waiting time

The FWT defines the maximum time for a PICC to start its response frame after the end of a PCD frame (Figure 19).

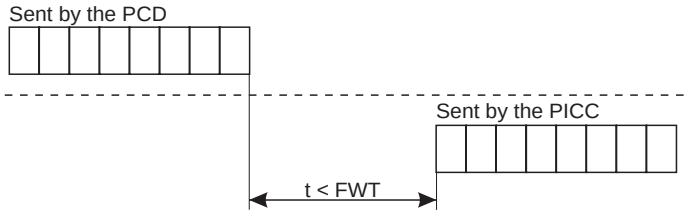

Figure 19 — Frame waiting time

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

FWT is calculated by the following formula :

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

FWI is coded in the range from 0 to 14. The value of 15 is RFU. The default value of FWI is 4.

FWT_{MIN} = ~302 µs

$FWT_{DEFAULT}$ = ~4833 µs

FWT_{MAX} = ~4949 ms

The FWT shall be used to detect a transmission error or an unresponsive PICC. The PCD gets back the right to send if the start of a response from the PICC is not received within FWT.

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 (Figure 20) :

- The most significant bits b8, b7 code a power level indication. For a definition of the power level indication, see 7.3.
- 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.

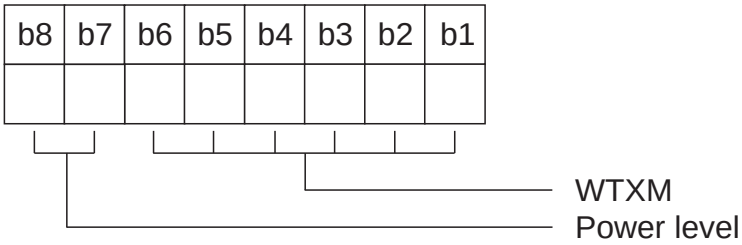

Figure 20 — Coding of INF field of an S(WTX) request

The PCD shall acknowledge by sending an S(WTX) response containing also a 1 byte long INF field that consists of two parts (Figure 21) and contains the same WTXM as received in the request :

- The most significant bits b8, b7 are (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.

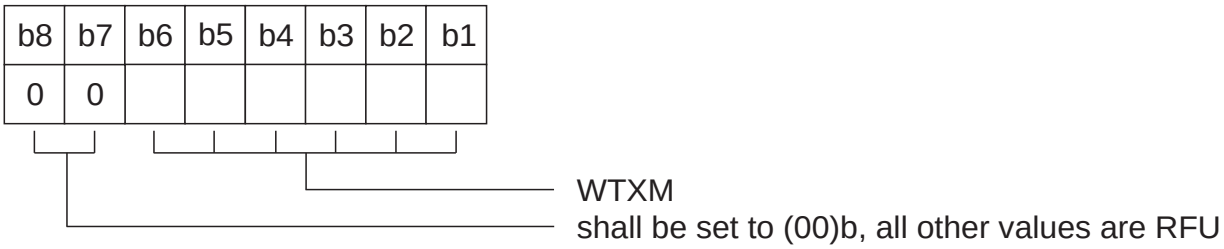

Figure 21 — Coding of INF field of an S(WTX) response

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

FWT_{TEMP} = FWT x WTXM.

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

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 as defined in 5.2.5.

7.3 Power level indication

The power level indication is coded as shown in the table below using two bits embedded in the CID field (when present) and in the S-block returned by the PICC as described in 7.1.1.2 and 7.2.

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 by the PCD of the power level indication is optional.

7.4 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 into receive mode.

The PCD shall not initiate a new pair of command / response until the current pair of command / response has been completed.

7.4.1 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.

7.4.2 Chaining

The chaining procedure 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.

An example of chaining is given in the following figure. The string of 16 bytes is 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.

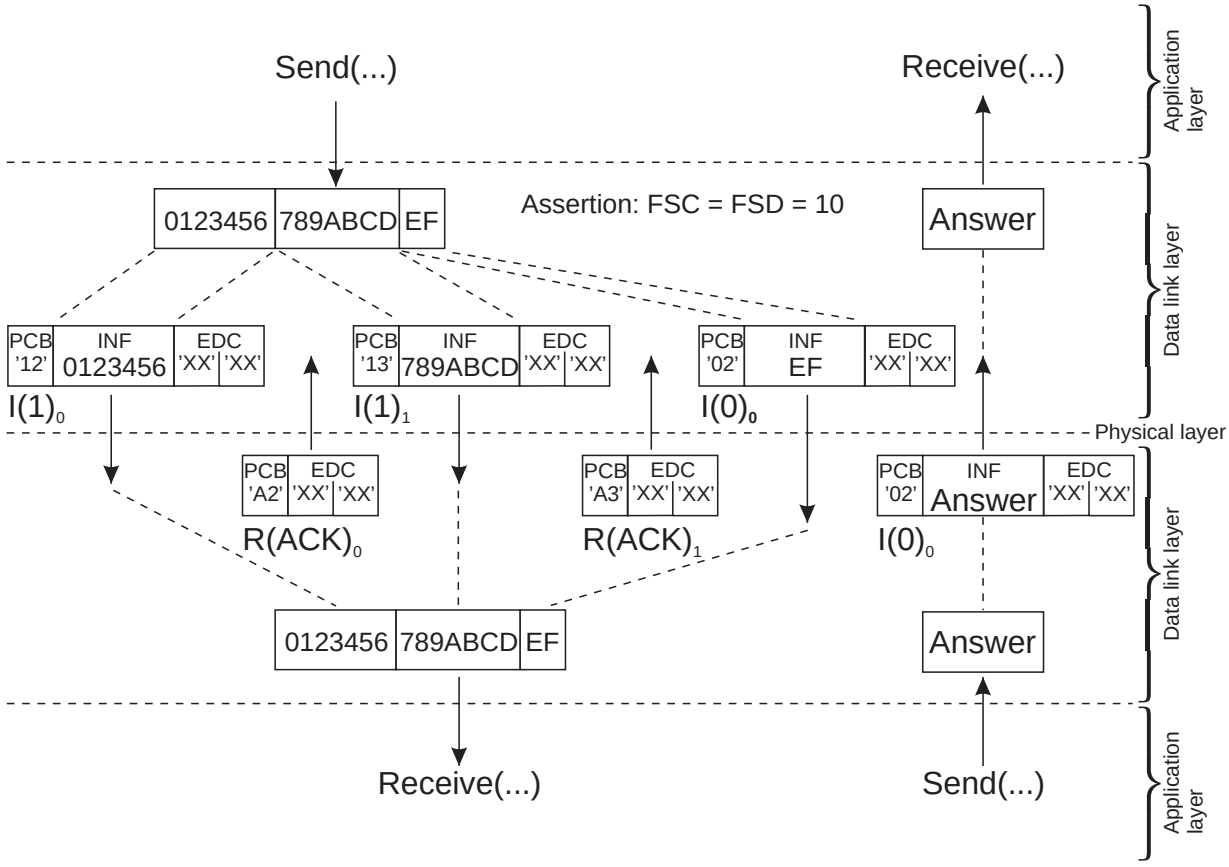

Figure 22 — Chaining

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

7.4.3 Error detection and recovery

The definitions made in this clause overrule the protocol rules defined in 7.4.5.

The following errors shall be detected by the PCD :

1. Transmission error (Frame error or EDC error) or FWT time-out

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

- Re-transmission of blocks (optional),
- Use of S(DESELECT) request,
- Ignore the PICC.

2. Protocol error (infringement of PCB coding or infringement of protocol rules)

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

   - Use of S(DESELECT) request,
   - Ignore the PICC.

The following errors shall be detected by the PICC :

- Transmission error (Frame error or EDC error),
- 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.

7.4.4 Block numbering rules

7.4.4.1 PCD rules

Rule A. The PCD block number initial value shall be 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 toggles the current block number for that PICC before optionally sending a block.

7.4.4.2 PICC rules

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

Rule D. When an I-block is received (independent of its block number), the PICC toggles its block number before sending a 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 toggles its block number before sending a block.

7.4.5 Block handling rules

7.4.5.1 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 is always followed by an S(...) response block (see 7.2 and 8).

7.4.5.2 PCD rules

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

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

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 is re-transmitted.

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

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

7.4.5.3 PICC rules

Rule 9. The PICC is allowed to sent 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.

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 is 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 is 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 is continued.

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 65536/*fc* (~4833 µs).

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.

Annex A
(informative)
Multi-Activation example

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

Table A1 — Multi-Activation

PCD Action	Status PICC 1	Status PICC 2	Status PICC 3
Power On field			
Three PICC 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
...			

Annex B
(informative)
Protocol scenarios

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.1 Notation

Any block	===>	Correctly received
Any block	=≠=>	Erroneously received
Any block	= =>	Nothing received (FWT time-out)
Separator line	____	Shows the 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 1 for the PICC.

B.2 Error-free operation

B.2.1 Exchange of I-blocks

Scenario 1 Exchange of I-blocks

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

B.2.2 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.2.3 DESELECT

Scenario 3 DESELECT

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

B.2.4 Chaining function

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 Error handling

B.3.1 Exchange of I-blocks

Scenario 6 Start of protocol

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

B.3.2 Request for waiting time extension

Scenario 10 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 11 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 12 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 13 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 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 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.3.3 DESELECT

Scenario 15 DESELECT

	Comment	Block No. (0)	PCD		PICC	Block No. (1)	Comment
11.	rule 1		$I(0)_0$	===>		0	rule D
12.	rule B			<===	$I(0)_0$		rule 10
13.			S(DESELECT) request	=≠=>			
14.	time-out			<= =	-		
15.	rule 8		S(DESELECT) request	===>			
16.				<===	S(DESELECT) response		rule 3

B.3.4 Chaining function

Scenario 16 PCD uses chaining

	Comment	Block No. (0)	PCD		PICC	Block No. (1)	Comment
1.	rule 1		$I(1)_0$	===>		0	rule D
2.				<=≠=	$R(ACK)_0$		rule 2
3.	rule 4		$R(NAK)_0$	===>			
4.	rule B	1		<===	$R(ACK)_0$		rule 11
5.	rule 7		$I(1)_1$	===>		1	rule D
6.	rule B	0		<===	$R(ACK)_1$		rule 2
7.	rule 7		$I(0)_0$	===>		0	rule D
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 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		$I(1)_1$	=≠=>			
4.	time-out			<= =	-		
5.	rule 4		$R(NAK)_1$	===>			
6.		no change		<===	$R(ACK)_0$		rule 12
7.	rule 6		$I(1)_1$	===>		1	rule D
8.	rule B	0		<===	$R(ACK)_1$		rule 2
9.	rule 7		$I(0)_0$	===>		0	rule D
10.	rule B	1		<===	$I(0)_0$		rule 10
11.			$I(0)_1$	===>		1	rule D
12.	rule B	0		<===	$I(0)_1$		rule 10

Scenario 18 PCD uses chaining

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

Scenario 19 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		$R(ACK)_1$	=≠=>			
4.	time-out			<= =	-		
5.	rule 5		$R(ACK)_1$	===>		1	rule E
6.	rule B	0		<===	$I(1)_1$		rule 13
7.	rule 2		$R(ACK)_0$	===>		0	rule E
8.	rule B	1		<===	$I(0)_0$		rule 13
9.			$I(0)_1$	===>		1	rule D
10.	rule B	0		<===	$I(0)_1$		rule 10

Scenario 20 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 D
4.				<=≠=	$I(1)_1$		rule 13
5.	rule 5		$R(ACK)_1$	===>		no change	
6.	rule B	0		<===	$I(1)_1$		rule 11
7.	rule 2		$R(ACK)_0$	===>		0	rule D
8.	rule B	1		<===	$I(0)_0$		rule 13
9.			$I(0)_1$	===>		1	rule D
10.	rule B	0		<===	$I(0)_1$		rule 10

Annex C
(informative)
Block and frame coding overview

This clause gives an overview of the different block and frame coding sent by the PCD. The type of a block respectively frame is indicated by the first byte.

Definitions made in ISO/IEC 14443-3:

REQA	(0100110)b (7 bit)
WUPA	(1010010)b (7 bit)
REQB / WUPB	(00000101)b
SLOT MARKER (Type B only)	(xxxx0101)b
Select (Type A only)	(1001xxxx)b
ATTRIB (Type B only)	(00011101)b
HLTA	(01010000)b
HLTB	(01010000)b

Definitions made in this part of ISO/IEC 14443 :

RATS	(11100000)b
PPS	(1101xxxx)b
I-block	(00xxxxxx)b (not (00xxx101)b)
R-block	(10xxxxxx)b (not (1001xxxx)b)
S-block	(11xxxxxx)b (not (1110xxxx)b and not (1101xxxx)b)

The next table describes the first byte of the defined block and frame coding.

Table C1 — Block and frame coding

Bit	PCB-I	PCB-R	PCB-S DESELECT	PCB-S WTX	REQB / WUPB	SLOT MARKER	SELECT	ATTRIB	HLTA	HLTB	RATS	PPS
b8	0	1	1		0	X	1	0	0	0	1	1
b7	0	0	1		0	X	0	0	1	1	1	1
b6	0 (RFU)	1	0	1	X	X	0	0	0	0	1	0
b5	More	Error	0	1	X	X	1	1	1	1	0	1
b4	CID	CID	CID		0	X	X	1	0	0	0	X
b3	NAD	0 (no NAD)	0 (no NAD)		1	1	X	1	0	0	0	X
b2	1	1 (RFU)	1 (RFU)		0	0	X	0	0	0	0	X
b1	block number	block number	0 (RFU)		1	1	X	1	0	0	0	X