Identification cards — Contactless integrated circuit cards — Proximity cards
Part 4: Transmission protocol
National foreword

This British Standard is the UK implementation of ISO/IEC 14443-4:2016+A1:2016. It supersedes BS ISO/IEC 14443-4:2016 which is withdrawn.

The start and finish of text introduced or altered by amendment is indicated in the text by tags. Tags indicating changes to ISO/IEC text carry the number of the ISO/IEC amendment. For example, text altered by ISO/IEC amendment 1 is indicated by [A] [].

The UK participation in its preparation was entrusted to Technical Committee IST/17, Cards and personal identification.

A list of organizations represented on this committee can be obtained on request to its secretary.

This publication does not purport to include all the necessary provisions of a contract. Users are responsible for its correct application.

© The British Standards Institution 2017.
Published by BSI Standards Limited 2017
ISBN 978 0 580 94404 8
ICS 35.240.15

Compliance with a British Standard cannot confer immunity from legal obligations.

This British Standard was published under the authority of the Standards Policy and Strategy Committee on 30 June 2016.

Amendments/corrigenda issued since publication

<table>
<thead>
<tr>
<th>Date</th>
<th>Text affected</th>
</tr>
</thead>
</table>
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
Contents

Foreword .............................................................................................................................................. v
Introduction .......................................................................................................................................... vi
1 Scope .................................................................................................................................................. 1
2 Normative references ..................................................................................................................... 1
3 Terms and definitions .................................................................................................................... 1
4 Symbols and abbreviated terms .................................................................................................... 2
5 Protocol activation of PICC Type A ............................................................................................... 4
  5.1 Request for answer to select ........................................................................................................ 5
  5.2 Answer to select .......................................................................................................................... 7
    5.2.1 Structure of the bytes .......................................................................................................... 7
    5.2.2 Length byte .......................................................................................................................... 7
    5.2.3 Format byte .......................................................................................................................... 7
    5.2.4 Interface byte TA(1) ............................................................................................................ 8
    5.2.5 Interface byte TB(1) ............................................................................................................ 9
    5.2.6 Interface byte TC(1) ............................................................................................................ 9
    5.2.7 Historical bytes ................................................................................................................... 10
  5.3 Protocol and parameter selection request ................................................................................ 10
    5.3.1 Start byte ............................................................................................................................ 10
    5.3.2 Parameter 0 ........................................................................................................................ 11
    5.3.3 Parameter 1 ........................................................................................................................ 11
  5.4 Protocol and parameter selection response ............................................................................ 11
  5.5 Activation frame waiting time ................................................................................................. 12
  5.6 Error detection and recovery ................................................................................................... 12
    5.6.1 Handling of RATS and ATS ............................................................................................... 12
    5.6.2 Handling of PPS request and PPS response ..................................................................... 12
    5.6.3 Handling of the CID during activation ............................................................................... 13
6 Protocol activation of PICC Type B ............................................................................................... 13
7 Half-duplex block transmission protocol ....................................................................................... 13
  7.1 Block format ............................................................................................................................... 14
    7.1.1 Length field ........................................................................................................................ 15
    7.1.2 Prologue field ...................................................................................................................... 15
    7.1.3 Information field ............................................................................................................... 19
    7.1.4 Epilogue field ..................................................................................................................... 19
  7.2 Frame waiting time .................................................................................................................... 19
  7.3 Frame waiting time extension ................................................................................................... 20
  7.4 Power level indication .............................................................................................................. 20
  7.5 Protocol operation .................................................................................................................... 20
    7.5.1 S(PARAMETERS) blocks ................................................................................................. 20
    7.5.2 Multi-Activation ............................................................................................................... 21
    7.5.3 Chaining ............................................................................................................................. 21
    7.5.4 Block numbering rules ....................................................................................................... 22
    7.5.5 Block handling rules ......................................................................................................... 23
    7.5.6 PICC presence check ........................................................................................................ 24
    7.5.7 Error detection and recovery ............................................................................................ 24
8 Protocol deactivation of PICC Type A and Type B ...................................................................... 25
  8.1 Deactivation frame waiting time ............................................................................................. 25
  8.2 Error detection and recovery .................................................................................................. 25
9 Activation of bit rates and framing options in the PROTOCOL state ........................................... 25
10 Frame with error correction ....................................................................................................... 29
Annex A (informative) Multi-Activation example ............................................................................. 36
Annex B (informative) Protocol scenarios ...........................................................................................................37
Annex C (informative) Block and frame coding overview ..........................................................................................46
Annex D (normative) Bit rates of 3fc/4, fc, 3fc/2 and 2fc from PCD to PICC .........................................................48
Annex E (informative) CRC_32 encoding ......................................................................................................................50
Annex F (informative) Frame with error correction ...................................................................................................52
Annex G (informative) Framing options .......................................................................................................................54
Bibliography ..................................................................................................................................................................55
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.


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


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:

\[ 1 \text{etu} = \frac{128}{(D \times fc)} \]

the initial value of the divisor \(D\) is 1, giving the initial etu as follows:

\[ 1 \text{etu} = \frac{128}{fc} \]

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.

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

cn Hamming control bit n

d vector containing 56 data bits

dn 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

H matrix needed to calculate Hamming syndrome \( \mathbf{s} \) (6 rows, 62 columns)

\( h'_{m,n} \) element in row m and column n of matrix \( H' \)

\( H' \) matrix needed to get matrix A (6 rows, 62 columns)

\( \mathbf{h}'_n \) column vector of matrix \( H' \)

HLTA HALT Command, Type A

\( I_{6 \times 6} \) 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

\( \mathbf{s} \) 6-bit vector containing Hamming syndrome

\( \mathbf{s}' \) error position code

s error position

S-block Supervisory block

SAK Select AcKnowledgement

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

\( \mathbf{y} \) 64-bit vector (\( \mathbf{y}' \) with no padding bits)

\( \mathbf{y}' \) 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:

- (xxxx)b data bit representation;
- ‘XY’ hexadecimal notation, equal to XY to the base 16.

For the purposes of this part of ISO/IEC 14443, the following general rules apply:

- A PICC or PCD sending RFU bits shall set these bits to the value indicated herein or to (0)b if no value is given.
- A PICC or PCD receiving RFU bits shall disregard the value of these bits and shall maintain and not change its function, unless explicitly stated otherwise.

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.
5.1 Request for answer to select

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

[NOTE] This PCD requirement is added for PCD’s compatibility with future PICCs when ISO/IEC further defines the behaviour for the RFU values of ‘D’ – ‘F’.

— Until the RFU values ‘D’ – ‘F’ are assigned by ISO/IEC, a PICC receiving an FSDI with a value = ‘D’ – ‘F’ should interpret it as FSDI = ‘C’ (FSD = 4096 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’.

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.

— 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;

[NOTE] Text deleted

Figure 3 — Coding of RATS parameter byte

<table>
<thead>
<tr>
<th>b8</th>
<th>b7</th>
<th>b6</th>
<th>b5</th>
<th>b4</th>
<th>b3</th>
<th>b2</th>
<th>b1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

CID
FSDI

T0 consists of three parts (see Figure 5).
contain the following optional bytes when this format byte is present.

The format byte T0 is optional and is present as soon as the length is greater than 1. The ATS can only be transmitted if the format byte is present.

The length byte TL is mandatory and specifies the length of the transmitted ATS including itself. The maximum value of TL shall not exceed FSD-2. Therefore, the maximum value of TL shall not exceed FSD-2.

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

5.2.1 Structure of the bytes

5.2.2 Length byte

5.2.3 Format byte

— historical bytes T1 to Tk.
— interface bytes TA(1), TB(1), TC(1);
Table 1 — FSDI to FSD conversion

<table>
<thead>
<tr>
<th>FSDI</th>
<th>'0'</th>
<th>'1'</th>
<th>'2'</th>
<th>'3'</th>
<th>'4'</th>
<th>'5'</th>
<th>'6'</th>
<th>'7'</th>
<th>'8'</th>
<th>'9'</th>
<th>'A'</th>
<th>'B'</th>
<th>'C'</th>
<th>'D'–'F'</th>
</tr>
</thead>
<tbody>
<tr>
<td>FSD (bytes)</td>
<td>16</td>
<td>24</td>
<td>32</td>
<td>40</td>
<td>48</td>
<td>64</td>
<td>96</td>
<td>128</td>
<td>256</td>
<td>512</td>
<td>1 024</td>
<td>2 048</td>
<td>4 096</td>
<td>RFU</td>
</tr>
</tbody>
</table>

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](image)

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).

\[ b_8 \text{ is RFU.} \]
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.

- b4 shall be set to (0)b.

- 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.\footnote{A PCD receiving $TA(1)$ with $b_4 = (1)b$ shall interpret it as $(b_8$ to $b_1) = (00000000)b$, implying only ~106 kbit/s supported in both directions. The definition of $TA(1)$ with $b_4 = (1)b$ is otherwise undefined.}

### 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 $b_8$ to $b_5$ is called $FWI$ and codes $FWT$ (see 7.2).
- The least significant half byte $b_4$ to $b_1$ 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 $TB(1)$](image)

$SFGT$ 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)

Until the RFU value 15 is assigned by ISO/IEC, a PCD receiving $SFGI = 15$ shall interpret it as $SFGI = 0$. Until the RFU value 15 is assigned by ISO/IEC, a PCD receiving $FWI = 15$ shall interpret it as $FWI = 4$. \footnote{Until the RFU value 15 is assigned by ISO/IEC, a PCD receiving $SFGI = 15$ shall interpret it as $SFGI = 0$. Until the RFU value 15 is assigned by ISO/IEC, a PCD receiving $FWI = 15$ shall 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).

- $b_8$ to $b_3$ are each RFU. \footnote{$b_8$ to $b_3$ are each RFU.}
- The bits $b_2$ and $b_1$ 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.
5.3 Protocol and parameter selection request

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

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.

BS ISO/IEC 14443-4:2016

© ISO/IEC 2016 – All rights reserved
5.3.2 Parameter 0

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

The PCD shall set (b4 to b1) = (0001)b and (b8 to b6) = (000)b. A PICC receiving (b4 to b1) <> (0001)b and/or receiving (b8 to b6) <> (000)b shall apply 5.6.2.2 (b).

<table>
<thead>
<tr>
<th>b8</th>
<th>b7</th>
<th>b6</th>
<th>b5</th>
<th>b4</th>
<th>b3</th>
<th>b2</th>
<th>b1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

shall be set to (0001)b, all other values are RFU

PPS1 is transmitted, if bit is set to (1)b

shall be set to (000)b, all other values are RFU

Figure 11 — Coding of PPS0

5.3.3 Parameter 1

PPS1 consists of three parts (see Figure 12).

— b8 to b5 shall each be (0)b; a PICC receiving any bit b8 to b5 set to (1)b shall apply 5.6.2.2 (b).
— The two-bit value field (b4, b3) is called 'DSI' and codes the selected divisor integer from PICC to PCD.
— The two-bit value field (b2, b1) is called 'DRI' and codes the selected divisor integer from PCD to PICC.

<table>
<thead>
<tr>
<th>b8</th>
<th>b7</th>
<th>b6</th>
<th>b5</th>
<th>b4</th>
<th>b3</th>
<th>b2</th>
<th>b1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

DRI

DSI

Each bit shall be set to (0)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>
<thead>
<tr>
<th>DRI, DSI</th>
<th>(00)b</th>
<th>(01)b</th>
<th>(10)b</th>
<th>(11)b</th>
</tr>
</thead>
<tbody>
<tr>
<td>D</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>8</td>
</tr>
</tbody>
</table>

Table 2 — DRI, DSI to D conversion

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. The PCD shall not change 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.

![Figure 13 — Protocol and parameter selection response](image)

### 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/fc (~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

- received a valid RATS, the PICC
  - shall send back its ATS, and
  - shall disable the RATS (stop responding to received RATS);
- received a valid HLTA, the PICC
  - shall process the command and shall enter HALT state.
- received an invalid command, an error or a RATS command 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).

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

The 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:

- Protocol operation.
- Power indication, and
- Maximum frame waiting time,
- Block format,
- Other relevant elements of the frame format are

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:

- Protocol operation.
- Power indication, and
- Maximum frame waiting time,
- Block format,
- Other relevant elements of the frame format are

14. **Block format**

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

The enhanced block format specified in Clause 10 and consists of the following:

- A four-byte epilogue field (mandatory).
- An information field (optional);
- A prologue field (mandatory);
- A length field (mandatory);

The standard block format as specified in ISO/IEC 14443-3 and consists of the following:

- A two-byte epilogue field (mandatory).
- An information field (optional);
- A prologue field (mandatory);
- A length field (mandatory);

Figure 14 — Standard block format

Figure 15 — Enhanced block format

NOTE 1 The items in brackets indicate optional requirements.

### 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.

### 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 is used to convey information for use by the application layer.
- R-block is used to convey positive or negative acknowledgements. An R-block never contains an INF field. The acknowledgement relates to the last received block.  

---

© ISO/IEC 2016 – All rights reserved

15
The S-block is 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, and
3) “PARAMETERS” containing an n-byte long INF field with n ≥ 0.

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

The PCD and PICC maximum frame size 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. The setting of (b8, b7) is used to identify its block type as defined in Table 3.

A PICC or PCD receiving (b8, b7) = (01)b shall treat it as a protocol error.

<table>
<thead>
<tr>
<th>(b8,b7)</th>
<th>Block Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>(00)b</td>
<td>I-block</td>
</tr>
<tr>
<td>(01)b</td>
<td>RFU</td>
</tr>
<tr>
<td>(10)b</td>
<td>R-block</td>
</tr>
<tr>
<td>(11)b</td>
<td>S-block</td>
</tr>
</tbody>
</table>

The coding of I-block PCB is shown in Figure 16.

A PICC or PCD receiving an I-block with b2 = (0)b shall treat it as a protocol error.

A PICC or PCD receiving an I-block with b6 = (1)b should treat it as a protocol error.

The coding of R-block PCB is shown in Figure 17.
The protocol defines three fundamental types of blocks: 

- **PCB** is used to convey the information required to control the data transmission.
- **I-block** is used to convey information for use by the application layer.
- **R-block** is used to convey positive or negative acknowledgements. An R-block never contains an INF field.

The coding of S-block PCB is shown in Figure 18.

A PICC or PCD receiving an R-block with b6 = (0)b or b3 = (1)b shall treat it as a protocol error.

A PICC or PCD receiving an R-block with b2 = (0)b should treat it as a protocol error.

The coding of S-block PCB is shown in Figure 18.

A PICC or PCD receiving S-block with b3 = (1)b shall treat it as a protocol error.

A PICC or PCD receiving S-block with b1 = (1)b should treat it as a protocol error.

A PICC or PCD receiving S-block with b2 = (0)b and (b6,b5) <> (11)b shall treat it as a protocol error.

A PICC or PCD receiving S-block with b2 = (1)b and (b6,b5) = (01)b or (10)b shall treat it as a protocol error.
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).

<table>
<thead>
<tr>
<th>b8</th>
<th>b7</th>
<th>b6</th>
<th>b5</th>
<th>b4</th>
<th>b3</th>
<th>b2</th>
<th>b1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- 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).
- (b6,b5) shall be set to (00)b.
- A PICC or PCD receiving (b6,b5) = (01)b or (10)b or (11)b shall treat it as a protocol error.  
- The bits b4 to b1 code the CID.

![Figure 19 — Coding of card identifier](image)

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 usage of NAD shall be compliant to the definition of NAD in ISO/IEC 7816-3.

A PICC or a PCD receiving a NAD not compliant to ISO/IEC 7816-3 shall treat it as 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).

\[
\text{FWT} = \left( \frac{256 \times 16}{f_c} \right) \times 2^{\text{FWI}}
\]

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 = FWT_{MIN} (~302 µs).

For FWI = 14, FWT = FWT_{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.

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

Figure 20 — Frame waiting time
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 shall interpret it as FWI = 4.

NOTE 2 This PCD requirement is added for PCD’s compatibility with future PICCs when ISO/IEC further 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).

— b8 and b7 code the power level indication (see 7.4).
— b6 to b1 codes the WTXM. The WTXM is coded in the range from 1 to 59. The values 0 and 60 to 63 are RFU.
— When receiving WTXM = 0 or WTXM = 60 – 63, the PCD shall treat it as a protocol error.

<table>
<thead>
<tr>
<th>b8</th>
<th>b7</th>
<th>b6</th>
<th>b5</th>
<th>b4</th>
<th>b3</th>
<th>b2</th>
<th>b1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

WTXM
Power level indication

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.

— b8 and b7 shall be (00)b and all other values are RFU.
— b6 to b1 codes the acknowledged WTXM value used to define a temporary FWT.
— The PICC shall treat (b8,b7) <> (00)b as protocol error.
— If PICC receives a WTXM value which does not match the WTXM sent by PICC, the PICC shall treat it as a protocol error.

<table>
<thead>
<tr>
<th>b8</th>
<th>b7</th>
<th>b6</th>
<th>b5</th>
<th>b4</th>
<th>b3</th>
<th>b2</th>
<th>b1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

WTXM
shall be set to (00)b, all other values are RFU

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

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

\[
FWT_{\text{TEMP}} = FWT \times \text{WTXM}
\]

The time FWT_{\text{TEMP}} requested by the PICC starts after the PCD has sent the S(WTX) response.
**FWT\textsubscript{MAX}** shall be used, when the formula results in a value higher than FWT\textsubscript{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).

| 00b | PICC does not support the power level indication |
| 01b | Insufficient power for full functionality |
| 10b | Sufficient power for full functionality |
| 11b | 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 $S(\text{WTX})$ 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 applicable 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 PCB coding or infringement of 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

<table>
<thead>
<tr>
<th>Tag (Hex)</th>
<th>Description</th>
<th>Length (Hex)</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>'A0'</td>
<td>S(PARAMETERS) block information</td>
<td>L</td>
<td>Function Tags Identifier (see Table 5)</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Tag (Hex)</th>
<th>Description</th>
<th>Length (Hex)</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>'A0'</td>
<td>S(PARAMETERS) block information</td>
<td>L</td>
<td></td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Tag (Hex)</th>
<th>Description</th>
<th>Length (Hex)</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>'A1'</td>
<td>Bit rates Request</td>
<td>'0'</td>
<td>—</td>
</tr>
<tr>
<td>'A2'</td>
<td>Bit rates Indication</td>
<td>L</td>
<td></td>
</tr>
<tr>
<td>'80'</td>
<td>'02'</td>
<td></td>
<td>Supported bit rates from PCD to PICC</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1st byte is specified in Figure 24</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2nd byte set to '00', other values are RFU</td>
</tr>
<tr>
<td>'81'</td>
<td>'02'</td>
<td></td>
<td>Supported bit rates from PICC to PCD</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1st byte is specified in Figure 24</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2nd byte set to '00', other values are RFU</td>
</tr>
<tr>
<td>'82'</td>
<td>'01'</td>
<td></td>
<td>Supported framing options PICC to PCD (see Figure 25)</td>
</tr>
<tr>
<td>'A3'</td>
<td>Bit rates Activation</td>
<td>L</td>
<td></td>
</tr>
<tr>
<td>'83'</td>
<td>'02'</td>
<td></td>
<td>Selected bit rate from PCD to PICC</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1st byte is specified in Figure 24</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2nd byte set to '00', other values are RFU</td>
</tr>
<tr>
<td>'84'</td>
<td>'02'</td>
<td></td>
<td>Selected bit rate from PICC to PCD</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1st byte is specified in Figure 24</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2nd byte set to '00', other values are RFU</td>
</tr>
<tr>
<td>'85'</td>
<td>'01'</td>
<td></td>
<td>Selected framing options PICC to PCD (see Figure 24)</td>
</tr>
<tr>
<td>'A4'</td>
<td>Bit rates Acknowl-</td>
<td>'0'</td>
<td>—</td>
</tr>
<tr>
<td>edgement</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

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

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.
### 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.

<table>
<thead>
<tr>
<th>Step</th>
<th>PCD</th>
<th>PICC</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>S(PARAMETERS)('A0 02 A1 00' CRC)</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>S(PARAMETERS) ('A0 0A' 'A2 08' '80 02 19 00' '81 02 49 00' CRC)</td>
</tr>
<tr>
<td>3</td>
<td>S(PARAMETERS) ('A0 0A' 'A3 08' '83 02 10 00' '84 02 40 00' CRC)</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>S(PARAMETERS)('A0 02 A4 00' CRC)</td>
</tr>
</tbody>
</table>

**Figure 27 — Frame with error correction**
10.3 Type A PCD frame format for bit rates of $\frac{fc}{8}$, $\frac{fc}{4}$ and $\frac{fc}{2}$ 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.

<table>
<thead>
<tr>
<th>SOF</th>
<th>Sync</th>
<th>Enhanced block with error correction</th>
<th>EOF</th>
</tr>
</thead>
</table>

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](image)

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 $h'_n$, with $n = 1, 2, 4, 8, 16$ and $32$, of $H'$.

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

Figure 30 — Matrix $H'$ generation

\[ H' = \begin{pmatrix} 1 & 0 & 1 & 0 & \ldots & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & \ldots & 1 & 1 & 1 \end{pmatrix} \]

Figure 31 — Matrix $H'$

\[ A = \begin{pmatrix} 1 & 1 & 0 & 1 & \ldots & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & \ldots & 1 & 1 & 1 \end{pmatrix} \]

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 $b_1$ of the first byte and $d_{56}$ is bit $b_8$ of the seventh byte of any 7-byte sub-block from enhanced block.

$$\xi = A \times \mathbf{d}$$

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 $I_6 \times 6$.

$$H = A | I_{6,6} = \begin{pmatrix} 1 & 1 & 0 & 1 & \ldots & 0 & 1 & 0 & 1 & 0 & \ldots & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & \ldots & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & \ldots & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & \ldots & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & \ldots & 0 & 0 \\ 0 & 0 & 0 & \ldots & 1 & 1 & 1 & 0 & 0 & 0 & \ldots & 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 \( s \) shall be calculated using formula in Figure 35. To get \( y \) from \( y' \), the padding bits of the received data \( y'_n \) on position 57 and 64 shall be removed. \( y'_1 \) is bit \( b_1 \) of the first received byte and \( y'_64 \) is bit \( b_8 \) of the eighth received byte of any 8-byte sub-block from enhanced block with error correction.

\[
g = H \times y
\]

Figure 35 — Syndrome calculation

The numerical interpretation \( s' \) of the syndrome \( g \) 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.
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 \( s \) shall be calculated using the formula in Figure 35. To get \( y \) from \( y' \), the padding bits of the received data \( y' \) on position 57 and 64 shall be removed. \( y'_{1} \) 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.

Figure 35 — Syndrome calculation

The numerical interpretation \( s' \) of the syndrome \( s \) 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**

<table>
<thead>
<tr>
<th>Tags (Hex)</th>
<th>Description</th>
<th>Length (Hex)</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>'A5'</td>
<td>Frame Format Request</td>
<td>0</td>
<td>—</td>
</tr>
<tr>
<td>'A6'</td>
<td>Frame Format Indication</td>
<td>L</td>
<td>Tags (Hex) Length (Hex) Value</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>'80' '01' Supported frames PCD to PICC (see Figure 36)(^a)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>'81' '01' Supported frames PICC to PCD (see Figure 36)(^b)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>'82' '01' Supported framing options PCD to PICC (see Figure 37)(^b)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>'83' '01' Supported framing options PICC to PCD (see Figure 37)(^c)</td>
</tr>
<tr>
<td>'A7'</td>
<td>Frame Format Activation</td>
<td>L</td>
<td>Tags (Hex) Length (Hex) Value</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>'84' '01' Selected frame PCD to PICC (see Figure 36)(^d)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>'85' '01' Selected frame PICC to PCD (see Figure 36)(^d)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>'86' '01' Selected framing options from PCD to PICC (see Figure 37)(^b)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>'87' '01' Selected framing options from PICC to PCD (see Figure 37)(^c)</td>
</tr>
<tr>
<td>'A8'</td>
<td>Frame Format Acknowledgement</td>
<td>0</td>
<td>—</td>
</tr>
</tbody>
</table>

\(^a\) 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).

**NOTE 2** Standard frame support is mandatory.

---

**Figure 36 — Frame Formats**
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.

<table>
<thead>
<tr>
<th>Step</th>
<th>PCD</th>
<th>PICC</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>S(PARAMETERS) ('A0 02 A5 00' CRC)</td>
<td>← →</td>
</tr>
<tr>
<td>2</td>
<td>S(PARAMETERS) ('A0 0E' 'A6 0C')</td>
<td>← →</td>
</tr>
<tr>
<td>3</td>
<td>S(PARAMETERS) ('A0 0E' 'A7 0C' '84 01 02' '85 01 02' '86 01 01' '87 01 01' CRC)</td>
<td>← →</td>
</tr>
<tr>
<td>4</td>
<td>← →</td>
<td>S(PARAMETERS) ('A0 02 A8 00' CRC)</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>PCD Action</th>
<th>Status PICC 1</th>
<th>Status PICC 2</th>
<th>Status PICC 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power On field</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Three PICCs enter the field.</td>
<td>IDLE</td>
<td>IDLE</td>
<td>IDLE</td>
</tr>
<tr>
<td>Activate PICC with CID = 1</td>
<td>ACTIVE(1)</td>
<td>IDLE</td>
<td>IDLE</td>
</tr>
<tr>
<td>Any data transmission with CID = 1</td>
<td>ACTIVE(1)</td>
<td>IDLE</td>
<td>IDLE</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Activate PICC with CID = 2</td>
<td>ACTIVE(1)</td>
<td>ACTIVE(2)</td>
<td>IDLE</td>
</tr>
<tr>
<td>Any data transmission with CID = 1,2</td>
<td>ACTIVE(1)</td>
<td>ACTIVE(2)</td>
<td>IDLE</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Activate PICC with CID = 3</td>
<td>ACTIVE(1)</td>
<td>ACTIVE(2)</td>
<td>ACTIVE(3)</td>
</tr>
<tr>
<td>Any data transmission with CID = 1,2,3</td>
<td>ACTIVE(1)</td>
<td>ACTIVE(2)</td>
<td>ACTIVE(3)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S(DESELECT) Command with CID = 3</td>
<td>ACTIVE(1)</td>
<td>ACTIVE(2)</td>
<td>HALT</td>
</tr>
<tr>
<td>S(DESELECT) Command with CID = 2</td>
<td>ACTIVE(1)</td>
<td>HALT</td>
<td>HALT</td>
</tr>
<tr>
<td>S(DESELECT) Command with CID = 1</td>
<td>HALT</td>
<td>HALT</td>
<td>HALT</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Any block</td>
<td>====&gt; correctly received</td>
</tr>
<tr>
<td>Any block</td>
<td>= ===&gt; erroneously received</td>
</tr>
<tr>
<td>Any block</td>
<td>=&gt; nothing received (FWT time-out)</td>
</tr>
<tr>
<td>Separator line</td>
<td>--- end of the smallest protocol operation</td>
</tr>
<tr>
<td>I(1)x</td>
<td>I-block with chaining bit set and block number x</td>
</tr>
<tr>
<td>I(0)x</td>
<td>I-block with chaining bit not set (last block of chain) and block number x</td>
</tr>
<tr>
<td>R(ACK)x</td>
<td>R-block indicating a positive acknowledge</td>
</tr>
<tr>
<td>R(NAK)x</td>
<td>R-block indicating a negative acknowledge</td>
</tr>
<tr>
<td>S(...)</td>
<td>S-block</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Comment</th>
<th>Block No. (0)</th>
<th>PCD</th>
<th>PICC</th>
<th>Block No. (1)</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. rule 1</td>
<td>0</td>
<td>I(0)</td>
<td>====&gt;</td>
<td>0</td>
<td>rule D</td>
</tr>
<tr>
<td>2. rule B</td>
<td>1</td>
<td>I(0)</td>
<td>&lt;==&gt;</td>
<td>I(0)</td>
<td>rule 10</td>
</tr>
<tr>
<td>3. rule B</td>
<td>I(0)</td>
<td>====&gt;</td>
<td>1</td>
<td>rule D</td>
<td></td>
</tr>
<tr>
<td>4. rule B</td>
<td>0</td>
<td>&lt;==&gt;</td>
<td>I(0)</td>
<td>rule 10</td>
<td></td>
</tr>
</tbody>
</table>
### B.3.2 Request for waiting time extension

**Scenario 2 Waiting time extension**

<table>
<thead>
<tr>
<th>Comment</th>
<th>Block No. (0)</th>
<th>PCD</th>
<th>PICC</th>
<th>Block No. (1)</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>rule 1</td>
<td>I(0)_b</td>
<td>===&gt;</td>
<td>0</td>
<td>rule D</td>
</tr>
<tr>
<td>2.</td>
<td></td>
<td></td>
<td></td>
<td>&lt;==</td>
<td>S(WTX) request</td>
</tr>
<tr>
<td>3.</td>
<td>rule 3</td>
<td></td>
<td></td>
<td>===&gt;</td>
<td>I(0)_b</td>
</tr>
<tr>
<td>4.</td>
<td>rule B</td>
<td>1</td>
<td></td>
<td>===&gt;</td>
<td>I(0)_b</td>
</tr>
<tr>
<td>5.</td>
<td>rule B</td>
<td>0</td>
<td></td>
<td>&lt;==</td>
<td>I(0)_b</td>
</tr>
</tbody>
</table>

### B.3.3 DESELECT

**Scenario 3 DESELECT**

<table>
<thead>
<tr>
<th>Comment</th>
<th>Block No. (0)</th>
<th>PCD</th>
<th>PICC</th>
<th>Block No. (1)</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>rule 1</td>
<td>I(0)_b</td>
<td>===&gt;</td>
<td>0</td>
<td>rule D</td>
</tr>
<tr>
<td>2.</td>
<td>rule B</td>
<td>1</td>
<td></td>
<td>&lt;==</td>
<td>I(0)_b</td>
</tr>
<tr>
<td>3.</td>
<td>S(DESELECT) request</td>
<td>===&gt;</td>
<td></td>
<td>&lt;==</td>
<td>S(DESELECT) response</td>
</tr>
</tbody>
</table>

### B.3.4 Chaining

**Scenario 4 PCD uses chaining**

<table>
<thead>
<tr>
<th>Comment</th>
<th>Block No. (0)</th>
<th>PCD</th>
<th>PICC</th>
<th>Block No. (1)</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>rule 1</td>
<td>I(1)_b</td>
<td>===&gt;</td>
<td>0</td>
<td>rule D</td>
</tr>
<tr>
<td>2.</td>
<td>rule B</td>
<td>1</td>
<td></td>
<td>&lt;==</td>
<td>R(ACK) _0</td>
</tr>
<tr>
<td>3.</td>
<td>rule 7</td>
<td>I(0)_b</td>
<td>===&gt;</td>
<td>1</td>
<td>rule D</td>
</tr>
<tr>
<td>4.</td>
<td>rule B</td>
<td>0</td>
<td></td>
<td>&lt;==</td>
<td>I(0)_b</td>
</tr>
<tr>
<td>5.</td>
<td>I(0)_b</td>
<td>===&gt;</td>
<td></td>
<td></td>
<td>rule D</td>
</tr>
<tr>
<td>6.</td>
<td>rule B</td>
<td>1</td>
<td></td>
<td>&lt;==</td>
<td>I(0)_b</td>
</tr>
</tbody>
</table>

**Scenario 5 PICC uses chaining**

<table>
<thead>
<tr>
<th>Comment</th>
<th>Block No. (0)</th>
<th>PCD</th>
<th>PICC</th>
<th>Block No. (1)</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>rule 1</td>
<td>I(0)_b</td>
<td>===&gt;</td>
<td>0</td>
<td>rule D</td>
</tr>
<tr>
<td>2.</td>
<td>rule B</td>
<td>1</td>
<td></td>
<td>&lt;==</td>
<td>I(1)_b</td>
</tr>
<tr>
<td>3.</td>
<td>rule 2</td>
<td>R(ACK) _1</td>
<td>===&gt;</td>
<td>1</td>
<td>rule E</td>
</tr>
<tr>
<td>4.</td>
<td>rule B</td>
<td>0</td>
<td></td>
<td>&lt;==</td>
<td>I(0)_b</td>
</tr>
<tr>
<td>5.</td>
<td>I(0)_b</td>
<td>===&gt;</td>
<td></td>
<td></td>
<td>rule D</td>
</tr>
<tr>
<td>6.</td>
<td>rule B</td>
<td>1</td>
<td></td>
<td>&lt;==</td>
<td>I(0)_b</td>
</tr>
</tbody>
</table>
### B.3.5 PICC Presence check

#### Scenario 6 PICC presence check using method 1

<table>
<thead>
<tr>
<th>Comment</th>
<th>Block No. (0)</th>
<th>PCD</th>
<th>PICC</th>
<th>Block No. (1)</th>
<th>Comment</th>
</tr>
</thead>
</table>
| 1. rule 1 and method 1 | I(0)
| 2. rule B        | 1           | <==>      | I(0)      | 0             | rule D      |

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

<table>
<thead>
<tr>
<th>Comment</th>
<th>Block No. (0)</th>
<th>PCD</th>
<th>PICC</th>
<th>Block No. (1)</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. rule 1 and method 2</td>
<td>R(NAK)</td>
<td>&lt;==&gt;</td>
<td>R(ACK)</td>
<td>0</td>
<td>rule E note</td>
</tr>
<tr>
<td>2. rule 6 note and method 2</td>
<td>R(NAK)</td>
<td>&lt;==&gt;</td>
<td>R(ACK)</td>
<td></td>
<td>rule 12</td>
</tr>
<tr>
<td>3. no change</td>
<td></td>
<td>&lt;==&gt;</td>
<td>R(ACK)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4. rule 6 note</td>
<td></td>
<td>&lt;==&gt;</td>
<td>R(ACK)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5. I(0)</td>
<td>&lt;==&gt;</td>
<td>0</td>
<td>rule D</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6. rule B</td>
<td>1</td>
<td>&lt;==&gt;</td>
<td>I(0)</td>
<td>rule D</td>
<td></td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Comment</th>
<th>Block No. (0)</th>
<th>PCD</th>
<th>PICC</th>
<th>Block No. (1)</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. rule 1</td>
<td>I(0)</td>
<td>&lt;==&gt;</td>
<td>0</td>
<td>rule D</td>
<td></td>
</tr>
<tr>
<td>2. rule B</td>
<td>1</td>
<td>&lt;==&gt;</td>
<td>I(0)</td>
<td>rule 10</td>
<td></td>
</tr>
<tr>
<td>3. method 2-a</td>
<td>R(NAK)</td>
<td>&lt;==&gt;</td>
<td>R(ACK)</td>
<td>rule E note</td>
<td></td>
</tr>
<tr>
<td>4. rule 6 note</td>
<td>no change</td>
<td>&lt;==&gt;</td>
<td>R(ACK)</td>
<td>rule 12</td>
<td></td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Comment</th>
<th>Block No. (0)</th>
<th>PCD</th>
<th>PICC</th>
<th>Block No. (1)</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. rule 1</td>
<td>I(0)</td>
<td>&lt;==&gt;</td>
<td>0</td>
<td>rule D</td>
<td></td>
</tr>
<tr>
<td>2. rule B</td>
<td>1</td>
<td>&lt;==&gt;</td>
<td>I(0)</td>
<td>rule 10</td>
<td></td>
</tr>
<tr>
<td>3. method 2-b</td>
<td>0</td>
<td>&lt;==&gt;</td>
<td>R(NAK)</td>
<td>rule 11</td>
<td></td>
</tr>
<tr>
<td>4. rule B</td>
<td>1</td>
<td>&lt;==&gt;</td>
<td>I(0)</td>
<td>rule D</td>
<td></td>
</tr>
<tr>
<td>5. I(0)</td>
<td>&lt;==&gt;</td>
<td>1</td>
<td>rule D</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6. rule B</td>
<td>0</td>
<td>&lt;==&gt;</td>
<td>I(0)</td>
<td>rule 10</td>
<td></td>
</tr>
</tbody>
</table>
B.3.6 Exchange of additional parameters

Scenario 25 Exchange of additional parameters

<table>
<thead>
<tr>
<th>Comment</th>
<th>Block No. (0)</th>
<th>PCD</th>
<th>PICC</th>
<th>Block No. (1)</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>rule 1</td>
<td>I(0)_h</td>
<td>====&gt;</td>
<td>0</td>
<td>rule D</td>
</tr>
<tr>
<td>2.</td>
<td>rule B</td>
<td>1</td>
<td>&lt;====</td>
<td>I(0)_h</td>
<td>rule 10</td>
</tr>
<tr>
<td>3.</td>
<td>S(PARAMETERS) request</td>
<td>====&gt;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.</td>
<td>1</td>
<td>&lt;====</td>
<td>S(PARAMETERS) response</td>
<td>rule 3</td>
<td></td>
</tr>
<tr>
<td>5.</td>
<td>I(0)_h</td>
<td>====&gt;</td>
<td>1</td>
<td>rule D</td>
<td></td>
</tr>
<tr>
<td>6.</td>
<td>rule B</td>
<td>0</td>
<td>&lt;====</td>
<td>I(0)_h</td>
<td>rule 10</td>
</tr>
</tbody>
</table>

B.4 Error handling

B.4.1 Exchange of I-blocks

Scenario 10 Start of protocol

<table>
<thead>
<tr>
<th>Comment</th>
<th>Block No. (0)</th>
<th>PCD</th>
<th>PICC</th>
<th>Block No. (1)</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>rule 1</td>
<td>I(0)_h</td>
<td>====&gt;</td>
<td>0</td>
<td>rule D</td>
</tr>
<tr>
<td>2.</td>
<td>rule B</td>
<td>1</td>
<td>&lt;====</td>
<td>I(0)_h</td>
<td>rule 10</td>
</tr>
<tr>
<td>3.</td>
<td>S(PARAMETERS) request</td>
<td>====&gt;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.</td>
<td>1</td>
<td>&lt;====</td>
<td>S(PARAMETERS) response</td>
<td>rule 3</td>
<td></td>
</tr>
<tr>
<td>5.</td>
<td>I(0)_h</td>
<td>====&gt;</td>
<td>1</td>
<td>rule D</td>
<td></td>
</tr>
<tr>
<td>6.</td>
<td>rule B</td>
<td>0</td>
<td>&lt;====</td>
<td>I(0)_h</td>
<td>rule 10</td>
</tr>
</tbody>
</table>

Scenario 11 Exchange of I-blocks

<table>
<thead>
<tr>
<th>Comment</th>
<th>Block No. (0)</th>
<th>PCD</th>
<th>PICC</th>
<th>Block No. (1)</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>rule 1</td>
<td>I(0)_h</td>
<td>====&gt;</td>
<td>0</td>
<td>rule D</td>
</tr>
<tr>
<td>2.</td>
<td>rule B</td>
<td>1</td>
<td>&lt;====</td>
<td>I(0)_h</td>
<td>rule 10</td>
</tr>
<tr>
<td>3.</td>
<td>I(0)_h</td>
<td>====&gt;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.</td>
<td>time-out</td>
<td>&lt;==</td>
<td>-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.</td>
<td>rule 4</td>
<td>R(NAK)_h</td>
<td>====&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6.</td>
<td>no change</td>
<td>&lt;====</td>
<td>R(ACK)_h</td>
<td>rule 12</td>
<td></td>
</tr>
<tr>
<td>7.</td>
<td>rule 6</td>
<td>I(0)_h</td>
<td>====&gt;</td>
<td>1</td>
<td>rule D</td>
</tr>
<tr>
<td>8.</td>
<td>rule B</td>
<td>0</td>
<td>&lt;====</td>
<td>I(0)_h</td>
<td>rule 10</td>
</tr>
<tr>
<td>9.</td>
<td>I(0)_h</td>
<td>====&gt;</td>
<td>0</td>
<td>rule D</td>
<td></td>
</tr>
<tr>
<td>10.</td>
<td>rule B</td>
<td>1</td>
<td>&lt;====</td>
<td>I(0)_h</td>
<td>rule 10</td>
</tr>
</tbody>
</table>
### B.3.6 Exchange of additional parameters

#### Scenario 25 Exchange of additional parameters

<table>
<thead>
<tr>
<th>Comment</th>
<th>Block No. (0)</th>
<th>PCD</th>
<th>PICC</th>
<th>Block No. (1)</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>rule 1</td>
<td>I(0)_b</td>
<td>==&gt;</td>
<td>0</td>
<td>rule D</td>
</tr>
<tr>
<td>2.</td>
<td></td>
<td></td>
<td>&lt;==&gt;</td>
<td>I(0)_b</td>
<td>rule 10</td>
</tr>
<tr>
<td>3.</td>
<td>rule 4</td>
<td>R(NAK)_0</td>
<td>==&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.</td>
<td>rule B</td>
<td>1</td>
<td>&lt;==</td>
<td>I(0)_h</td>
<td>rule 11</td>
</tr>
<tr>
<td>5.</td>
<td></td>
<td>I(0)_h</td>
<td>==&gt;</td>
<td>1</td>
<td>rule D</td>
</tr>
<tr>
<td>6.</td>
<td>rule B</td>
<td>0</td>
<td>&lt;==</td>
<td>I(0)_h</td>
<td>rule 10</td>
</tr>
</tbody>
</table>

### B.4 Error handling

#### B.4.1 Exchange of I-blocks

#### Scenario 10 Start of protocol

<table>
<thead>
<tr>
<th>Comment</th>
<th>Block No. (0)</th>
<th>PCD</th>
<th>PICC</th>
<th>Block No. (1)</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>rule 1</td>
<td>I(0)_b</td>
<td>==&gt;</td>
<td>0</td>
<td>rule D</td>
</tr>
<tr>
<td>2.</td>
<td></td>
<td></td>
<td>&lt;==&gt;</td>
<td>I(0)_b</td>
<td>rule 10</td>
</tr>
<tr>
<td>3.</td>
<td>S(PARAMETERS) request</td>
<td>==</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.</td>
<td></td>
<td>1</td>
<td>&lt;==</td>
<td>I(0)_h</td>
<td>rule 11</td>
</tr>
<tr>
<td>5.</td>
<td>I(0)_h</td>
<td>==&gt;</td>
<td>1</td>
<td>rule D</td>
<td></td>
</tr>
</tbody>
</table>

#### Scenario 11 Exchange of I-blocks

<table>
<thead>
<tr>
<th>Comment</th>
<th>Block No. (0)</th>
<th>PCD</th>
<th>PICC</th>
<th>Block No. (1)</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>rule 1</td>
<td>I(0)_b</td>
<td>==&gt;</td>
<td>0</td>
<td>rule D</td>
</tr>
<tr>
<td>2.</td>
<td></td>
<td></td>
<td>&lt;==&gt;</td>
<td>I(0)_b</td>
<td>rule 10</td>
</tr>
<tr>
<td>3.</td>
<td>time-out</td>
<td>&lt;=</td>
<td>==</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.</td>
<td>rule 4</td>
<td>R(NAK)_0</td>
<td>==&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.</td>
<td>rule B</td>
<td>1</td>
<td>&lt;==</td>
<td>I(0)_h</td>
<td>rule 11</td>
</tr>
<tr>
<td>6.</td>
<td></td>
<td>I(0)_h</td>
<td>==&gt;</td>
<td>1</td>
<td>rule D</td>
</tr>
<tr>
<td>7.</td>
<td>rule B</td>
<td>0</td>
<td>&lt;==</td>
<td>I(0)_h</td>
<td>rule 10</td>
</tr>
</tbody>
</table>

#### Scenario 12 Exchange of I-blocks

<table>
<thead>
<tr>
<th>Comment</th>
<th>Block No. (0)</th>
<th>PCD</th>
<th>PICC</th>
<th>Block No. (1)</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>rule 1</td>
<td>I(0)_b</td>
<td>==&gt;</td>
<td>0</td>
<td>rule D</td>
</tr>
<tr>
<td>2.</td>
<td></td>
<td></td>
<td>&lt;==&gt;</td>
<td>I(0)_b</td>
<td>rule 10</td>
</tr>
<tr>
<td>3.</td>
<td>rule 4</td>
<td>R(NAK)_0</td>
<td>==&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.</td>
<td>time-out</td>
<td>&lt;=</td>
<td>==</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.</td>
<td>rule B</td>
<td>1</td>
<td>&lt;==</td>
<td>I(0)_h</td>
<td>rule 11</td>
</tr>
<tr>
<td>6.</td>
<td></td>
<td>I(0)_h</td>
<td>==&gt;</td>
<td>1</td>
<td>rule D</td>
</tr>
</tbody>
</table>

#### Scenario 13 Exchange of I-blocks

<table>
<thead>
<tr>
<th>Comment</th>
<th>Block No. (0)</th>
<th>PCD</th>
<th>PICC</th>
<th>Block No. (1)</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>rule 1</td>
<td>I(0)_b</td>
<td>==&gt;</td>
<td>0</td>
<td>rule D</td>
</tr>
<tr>
<td>2.</td>
<td></td>
<td></td>
<td>&lt;==&gt;</td>
<td>I(0)_b</td>
<td>rule 10</td>
</tr>
<tr>
<td>3.</td>
<td>rule 4</td>
<td>R(NAK)_0</td>
<td>==&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.</td>
<td>time-out</td>
<td>&lt;=</td>
<td>==</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.</td>
<td>rule B</td>
<td>1</td>
<td>&lt;==</td>
<td>I(0)_h</td>
<td>rule 11</td>
</tr>
<tr>
<td>6.</td>
<td></td>
<td>I(0)_h</td>
<td>==&gt;</td>
<td>1</td>
<td>rule D</td>
</tr>
<tr>
<td>7.</td>
<td>rule B</td>
<td>0</td>
<td>&lt;==</td>
<td>I(0)_h</td>
<td>rule 10</td>
</tr>
</tbody>
</table>

#### Scenario 26 Exchange of I-blocks

<table>
<thead>
<tr>
<th>Comment</th>
<th>Block No. (0)</th>
<th>PCD</th>
<th>PICC</th>
<th>Block No. (1)</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>rule 1</td>
<td>I(0)_b</td>
<td>==&gt;</td>
<td>0</td>
<td>rule D</td>
</tr>
<tr>
<td>2.</td>
<td>rule B</td>
<td>1</td>
<td>&lt;==</td>
<td>I(0)_h</td>
<td>rule 10</td>
</tr>
<tr>
<td>3.</td>
<td>S(PARAMETERS) request</td>
<td>==</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.</td>
<td>time-out</td>
<td>&lt;=</td>
<td>==</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.</td>
<td>rule 8</td>
<td>S(PARAMETERS) request</td>
<td>==</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6.</td>
<td>&lt;== S(PARAMETERS) response</td>
<td>rule 3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7.</td>
<td>I(0)_h</td>
<td>==&gt;</td>
<td>1</td>
<td>rule D</td>
<td></td>
</tr>
<tr>
<td>8.</td>
<td>rule B</td>
<td>0</td>
<td>&lt;==</td>
<td>I(0)_h</td>
<td>rule 10</td>
</tr>
</tbody>
</table>
### B.4.2 Request for waiting time extension

#### Scenario 14 Request for waiting time extension

<table>
<thead>
<tr>
<th>Comment</th>
<th>Block No. (0)</th>
<th>PCD</th>
<th>PICC</th>
<th>Block No. (1)</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>rule 1</td>
<td>I(0)_b</td>
<td>====&gt;</td>
<td>0</td>
<td>rule D</td>
</tr>
<tr>
<td>2.</td>
<td></td>
<td></td>
<td></td>
<td>&lt;=≠= S(WTX) request</td>
<td>rule 9</td>
</tr>
<tr>
<td>3.</td>
<td>rule 4</td>
<td>R(NAK)_0</td>
<td>====&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.</td>
<td></td>
<td></td>
<td></td>
<td>&lt;=≠= S(WTX) request</td>
<td>rule 11</td>
</tr>
<tr>
<td>5.</td>
<td>rule 3</td>
<td>S(WTX) response</td>
<td>====&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6.</td>
<td>rule B</td>
<td>1</td>
<td>====&gt;</td>
<td>I(0)_b</td>
<td>rule 10</td>
</tr>
<tr>
<td>7.</td>
<td></td>
<td></td>
<td></td>
<td>&lt;=≠= I(0)_b</td>
<td></td>
</tr>
<tr>
<td>8.</td>
<td>rule B</td>
<td>0</td>
<td>====&gt;</td>
<td>I(0)_b</td>
<td>rule 10</td>
</tr>
</tbody>
</table>

#### Scenario 15 Request for waiting time extension

<table>
<thead>
<tr>
<th>Comment</th>
<th>Block No. (0)</th>
<th>PCD</th>
<th>PICC</th>
<th>Block No. (1)</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>rule 1</td>
<td>I(0)_b</td>
<td>====&gt;</td>
<td>0</td>
<td>rule D</td>
</tr>
<tr>
<td>2.</td>
<td></td>
<td></td>
<td></td>
<td>&lt;=≠= S(WTX) request</td>
<td>rule 9</td>
</tr>
<tr>
<td>3.</td>
<td>rule 4</td>
<td>R(NAK)_0</td>
<td>====&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.</td>
<td>time-out</td>
<td>&lt;= =</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.</td>
<td>rule 4</td>
<td>R(NAK)_0</td>
<td>====&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6.</td>
<td></td>
<td></td>
<td></td>
<td>&lt;=≠= S(WTX) request</td>
<td>rule 11</td>
</tr>
<tr>
<td>7.</td>
<td>rule 3</td>
<td>S(WTX) response</td>
<td>====&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8.</td>
<td>rule B</td>
<td>1</td>
<td>====&gt;</td>
<td>I(0)_b</td>
<td>rule 10</td>
</tr>
<tr>
<td>9.</td>
<td></td>
<td></td>
<td></td>
<td>&lt;=≠= I(0)_b</td>
<td></td>
</tr>
<tr>
<td>10.</td>
<td>rule B</td>
<td>0</td>
<td>====&gt;</td>
<td>I(0)_b</td>
<td>rule 10</td>
</tr>
</tbody>
</table>

#### Scenario 16 Request for waiting time extension

<table>
<thead>
<tr>
<th>Comment</th>
<th>Block No. (0)</th>
<th>PCD</th>
<th>PICC</th>
<th>Block No. (1)</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>rule 1</td>
<td>I(0)_b</td>
<td>====&gt;</td>
<td>0</td>
<td>rule D</td>
</tr>
<tr>
<td>2.</td>
<td></td>
<td></td>
<td></td>
<td>&lt;=≠= S(WTX) request</td>
<td>rule 9</td>
</tr>
<tr>
<td>3.</td>
<td>rule 3</td>
<td>S(WTX) response</td>
<td>====&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.</td>
<td>time-out</td>
<td>&lt;= =</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.</td>
<td>rule 4</td>
<td>R(NAK)_0</td>
<td>====&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6.</td>
<td></td>
<td></td>
<td></td>
<td>&lt;=≠= S(WTX) request</td>
<td>rule 11</td>
</tr>
<tr>
<td>7.</td>
<td>rule 3</td>
<td>S(WTX) response</td>
<td>====&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8.</td>
<td>rule B</td>
<td>1</td>
<td>====&gt;</td>
<td>I(0)_b</td>
<td>rule 10</td>
</tr>
<tr>
<td>9.</td>
<td></td>
<td></td>
<td></td>
<td>&lt;=≠= I(0)_b</td>
<td></td>
</tr>
<tr>
<td>10.</td>
<td>rule B</td>
<td>0</td>
<td>====&gt;</td>
<td>I(0)_b</td>
<td>rule 10</td>
</tr>
</tbody>
</table>
### Scenario 17 Request for waiting time extension

<table>
<thead>
<tr>
<th>Comment</th>
<th>Block No. (0)</th>
<th>PCD</th>
<th>PICC</th>
<th>Block No. (1)</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>rule 1</td>
<td>I(0)_b</td>
<td>====&gt;</td>
<td>0</td>
<td>rule D</td>
</tr>
<tr>
<td>2.</td>
<td></td>
<td></td>
<td></td>
<td>&lt;=≠=</td>
<td>S(WTX) request</td>
</tr>
<tr>
<td>3.</td>
<td>rule 3</td>
<td></td>
<td></td>
<td>&lt;=#=</td>
<td>I(0)_b</td>
</tr>
<tr>
<td>4.</td>
<td></td>
<td></td>
<td></td>
<td>&lt;=#=</td>
<td>I(0)_b</td>
</tr>
<tr>
<td>5.</td>
<td>rule 4</td>
<td></td>
<td></td>
<td>&lt;=≠=</td>
<td>I(0)_b</td>
</tr>
<tr>
<td>6.</td>
<td>rule B</td>
<td>1</td>
<td>====&gt;</td>
<td>1</td>
<td>rule D</td>
</tr>
</tbody>
</table>

### Scenario 18 Request for waiting time extension

<table>
<thead>
<tr>
<th>Comment</th>
<th>Block No. (0)</th>
<th>PCD</th>
<th>PICC</th>
<th>Block No. (1)</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>rule 1</td>
<td>I(0)_b</td>
<td>====&gt;</td>
<td>0</td>
<td>rule D</td>
</tr>
<tr>
<td>2.</td>
<td></td>
<td></td>
<td></td>
<td>&lt;=≠=</td>
<td>S(WTX) request</td>
</tr>
<tr>
<td>3.</td>
<td>rule 3</td>
<td></td>
<td></td>
<td>&lt;=#=</td>
<td>I(0)_b</td>
</tr>
<tr>
<td>4.</td>
<td></td>
<td></td>
<td></td>
<td>&lt;=#=</td>
<td>I(0)_b</td>
</tr>
<tr>
<td>5.</td>
<td>rule 4</td>
<td></td>
<td></td>
<td>&lt;=≠=</td>
<td>I(0)_b</td>
</tr>
<tr>
<td>6.</td>
<td>rule B</td>
<td>1</td>
<td>====&gt;</td>
<td>1</td>
<td>rule D</td>
</tr>
</tbody>
</table>

### B.4.3 DESELECT

### Scenario 19 DESELECT

<table>
<thead>
<tr>
<th>Comment</th>
<th>Block No. (0)</th>
<th>PCD</th>
<th>PICC</th>
<th>Block No. (1)</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>rule 1</td>
<td>I(0)_b</td>
<td>====&gt;</td>
<td>0</td>
<td>rule D</td>
</tr>
<tr>
<td>2.</td>
<td>rule B</td>
<td></td>
<td></td>
<td>&lt;=#=</td>
<td>I(0)_b</td>
</tr>
<tr>
<td>3.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>S(DESELECT) request</td>
</tr>
<tr>
<td>4.</td>
<td>time-out</td>
<td></td>
<td></td>
<td>&lt;=#=</td>
<td></td>
</tr>
<tr>
<td>5.</td>
<td>rule B</td>
<td></td>
<td></td>
<td>&lt;=#=</td>
<td>S(DESELECT) request</td>
</tr>
<tr>
<td>6.</td>
<td></td>
<td></td>
<td></td>
<td>&lt;=#=</td>
<td>S(DESELECT) response</td>
</tr>
</tbody>
</table>
### B.4.4 Chaining

#### Scenario 20 PCD uses chaining

<table>
<thead>
<tr>
<th>Comment</th>
<th>Block No. (0)</th>
<th>PCD</th>
<th>PICC</th>
<th>Block No. (1)</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>rule 1</td>
<td>I(1)_b</td>
<td>====&gt;</td>
<td>0</td>
<td>rule D</td>
</tr>
<tr>
<td>2.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.</td>
<td>rule 4</td>
<td>R(NAK)_0</td>
<td>====&gt;</td>
<td>0</td>
<td>rule 11</td>
</tr>
<tr>
<td>4.</td>
<td>rule B</td>
<td>1</td>
<td>====&gt;</td>
<td>R(ACK)_0</td>
<td>rule 2</td>
</tr>
<tr>
<td>5.</td>
<td>rule 7</td>
<td>I(1)_b</td>
<td>====&gt;</td>
<td>1</td>
<td>rule D</td>
</tr>
<tr>
<td>6.</td>
<td>rule B</td>
<td>0</td>
<td>====&gt;</td>
<td>R(ACK)_1</td>
<td>rule 2</td>
</tr>
<tr>
<td>7.</td>
<td>rule 7</td>
<td>I(0)_b</td>
<td>====&gt;</td>
<td>0</td>
<td>rule D</td>
</tr>
<tr>
<td>8.</td>
<td>rule B</td>
<td>1</td>
<td>====&gt;</td>
<td>I(0)_b</td>
<td>rule 10</td>
</tr>
</tbody>
</table>

| 9.      | I(0)_b       | ====> | 1             | rule D  |
| 10.     | rule B       | 0    | ====> | I(0)_b       | rule 10 |

#### Scenario 21 PCD uses chaining

<table>
<thead>
<tr>
<th>Comment</th>
<th>Block No. (0)</th>
<th>PCD</th>
<th>PICC</th>
<th>Block No. (1)</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>rule 1</td>
<td>I(1)_b</td>
<td>====&gt;</td>
<td>0</td>
<td>rule D</td>
</tr>
<tr>
<td>2.</td>
<td>rule B</td>
<td>1</td>
<td>====&gt;</td>
<td>R(ACK)_0</td>
<td>rule 2</td>
</tr>
<tr>
<td>3.</td>
<td>rule 7</td>
<td>I(1)_b</td>
<td>====&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.</td>
<td>time-out</td>
<td></td>
<td>&lt;= =</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.</td>
<td>rule 4</td>
<td>R(NAK)_1</td>
<td>====&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6.</td>
<td>no change</td>
<td></td>
<td>====&gt;</td>
<td>R(ACK)_0</td>
<td>rule 12</td>
</tr>
<tr>
<td>7.</td>
<td>rule 6</td>
<td>I(1)_b</td>
<td>====&gt;</td>
<td>1</td>
<td>rule D</td>
</tr>
<tr>
<td>8.</td>
<td>rule B</td>
<td>0</td>
<td>====&gt;</td>
<td>R(ACK)_1</td>
<td>rule 2</td>
</tr>
<tr>
<td>9.</td>
<td>rule 7</td>
<td>I(0)_b</td>
<td>====&gt;</td>
<td>0</td>
<td>rule D</td>
</tr>
<tr>
<td>10.</td>
<td>rule B</td>
<td>1</td>
<td>====&gt;</td>
<td>I(0)_b</td>
<td>rule 10</td>
</tr>
</tbody>
</table>

| 11.     | I(0)_b       | ====> | 1             | rule D  |
| 12.     | rule B       | 0    | ====> | I(0)_b       | rule 10 |
### Scenario 22 PCD uses chaining

<table>
<thead>
<tr>
<th>Comment</th>
<th>Block No. (0)</th>
<th>PCD</th>
<th>PICC</th>
<th>Block No. (1)</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>rule 1</td>
<td>I(1)_b</td>
<td>====&gt;</td>
<td>0</td>
<td>rule D</td>
</tr>
<tr>
<td>2.</td>
<td></td>
<td></td>
<td>&lt;=≠=</td>
<td>R(ACK)_0</td>
<td>rule 2</td>
</tr>
<tr>
<td>3.</td>
<td>rule 4</td>
<td>R(NAK)_0</td>
<td>====&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.</td>
<td>time-out</td>
<td></td>
<td>&lt;=</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>5.</td>
<td>rule 4</td>
<td>R(NAK)_0</td>
<td>====&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6.</td>
<td>rule B</td>
<td>1</td>
<td>&lt;=≠=</td>
<td>R(ACK)_0</td>
<td>rule 11</td>
</tr>
<tr>
<td>7.</td>
<td>rule 7</td>
<td>I(1)_h</td>
<td>====&gt;</td>
<td>1</td>
<td>rule D</td>
</tr>
<tr>
<td>8.</td>
<td>rule B</td>
<td>0</td>
<td>&lt;=≠=</td>
<td>R(ACK)_1</td>
<td>rule 2</td>
</tr>
<tr>
<td>9.</td>
<td>rule 7</td>
<td>I(0)_h</td>
<td>====&gt;</td>
<td>0</td>
<td>rule D</td>
</tr>
<tr>
<td>10.</td>
<td>rule B</td>
<td>1</td>
<td>&lt;=≠=</td>
<td>I(0)_h</td>
<td>rule 10</td>
</tr>
</tbody>
</table>

### Scenario 23 PICC uses chaining

<table>
<thead>
<tr>
<th>Comment</th>
<th>Block No. (0)</th>
<th>PCD</th>
<th>PICC</th>
<th>Block No. (1)</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>rule 1</td>
<td>I(0)_b</td>
<td>====&gt;</td>
<td>0</td>
<td>rule D</td>
</tr>
<tr>
<td>2.</td>
<td>rule B</td>
<td>1</td>
<td>&lt;=≠=</td>
<td>I(1)_b</td>
<td>rule 10</td>
</tr>
<tr>
<td>3.</td>
<td>rule 2</td>
<td>R(ACK)_1</td>
<td>====&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.</td>
<td>time-out</td>
<td></td>
<td>&lt;=</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>5.</td>
<td>rule 5</td>
<td>R(ACK)_1</td>
<td>====&gt;</td>
<td>1</td>
<td>rule E</td>
</tr>
<tr>
<td>6.</td>
<td>rule B</td>
<td>0</td>
<td>&lt;=≠=</td>
<td>I(1)_h</td>
<td>rule 13</td>
</tr>
<tr>
<td>7.</td>
<td>rule 2</td>
<td>R(ACK)_0</td>
<td>====&gt;</td>
<td>0</td>
<td>rule E</td>
</tr>
<tr>
<td>8.</td>
<td>rule B</td>
<td>1</td>
<td>&lt;=≠=</td>
<td>I(0)_h</td>
<td>rule 13</td>
</tr>
<tr>
<td>9.</td>
<td></td>
<td>I(0)_h</td>
<td>====&gt;</td>
<td>1</td>
<td>rule D</td>
</tr>
<tr>
<td>10.</td>
<td>rule B</td>
<td>0</td>
<td>&lt;=≠=</td>
<td>I(0)_h</td>
<td>rule 10</td>
</tr>
</tbody>
</table>

### Scenario 24 PICC uses chaining

<table>
<thead>
<tr>
<th>Comment</th>
<th>Block No. (0)</th>
<th>PCD</th>
<th>PICC</th>
<th>Block No. (1)</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>rule 1</td>
<td>I(0)_b</td>
<td>====&gt;</td>
<td>0</td>
<td>rule D</td>
</tr>
<tr>
<td>2.</td>
<td>rule B</td>
<td>1</td>
<td>&lt;=≠=</td>
<td>I(1)_b</td>
<td>rule 10</td>
</tr>
<tr>
<td>3.</td>
<td>rule 2</td>
<td>R(ACK)_1</td>
<td>====&gt;</td>
<td>1</td>
<td>rule E</td>
</tr>
<tr>
<td>4.</td>
<td></td>
<td></td>
<td>&lt;=≠=</td>
<td>I(1)_h</td>
<td>rule 13</td>
</tr>
<tr>
<td>5.</td>
<td>rule 5</td>
<td>R(ACK)_1</td>
<td>====&gt;</td>
<td>no change</td>
<td></td>
</tr>
<tr>
<td>6.</td>
<td>rule B</td>
<td>0</td>
<td>&lt;=≠=</td>
<td>I(1)_h</td>
<td>rule 11</td>
</tr>
<tr>
<td>7.</td>
<td>rule 2</td>
<td>R(ACK)_0</td>
<td>====&gt;</td>
<td>0</td>
<td>rule E</td>
</tr>
<tr>
<td>8.</td>
<td>rule B</td>
<td>1</td>
<td>&lt;=≠=</td>
<td>I(0)_h</td>
<td>rule 13</td>
</tr>
<tr>
<td>9.</td>
<td></td>
<td>I(0)_h</td>
<td>====&gt;</td>
<td>1</td>
<td>rule D</td>
</tr>
<tr>
<td>10.</td>
<td>rule B</td>
<td>0</td>
<td>&lt;=≠=</td>
<td>I(0)_h</td>
<td>rule 10</td>
</tr>
</tbody>
</table>
Annex C
(informative)

Block and frame coding overview

This Annex 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)\textsuperscript{b} (7 bit)
WUPA (1010010)\textsuperscript{b} (7 bit)
REQB/WUPB (00000101)\textsuperscript{b}
Slot-MARKER (Type B only) (xxxx0101)\textsuperscript{b}
SELECT (Type A only) (1001xxxx)\textsuperscript{b}
ATTRIB (Type B only) (00011101)\textsuperscript{b}
HLTA (01010000)\textsuperscript{b}
HLTB (01010000)\textsuperscript{b}

Definitions made in this part of ISO/IEC 14443:

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

Table C.1 — Block and frame coding

<table>
<thead>
<tr>
<th>Bit</th>
<th>I-block PCB</th>
<th>R-block PCB</th>
<th>S-block PCB</th>
</tr>
</thead>
<tbody>
<tr>
<td>b8</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>b7</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>b6</td>
<td>0 (1 is RFU)</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>b5</td>
<td>Chaining ACK/NAK</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>b4</td>
<td>CID CID CID</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>b3</td>
<td>NAD 0 (no NAD)</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>b2</td>
<td>1</td>
<td>1 (0 is RFU)</td>
<td>0</td>
</tr>
<tr>
<td>b1</td>
<td>Block number</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Table C.1 describes the first byte of the defined block and frame coding.

### Table C.1 — Block and frame coding

<table>
<thead>
<tr>
<th>Bit</th>
<th>I-block PCB</th>
<th>R-block PCB</th>
<th>S-block PCB</th>
<th>REQB/WUPB</th>
<th>Slot-MARKER</th>
<th>SE-LECT</th>
<th>AT-TRIB</th>
<th>HLTA</th>
<th>HLTB</th>
<th>RATS</th>
<th>PPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>b8</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>X</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>b7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>X</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>b6</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>X</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>b5</td>
<td>Chain-</td>
<td>ACK/NAK</td>
<td>0</td>
<td>0</td>
<td>X</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>b4</td>
<td>CID</td>
<td>CID</td>
<td>CID</td>
<td>0</td>
<td>X</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>X</td>
</tr>
<tr>
<td>b3</td>
<td>NAD</td>
<td>0 (no NAD)</td>
<td>0 (no NAD)</td>
<td>1</td>
<td>X</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>X</td>
</tr>
<tr>
<td>b2</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>X</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>X</td>
</tr>
<tr>
<td>b1</td>
<td>Block number</td>
<td>Block number</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>X</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>X</td>
</tr>
</tbody>
</table>
Annex D
(normative)

Bit rates of 3\(fc/4\), \(fc\), 3\(fc/2\) and 2\(fc\) from PCD to PICC

D.1 Function tags identifier definition

The information field shall contain tags and values as defined in Table D.1.

<table>
<thead>
<tr>
<th>Tag (Hex)</th>
<th>Description</th>
<th>Length (Hex)</th>
<th>Value</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>'A1'</td>
<td>Bit rates Request</td>
<td>'0'</td>
<td>—</td>
<td></td>
</tr>
</tbody>
</table>
| 'A2'     | Bit rates Indication                 | L            | '80'  | '02' Supported bit rates from PCD to PICC
|          |                                      |              |       | 1st byte is specified in Figure 24
|          |                                      |              |       | 2nd byte is specified in Figure D.1 |
|          |                                      | '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 |
| 'A3'     | Bit rates Activation                 | L            | '83'  | '02' Selected bit rate from PCD to PICC b
|          |                                      |              |       | 1st byte is specified in Figure 24
|          |                                      |              |       | 2nd byte is specified in Figure D.1 |
|          |                                      | '84'         | '02'  | Selected bit rate from PICC to PCD b
|          |                                      |              |       | 1st byte is specified in Figure 24
|          |                                      |              |       | 2nd byte set to ‘00’, other values are RFU |
| 'A4'     | Bit rates Acknowledgement            | '0'          | —     | |

---

a Shall be omitted for Type A PICCs.
b 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.
c 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; and
— the PICC shall apply no character separation.
D.1 Function tags identifier definition

The information field shall contain tags and values as defined in Table D.1.

Table D.1 — Function tags identifier definition

<table>
<thead>
<tr>
<th>Tag</th>
<th>(Hex)</th>
<th>Description</th>
<th>Length</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>'A1'</td>
<td></td>
<td>Bit rates</td>
<td></td>
<td></td>
</tr>
<tr>
<td>'0'</td>
<td></td>
<td>Request</td>
<td></td>
<td></td>
</tr>
<tr>
<td>'A2'</td>
<td></td>
<td>Bit rates</td>
<td></td>
<td></td>
</tr>
<tr>
<td>'L'</td>
<td></td>
<td>Indication</td>
<td></td>
<td></td>
</tr>
<tr>
<td>'80'</td>
<td>'02'</td>
<td>Supported bit rates from PCD to PICC</td>
<td>1st byte is specified in Figure 24</td>
<td>2nd byte is specified in Figure D.1</td>
</tr>
<tr>
<td>'81'</td>
<td>'02'</td>
<td>Supported bit rates from PICC to PCD</td>
<td>1st byte is specified in Figure 24</td>
<td>2nd byte set to '00', other values are RFU</td>
</tr>
<tr>
<td>'A3'</td>
<td></td>
<td>Bit rates</td>
<td></td>
<td></td>
</tr>
<tr>
<td>'L'</td>
<td></td>
<td>Activation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>'83'</td>
<td>'02'</td>
<td>Selected bit rate from PCD to PICC</td>
<td>1st byte is specified in Figure 24</td>
<td>2nd byte is specified in Figure D.1</td>
</tr>
<tr>
<td>'84'</td>
<td>'02'</td>
<td>Selected bit rate from PICC to PCD</td>
<td>1st byte is specified in Figure 24</td>
<td>2nd byte set to '00', other values are RFU</td>
</tr>
<tr>
<td>'85'</td>
<td>'01'</td>
<td>Selected framing options PICC to PCD</td>
<td>(see Figure 25)</td>
<td></td>
</tr>
</tbody>
</table>

---

**Figure D.1 — Coding of bit rates of 3fc/4, fc, 3fc/2 and 2fc from PCD to PICC**
Annex E
(informative)

CRC_32 encoding

E.1 CRC_32 encoding

This Annex is provided for explanatory purposes and indicates the bit patterns in enhanced block. It is included for the purpose of checking a CRC_32 encoding.

Initial Value = 'FFFFFFFF'.

EXAMPLE In Figure E.1, transmission of first byte = '06', second byte = '00', third byte = '0A', fourth byte = '01', fifth byte = '01', sixth byte = '02', and appended CRC_32 is illustrated. Calculated CRC_32 = '8098F1FE'.

<table>
<thead>
<tr>
<th>1st byte</th>
<th>2nd byte</th>
<th>3rd byte</th>
<th>4th byte</th>
<th>5th byte</th>
<th>6th byte</th>
<th>CRC_32</th>
</tr>
</thead>
<tbody>
<tr>
<td>06</td>
<td>00</td>
<td>0A</td>
<td>01</td>
<td>01</td>
<td>02</td>
<td>80</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>98</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>F1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>FE</td>
</tr>
</tbody>
</table>

Figure E.1 — Enhanced frame including CRC_32

E.2 Code sample written in ANSI C language for CRC_32 calculation

```c
// ComputeCrc32.cpp : Defines the entry point for the console application

#include <stdio.h>

unsigned long ComputeCrc32(unsigned char *Data) {
    unsigned long c;
    unsigned char d, e, f;
    unsigned int i;
    unsigned int Length = 6;
    // initial value
    c = 0xffffffff;
    // compute CRC
    for (i = 0; i < Length; i++) {
        d = Data[i];
        e = c ^ d;
        f = e ^ (e << 6);
        c = (c >>  8) ^ (f << 24) ^ (f << 23) ^ (f << 22) ^
            (f << 20) ^ (f << 19) ^ (f << 17) ^ (f << 16) ^
            (f << 14) ^ (f << 13) ^ (f << 12) ^ (f <<  8) ^
            (f <<  2) ^ (f <<  1) ^ (f >>  2);
    }
    // invert result
    c = c ^ 0xffffffff;
    return c;
}

int main(void) {
    unsigned char BuffCRC_32[6] = {0x06, 0x00, 0x0A, 0x01, 0x01, 0x02};
    unsigned int Crc32;
    int i;
    ```
CRC_32 encoding

E.1 CRC_32 encoding

This Annex is provided for explanatory purposes and indicates the bit patterns in enhanced block. It is included for the purpose of checking a CRC_32 encoding.

Initial Value = 'FFFFFFFF'.

EXAMPLE In Figure E.1, transmission of first byte = '06', second byte = '00', third byte = '0A', fourth byte = '01', fifth byte = '01', sixth byte = '02', and appended CRC_32 is illustrated. Calculated CRC_32 = '8098F1FE'.

Figure E.1 — Enhanced frame including CRC_32

E.2 Code sample written in ANSI C language for CRC_32 calculation

```c
#include <stdio.h>

unsigned long ComputeCrc32(unsigned char *Data) {
    unsigned long c;
    unsigned char d, e, f;
    unsigned int i;
    unsigned int Length = 6;

    // initial value
    c = 0xffffffff;

    // compute CRC
    for (i = 0; i < Length; i++) {
        d = Data[i];
        e = c ^ d;
        f = e ^ (e << 6);
        c = (c >> 8) ^ (f << 24) ^ (f << 23) ^ (f << 22) ^ (f << 20) ^ (f << 19) ^ (f << 17) ^ (f << 16) ^ (f << 14) ^ (f << 13) ^ (f << 12) ^ (f << 8) ^ (f << 2) ^ (f << 1) ^ (f >> 2);
    }

    // invert result
    c = c ^ 0xffffffff;
    return c;
}

int main(void) {
    unsigned char BuffCRC_32[6] = {0x06, 0x00, 0x0A, 0x01, 0x01, 0x02};
    unsigned int Crc32;
    int i;
    // revert byte order to get least significant byte first
    printf("\n");
    printf("CRC-32 reference results ISO/IEC 14443-4\n");
    printf("Crc-32 G(x) = x^32 + x^26 + x^23 + x^22 + x^16 + x^12 + x^11 + x^10 + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1\n\n");
    printf("CRC_32 of [ "");
    for(i=0; i<6; i++) printf("%02X ",BuffCRC_32[i]);
    Crc32 = ComputeCrc32(BuffCRC_32);
    printf("] Transmitted: %02X then %02X then %02X then %02X.
", (Crc32 & 0xFF), ((Crc32 >> 8) & 0xFF), ((Crc32 >> 16) & 0xFF), ((Crc32 >> 24) & 0xFF));

    return 0;
}
```

© ISO/IEC 2016 – All rights reserved
Annex F
(informative)

Frame with error correction

F.1 Example

This Annex is provided for explanatory purposes and shows how an I-block in using enhanced block format is transmitted. The example in Figure F.1 uses CID = '01' and INF-field = '0102'.

The enhanced block is divided into 7-byte blocks. In our example, there are two blocks. The second block needs to be padded with 4 'FF' bytes, illustrated in italic letters in Figure F.1. Each block has appended the Hamming control byte (bold letters in Figure F.1). These two blocks are called enhanced block with error correction. Before this enhanced block with error correction SYNC is prepended. Up to here everything is the same for all communication types.

When using the frame format defined in 10.2, the frame with error correction starts with S and ends with E as specified in ISO/IEC 14443-3:2016, Clause 4. SYNC (which cannot be suppressed) and the enhanced block with error correction are sent without parity bits of ISO/IEC 14443-3:2016, 6.2.3.2.

When using the frame format defined in 10.3, the frame with error correction starts with SOF and ends with EOF. SYNC and the enhanced block with error correction are sent in the character transmission format defined in ISO/IEC 14443-3:2016, 7.1.1. Start bit, stop bit, SOF, EOF and SYNC may be suppressed in accordance with 10.5.

F.2 Code sample written in ANSI C language for Hamming control byte calculation

```c
#include <stdio.h>

unsigned char ComputeHammingByte(unsigned char* pEnhancedSubBlock)
{
    unsigned char currentByte;
    unsigned char currentBit;
    unsigned char c = 0x00; // Hamming control byte
    unsigned char A[] = {
        0x03, 0x05, 0x06, 0x07, 0x09, 0x0A, 0x0B, 0x0C,
        0x0D, 0x0E, 0x0F, 0x11, 0x12, 0x13, 0x14, 0x15,
    };
    for (currentByte = 0; currentByte < 7; currentByte++) // 7 bytes
    {
        for (currentBit = 0; currentBit < 8; currentBit++) // 8 bits each
        {
            if ((pEnhancedSubBlock[currentByte] >> currentBit) & 0x01)
            {
                c = c ^ A[currentByte*8 + currentBit];
            }
        }
    }
    c = (c << 1) | 0x81;
    return c;
}
```

Figure F.1 — Composition of frame with error correction
Annex F
(informative)
Frame with error correction

This Annex is provided for explanatory purposes and shows how an I-block in using enhanced block format is transmitted. The example in Figure F.1 uses CID = '01' and INF-field = '0102'.

Figure F.1 — Composition of frame with error correction

The enhanced block is divided into 7-byte blocks. In our example, there are two blocks. The second block needs to be padded with 4 'FF' bytes, illustrated in italic letters in Figure F.1. Each block has appended the Hamming control byte (bold letters in Figure F.1). These two blocks are called enhanced block with error correction. Before this enhanced block with error correction, SYNC is prepended. Up to here everything is the same for all communication types.

When using the frame format defined in 10.2, the frame with error correction starts with S and ends with E as specified in ISO/IEC 14443-3:2016, Clause 4. SYNC (which cannot be suppressed) and the enhanced block with error correction are sent without parity bits of ISO/IEC 14443-3:2016, 6.2.3.2.

When using the frame format defined in 10.3, the frame with error correction starts with SOF and ends with EOF. SYNC and the enhanced block with error correction are sent in the character transmission format defined in ISO/IEC 14443-3:2016, 7.1.1. Start bit, stop bit, SOF, EOF and SYNC may be suppressed in accordance with 10.5.

F.2 Code sample written in ANSI C language for Hamming control byte calculation

```c
#include <stdio.h>

unsigned char ComputeHammingByte(unsigned char* pEnhancedSubBlock)
{
    unsigned char currentByte;
    unsigned char currentBit;
    unsigned char c = 0x00; // Hamming control byte
    unsigned char A[64] = {   // Hamming control generation matrix A
        0x03, 0x05, 0x06, 0x07, 0x09, 0x0A, 0x0B, 0x0C,
        0x0D, 0x0E, 0x0F, 0x11, 0x12, 0x13, 0x14, 0x15,
        0x16, 0x17, 0x18, 0x19, 0x1A, 0x1B, 0x1C, 0x1D,
        0x1E, 0x1F, 0x21, 0x22, 0x23, 0x24, 0x25, 0x26,
        0x27, 0x28, 0x29, 0x2A, 0x2B, 0x2C, 0x2D, 0x2E,
        0x2F, 0x30, 0x31, 0x32, 0x33, 0x34, 0x35, 0x36,
        0x37, 0x38, 0x39, 0x3A, 0x3B, 0x3C, 0x3D, 0x3E,
    };

    // iterate through all bits of the 7-byte sub-block from enhanced block
    for (currentByte = 0; currentByte < 7; currentByte++) // 7 bytes
    {
        for (currentBit = 0; currentBit < 8; currentBit++)  // 8 bits each
        {
            // if bit is equal 1 XOR with corresponding column of matrix A
            if ((pEnhancedSubBlock[currentByte] >> currentBit) & 0x01)
            {  
                c = c ^ A[currentByte*8 + currentBit];
            }
        }
    }

    // add padding bits
    c = (c << 1) | 0x81;
    return c;
}

int main(void)
{
    unsigned char EnhancedSubBlock[7] = {0x06, 0x00, 0x0A, 0x01, 0x01, 0x02, 0x80};
    unsigned char HammingByte;
    int i;
    HammingByte = ComputeHammingByte(EnhancedSubBlock);
    printf("Hamming Control Byte reference results ISO/IEC 14443-4\n")
    printf("Hamming Control Byte of enhanced sub-block [ ");
    for(i=0; i<7; i++) printf("%02X ",EnhancedSubBlock[i]);
    printf("] is %02X \n\n", HammingByte);
    return 0;
}
```
Annex G
(informative)

Framing options

This Annex illustrates all possible framing options dependent on communication type, bit rate, frame format and communication direction.

In Table G.1, “yes” means that framing options may be applied and “no” means that framing options shall not be applied.

<table>
<thead>
<tr>
<th></th>
<th>Standard frame</th>
<th>Frame with error correction</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PCD to PICC</td>
<td>PICC to PCD</td>
</tr>
<tr>
<td>Type A, bit rate &lt;= fc/16</td>
<td>no(^a)</td>
<td>no(^a)</td>
</tr>
<tr>
<td>Type A, fc/16 &lt; bit rate &lt;= fc/2</td>
<td>no(^b)</td>
<td>no(^a)</td>
</tr>
<tr>
<td>Type A, bit rate &gt; fc/2</td>
<td>no(^a)</td>
<td>N/A</td>
</tr>
<tr>
<td>Type B, bit rate &lt;= fc/2</td>
<td>no(^b)</td>
<td>yes</td>
</tr>
<tr>
<td>Type B, bit rate &gt; fc/2</td>
<td>yes</td>
<td>N/A</td>
</tr>
</tbody>
</table>

\(^a\) There are no start bits, stop bits, SOF and EOF in the frame format.

\(^b\) Start bits and stop bits are needed in the frame format in case of long series of ‘00’ of ‘FF’ bytes.

When activating framing options more than once, the new settings override the previous one, e.g. first activating SYNC suppression leads to SYNC suppression after the acknowledgement from the card. Then activating for instance Start/Stop bit suppression, only Start/Stop bit suppression is active after the acknowledgement from the card.
Annex G

(Framing options)

This Annex illustrates all possible framing options dependent on communication type, bit rate, frame format and communication direction.

In Table G.1, “yes” means that framing options may be applied and “no” means that framing options shall not be applied.

Table G.1 — Allowed application of framing options

<table>
<thead>
<tr>
<th>Standard frame</th>
<th>Frame with error correction</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCD to PICC</td>
<td>PICC to PCD</td>
</tr>
<tr>
<td>PCD to PICC</td>
<td>PICC to PCD</td>
</tr>
<tr>
<td>Type A, bit rate ( \leq FC/16 )</td>
<td>noa</td>
</tr>
<tr>
<td>Type A, bit rate ( &gt; FC/2 )</td>
<td>yes</td>
</tr>
<tr>
<td>Type B, bit rate ( \leq FC/2 )</td>
<td>nob</td>
</tr>
<tr>
<td>Type B, bit rate ( &gt; FC/2 )</td>
<td>yes</td>
</tr>
</tbody>
</table>

a There are no start bits, stop bits, SOF and EOF in the frame format.
b Start bits and stop bits are needed in the frame format in case of long series of '00' of 'FF' bytes.

When activating framing options more than once, the new settings override the previous one, e.g. first activating SYNC suppression leads to SYNC suppression after the acknowledgement from the card. Then activating for instance Start/Stop bit suppression, only Start/Stop bit suppression is active after the acknowledgement from the card.

### Bibliography

[1] ISO/IEC 7810, Identification cards — Physical characteristics


[6] ISO/IEC 13239, Information technology — Telecommunications and information exchange between systems — High-level data link control (HDLC) procedures

[7] ISO/IEC 15693 (all parts), Identification cards — Contactless integrated circuit cards — Vicinity cards


British Standards Institution (BSI)

BSI is the national body responsible for preparing British Standards and other standards-related publications, information and services.

BSI is incorporated by Royal Charter. British Standards and other standardization products are published by BSI Standards Limited.

About us
We bring together business, industry, government, consumers, innovators and others to shape their combined experience and expertise into standards-based solutions.

The knowledge embodied in our standards has been carefully assembled in a dependable format and refined through our open consultation process. Organizations of all sizes and across all sectors choose standards to help them achieve their goals.

Information on standards
We can provide you with the knowledge that your organization needs to succeed. Find out more about British Standards by visiting our website at bsigroup.com/standards or contacting our Customer Services team or Knowledge Centre.

Buying standards
You can buy and download PDF versions of BSI publications, including British and adopted European and international standards, through our website at bsigroup.com/shop, where hard copies can also be purchased.

If you need international and foreign standards from other Standards Development Organizations, hard copies can be ordered from our Customer Services team.

Copyright in BSI publications
All the content in BSI publications, including British Standards, is the property of and copyrighted by BSI or some person or entity that owns copyright in the information used (such as the international standardization bodies) and has formally licensed such information to BSI for commercial publication and use.

Save for the provisions below, you may not transfer, share or disseminate any portion of the standard to any other person. You may not adapt, distribute, commercially exploit, or publicly display the standard or any portion thereof in any manner whatsoever without BSI’s prior written consent.

Storing and using standards
Standards purchased in soft copy format:

- A British Standard purchased in soft copy format is licensed to a sole named user for personal or internal company use only.
- The standard may be stored on more than one device provided that it is accessible by the sole named user only and that only one copy is accessed at any one time.
- A single paper copy may be printed for personal or internal company use only.

Standards purchased in hard copy format:

- A British Standard purchased in hard copy format is for personal or internal company use only.
- It may not be further reproduced – in any format – to create an additional copy. This includes scanning of the document.

If you need more than 1 copy of the document, or if you wish to share the document on an internal network, you can save money by choosing a subscription package (see “Subscriptions”).

Reproducing extracts
For permission to reproduce content from BSI publications contact the BSI Copyright & Licensing team.

Subscriptions
Our range of subscription services are designed to make using standards easier for you. For further information on our subscription products go to bsigroup.com/subscriptions.

With British Standards Online (BSOL) you’ll have instant access to over 55,000 British and adopted European and international standards from your desktop. It’s available 24/7 and is refreshed daily so you’ll always be up to date.

You can keep in touch with standards developments and receive substantial discounts on the purchase price of standards, both in single copy and subscription format, by becoming a BSI Subscribing Member.

PLUS is an updating service exclusive to BSI Subscribing Members. You will automatically receive the latest hard copy of your standards when they’re revised or replaced.

To find out more about becoming a BSI Subscribing Member and the benefits of membership, please visit bsigroup.com/shop.

With a Multi-User Network Licence (MUNL) you are able to host standards publications on your intranet. Licences can cover as few or as many users as you wish. With updates supplied as soon as they’re available, you can be sure your documentation is current. For further information, email subscriptions@bsigroup.com.

Revisions
Our British Standards and other publications are updated by amendment or revision. We continually improve the quality of our products and services to benefit your business. If you find an inaccuracy or ambiguity within a British Standard or other BSI publication please inform the Knowledge Centre.

Useful Contacts
Customer Services
Tel: +44 345 086 9001
Email (orders): orders@bsigroup.com
Email (enquiries): cservices@bsigroup.com

Subscriptions
Tel: +44 345 086 9001
Email: subscriptions@bsigroup.com

Knowledge Centre
Tel: +44 20 8996 7004
Email: knowledgecentre@bsigroup.com

Copyright & Licensing
Tel: +44 20 8996 7070
Email: copyright@bsigroup.com

BSI Group Headquarters
389 Chiswick High Road London W4 4AL UK