

INTERNATIONAL STANDARD

ISO/IEC
7816-3

First edition
1989-09-15

AMENDMENT 1
1992-12-01

Identification cards – Integrated circuit(s) cards with contacts –

Part 3 :

Electronic signals and transmission protocols

AMENDMENT 1 : Protocol type T = 1, asynchronous
half duplex block transmission protocol

Cartes d'identification – Cartes à circuit(s) intégré(s) à contacts –

Partie 3 : Signaux électroniques et protocoles de transmission

*AMENDEMENT 1 : Type de protocole T = 1, protocole de transmission par blocs
half-duplex asynchrone*



Reference number
ISO/IEC 7816-3:1989/Amd.1:1992 (E)

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

Amendment 1 to International Standard ISO/IEC 7816-3:1989 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*.

ISO/IEC 7816-3 consists of the following parts, under the general title *Identification cards – Integrated circuit(s) cards with contacts*

- *Part 1 : Physical characteristics*
- *Part 2 : Dimensions and location of the contacts*
- *Part 3 : Electronic signals and transmission protocols*

Annex A of this amendment to ISO/IEC 7816-3 is for information only.

© ISO/IEC 1992

All rights reserved. No part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from the publisher.

International Organization for Standardization

Case postale 56 • CH-1211 Genève 20 • Switzerland

Printed in Switzerland

Identification cards – Integrated circuit(s) cards with contacts –

Part 3 :

Electronic signals and transmission protocols

AMENDMENT 1 : Protocol type T = 1, asynchronous half duplex block transmission protocol

Replace the existing clause 9 and add a new annex A as follows:

IECNORM.COM : Click to view the full PDF of ISO/IEC 7816-3:1989/Amd.1:1992

9 Protocol type T=1, asynchronous half duplex block transmission protocol

This transmission protocol is defined as the protocol type T=1 in a TD₁ byte of the Answer-to-Reset (see 6.1.4.3). This clause defines the structure and processing of commands for transmission control and for IC card specific control in an asynchronous half duplex block transmission protocol. These commands may be initiated either by the interface device or the card.

The main characteristics of the block transmission protocol are the following:

- a) A block is the smallest data unit which can be transmitted between an IC card (ICC) and an interface device (IFD).

A block may be used to convey

- (1) application data which is transparent to the transmission protocol;
- (2) transmission control data including transmission error handling, (see 9.1).

- b) The definition of the block structure allows the correct reception of the whole block to be checked before processing the data conveyed by the block.
- c) The identification of a block (recognizing the start and the end of a block) is handled in the character component of the data link layer.
- d) The protocol starts either after the answer to reset or the protocol type selection (PTS) (clause 7) with a first block sent by the IFD and continues with alternating right to send a block.

This protocol uses the character frame defined in the answer to reset and the physical parameters defined by the global interface bytes (see 6.1.4.4) unless modified by the protocol type selection. This clause covers the frame structure of a block and organization of

- Data transmission control such as flow control, block chaining, and error correction
- Specific interface control

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

- The physical layer exchanges bits according to 6.1
- The data link layer is defined by the character component and block component. The character component exchanges characters according to 6.1.2. The character repetition and error signalling described in 6.1.3 shall not be used, so that the minimum delay between two consecutive characters may be reduced to 11 etu, according to the interface character TC1. The block component exchanges blocks defined in 9.6.2.
- The application layer processes commands, which involves the exchange of at least one block or chain of blocks in each direction.

9.1 Definitions and abbreviations

9.1.1 Definitions

For the purposes of this part of ISO/IEC 7816, the following definitions apply to clause 9.

9.1.1.1 block: A succession of characters comprising two or three fields defined as prologue field, information field and epilogue field.

9.1.1.2 destination node address; DAD: The portion of the subfield node address (NAD) which is used to identify the intended receiver of the block.

9.1.1.3 epilogue field: The final field of a block. It contains the error detection code (EDC) byte(s).

9.1.1.4 error detection code; EDC: The result of an error checking method, applied to all characters in the prologue field and in the information field. It is transmitted in the epilogue field.

9.1.1.5 field: One of the three components of the block defined as prologue, information or epilogue.

9.1.1.6 information block; I-block: A block whose primary purpose is to transmit application layer information.

9.1.1.7 information field; INF: The field of a block which contains data (generally application data).

9.1.1.8 length; LEN: A subfield of the prologue field. It contains the number of bytes transmitted in the information field of the block.

9.1.1.9 node address; NAD: A subfield of the prologue field. It indicates both destination and source node addresses of the block and VPP state control.

9.1.1.10 prologue field: The first field of a block. It contains subfields for node address (NAD), protocol control byte (PCB) and length (LEN).

9.1.1.11 protocol control byte; PCB: A subfield of the prologue field. It contains transmission control information.

9.1.1.12 receive ready block; R-block: A block which contains positive or negative acknowledgements. It contains the block number of the expected I-block.

9.1.1.13 source node address; SAD: The portion of the subfield node address (NAD) which is used to identify the sender of the block.

9.1.1.14 subfield: A functional component of a field.

9.1.1.15 supervisory block; S-block: A block which contains transmission control information.

9.1.1.16 transmission control: A function used to control the data transmission between the interface device (IFD) and the IC card (ICC). It includes VPP state control, block transmission with sequence control, synchronization, and recovery of transmission errors.

9.1.2 Abbreviations

BGT	Block guardtime	IFSI	Information field size integer
BWI	Block waiting time integer	INF	Information field
BWT	Block waiting time	LEN	Length
CRC	Cyclic redundancy check	LRC	Longitudinal redundancy check
CWI	Character waiting time integer	NAD	Node address
CWT	Character waiting time	OSI	Open systems interconnection
DAD	Destination node address	PCB	Protocol control byte
EDC	Error detection code	R-block	Receive ready block
I-block	Information block	R	Receive ready
ICC	Integrated circuit(s) card, IC card	SAD	Source node address
IFD	Interface device	S-block	Supervisory block
IFS	Information field size	WTX	Waiting time extension
IFSC	Information field size for the card	XOR	Exclusive-OR
IFSD	Information field size for the interface device		

9.2 Character frame

The character frame is as defined for the answer to reset in 6.1.2 and 6.1.4.4 (last paragraph) unless modified by PTS in 7.2 (5th paragraph).

Parity checking may be used in addition to EDC (see 9.4.3) to determine the validity of a block.

9.3 Block frame

A block is a series of characters (9.2) containing data bytes (as defined in 6.1.2). A block consists of the following fields

- prologue field (mandatory)
- information field (optional)
- epilogue field (mandatory)

as shown in figure 9.

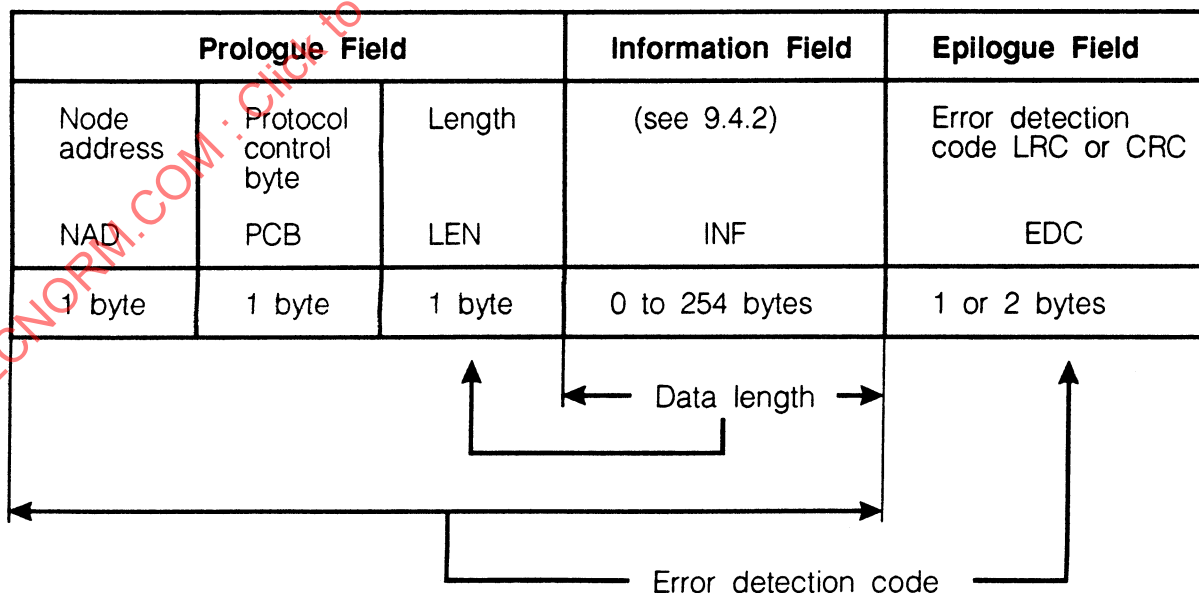


Figure 9 - Block Structure

9.4 Basic elements of the block

9.4.1 Prologue field

This field is mandatory and consists of three bytes: the node address, the protocol control byte and the length.

9.4.1.1 Node address (NAD)

The node address (NAD) is a byte used to identify the source and the intended destination of the block. NAD may be used to distinguish between multiple logical connections when they coexist.

The bits b1 to b3 indicate the source node address SAD and the bits b5 to b7 indicate the destination node address DAD. The bits b4 and b8 are used for VPP state control (see 9.6.1.1).

The NAD of the first block sent by the IFD shall establish the association of the addresses SAD and DAD by which a logical connection is defined. Subsequent blocks in which the NAD field contains the same SAD/DAD address pair are associated with the same logical connection. Other logical connections during the subsequent information exchange may be similarly defined by other SAD/DAD pairs.

NOTE - For example, blocks sent by the IFD with the values x_D as SAD and y_D as DAD and blocks sent by the ICC with the values $x_C=y_D$ and $y_C=x_D$ belong to one logical connection - denoted by (x,y) -, whereas blocks sent by the IFD with v_D as SAD and w_D as DAD and blocks sent by the ICC with the values $v_C=w_D$ and $w_C=v_D$ belong to a second logical connection (v,w).

When the addressing is not used, the values of SAD and DAD are set to 0. Any other node addresses where SAD and DAD are identical are reserved for future use.

9.4.1.2 Protocol control byte (PCB)

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

There are three fundamental types of blocks defined by the protocol:

- An information block (I-block) is used to convey information for use by the application layer. In addition, it conveys a positive or negative acknowledgement.
- A receive ready block (R-block) is used to convey positive or negative acknowledgements; its information field shall not be present.
- A supervisory block (S-block) is used to exchange control information between the IFD and the ICC; its information field may be present depending on its controlling function.

NOTE - This separation allows the design of the protocol control and the application portions of the device microcode to be relatively independent of one another.

The PCB defines whether a block is an I-block, an R-block, or an S-block. For PCB coding details see 9.6.2.4.

9.4.1.3 Length (LEN)

LEN indicates the number of bytes transmitted in the information field of the block (see also 9.5.1).

The coding shall be

'00' to 'FE' codes the number of bytes in the information field from 0 to 254.

'FF' is reserved for future use.

9.4.2 Information field (INF)

The presence of INF is optional. When present, INF conveys either application data in I-blocks or non application control and status information in S-blocks. The number of bytes transmitted is indicated by LEN.

9.4.3 Epilogue field

This field is mandatory. It contains the error detection code (EDC) of the transmitted block.

The protocol definition permits this to be either an LRC (longitudinal redundancy check), or a CRC (cyclic redundancy check). The LRC is one byte in length, while the CRC is two bytes. The LRC is calculated as the exclusive OR (XOR) of all the bytes starting with the NAD through the last byte of the information field, and is typically referred to simply as the checksum. For CRC see ISO 3309.

9.5 Specific interface parameters

In the Answer-to-Reset, the specific interface characters for T=1 are indicated starting with TA_i (i>2).

9.5.1 Information field sizes (IFS)

9.5.1.1 Information field size for the card (IFSC)

IFSC is the maximum length of information field of blocks which can be received by the card. The initial value of IFSC is given by IFSI in the specific interface character TA_i (i>2). The default value is 32.

The coding shall be

'00' is reserved for future use.
'01' to 'FE' codes values for IFSC from 1 to 254.
'FF' is reserved for future use.

NOTE - The block size is the total number of all bytes transmitted in the prologue, information, and epilogue fields. The maximum block size is equal to IFSC plus 4 or 5 (depending upon the length of the epilogue field).

9.5.1.2 Information field size for the interface device (IFSD)

IFSD is the maximum length of the information field for blocks which can be received by the interface device. The initial value is defined as 32.

The coding shall be as for IFSC in 9.5.1.1.

9.5.2 Waiting times

9.5.2.1 Character waiting time (CWT)

The character waiting time (CWT) is defined as the maximum time between the leading edges of two consecutive characters in the same block. See figure 10.

The least significant half byte (b4 to b1) of TB_i ($i \geq 2$) codes the character waiting time integer CWI so that CWT is calculated by

$$CWT = (2^{CWI} + 11) \text{ work etu}$$

Therefore the minimum value of CWT is equal to 12 work etu. See 6.1.4.4 definition of work etu.

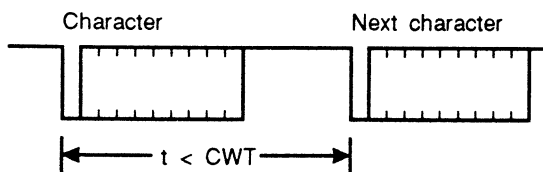


Figure 10 - Character waiting time (CWT)

The default value of CWI is 13.

NOTE - When there is a potential error in the length, the CWT may be used by the receiving node to detect the end of a block.

9.5.2.2 Block waiting time (BWT)

The Block waiting time (BWT) is defined as the maximum time between the leading edges of the last character which gave the right to send to the card and the first character to be sent by the card. See figure 11.

The most significant half byte (b8 to b5) of TB_i ($i \geq 2$) codes the block waiting time integer BWI so that BWT is calculated by

$$BWT = 2^{BWI} \times 960 \times 372 / f_s \text{ s} + 11 \text{ work etu}$$

for an external clock card

$$BWT = 2^{BWI} / 10 \text{ s} + 11 \text{ work etu}$$

for an internal clock card

where $0 \leq BWI \leq 9$ and $BWI > 9$ is reserved for future use.

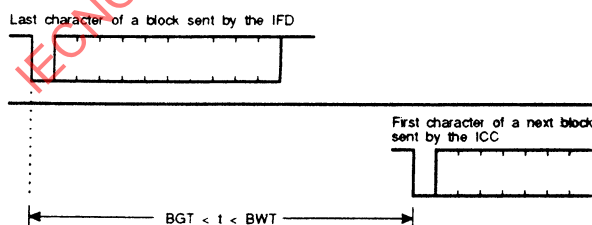


Figure 11 - Block guardtime (BGT) and block waiting time (BWT)

The default value of BWI is 4.

The block waiting time is used to detect an unresponsive card.

9.5.2.3 Block guardtime (BGT)

The block guardtime (BGT) is defined as the minimum time between the leading edges of two consecutive characters sent in opposite directions. Consequently the delay between the last character of a received block and the first character of a transmitted block shall be at least BGT but less than BWT. See figure 11.

The value of BGT shall be 22 work etu.

9.5.3 Indication of protocol options

The specific interface character TC_i ($i \geq 2$) indicates the form of the error detection code by

bit b1 = 1 Use of CRC

= 0 Use of LRC (default value)

Bits b2 to b8 are reserved for future use and shall be set to 0.

9.6 Protocol operation

9.6.1 Data link layer - Character component

9.6.1.1 VPP control

The state of VPP is controlled by bits b8 and b4 of the NAD byte sent by the card, and by the PCB character which immediately follows. Bits b8 and b4 of NAD designate

b8=0, b4=0 requires VPP to be returned to or kept in idle state.

b8=1, b4=0 requires setting the active state of VPP. VPP is returned to idle state on receipt of the PCB.

b8=0, b4=1 requires setting the active state of VPP until reception of another NAD byte by the interface device.

b8=1, b4=1 is forbidden.

If a parity error occurs on the NAD, VPP is returned to or kept in idle state.

If a time-out occurs, i.e. if the card fails to send an expected character within CWT or BWT, VPP is returned to or kept in idle state.

All transitions on VPP triggered by a character shall occur within 12 etus from the leading edge of this character.

9.6.1.2 Error-free operation

After answer to reset with or without protocol type selection, the interface device has the right to send. This is considered to be the beginning of the block transmission protocol. When protocol T=1 has been designated an interface device sends only blocks.

When either the ICC or IFD has sent a complete block, it switches to receiving state. When either the ICC or IFD has received the number of characters in accordance with the length subfield, it assumes that it has the right to send.

9.6.2 Data link layer - Block component

In the procedure description the following notation will be used.

9.6.2.1 Notation (see 9.4.1.2)

The following notations are used to identify terms used in the descriptions of how the protocol functions.

I-blocks are denoted by

$I(N(S),M)$ $N(S)$ is the send sequence number and M is the more data bit (see 9.6.2.2.2).

$N_a(S), N_b(S)$ To distinguish the sequence numbers sent by sources A or B the indices a and b are added to $N(S)$.

R-blocks are denoted by

$R(N(R))$ R-block indicating receive ready (R), and $N(R)$. $N(R)$ is the block number of the next message the receiver expects to receive.

S-blocks are denoted by

$S(\text{RESYNCH request})$ S-block indicating RESYNCH request

$S(\text{RESYNCH response})$ S-block indicating RESYNCH response

$S(\text{IFS request})$ S-block indicating IFS offer

$S(\text{IFS response})$ S-block indicating IFS offer acknowledgement

$S(\text{ABORT request})$ S-block indicating ABORT request

$S(\text{ABORT response})$ S-block indicating ABORT response

$S(\text{WTX request})$ S-block indicating WTX request

$S(\text{WTX response})$ S-block indicating WTX response

$S(\text{Vpp error response})$ S-block sent by the IFD to inform the card of a Vpp error

The S-blocks $S(\text{IFS}...)$ and $S(\text{WTX}...)$ contain INF, whose coding is defined by rules 3 and 4 in 9.6.2.2.3.

9.6.2.2 Error-free operation

9.6.2.2.1 General procedures (see 9.4.1.2)

The first block transmitted by the IFD to the ICC after the answer to reset or after the protocol type selection, shall be either an I-block or an S-block.

After a block (I-, R- or S-block) has been sent, an acknowledgement shall be received before start of the transmission of the next block, as described in the following.

An I-block carries its send sequence number $N(S)$. $N(S)$ consists of one bit and is counted modulo 2. Its value starts with $N(S)=0$ either after the start of the block transmission protocol or after resynchronization and is incremented after sending each I-block.

The numbers $N(S)$ of I-blocks sent by the IFD and of those sent by the ICC are counted independently from each other. Acknowledgement of a transmitted I-block is indicated when $N(S)$ of the next received I-block is different from $N(S)$ of the previously received I-block.

An R-block carries $N(R)$, which is the value of $N(S)$ in the next expected I-block. Acknowledgement of an I-block is indicated when the $N(R)$ of the received next R-block is different from the $N(S)$ of the sent I-block (see 9.6.2.2.3, Rule 2.2).

In errorfree operation R-blocks are used for chaining (see 9.6.2.2.2).

An S-block carries no number. An $S(... \text{ request})$ -block carries no acknowledgement. An $S(... \text{ response})$ -block acknowledges the received $S(... \text{ request})$ -block.

9.6.2.2.2 Chaining

Chaining is a function of the block transmission protocol. It allows the IFD or ICC to transmit information (application data) longer than IFSC or IFSD.

If the IFD or ICC has to transmit information longer than IFSC or IFSD respectively, it should divide the information into blocks, each with LEN less than or equal to IFSC or IFSD, and send the multiple blocks using chaining function.

The chaining of blocks is controlled by the M-bit ("More data bit") in the PCB of an I-block. The M-bit indicates two states of an I-block:

0 = last block of chain

1 = chained data follows in subsequent block(s)

Figure 12 illustrates the chaining function.

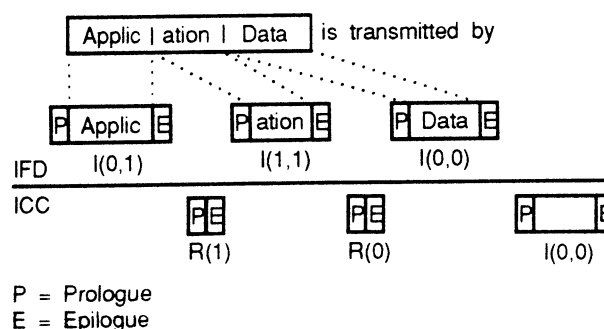


Figure 12 - Chaining function

When the receiver receives a More data I-block correctly, it shall send $R(N(R))$, where $N(R)$ is equal to $N(S)$ of the next I-block.

NOTE - I-blocks with LEN=0 may be used within a chain (see scenario in A.1.4.3).

9.6.2.2.3 Protocol rules for the error-free operation

Rule 1: The interface device sends the first block: either an I-block with $N(S)=0$ and M (More data bit), designated as I(0,M), or an S-block.

Rule 2.1: I($N_a(S)$,0) sent by A

is acknowledged by

I($N_b(S)$,M) sent by B

to transmit application data and to indicate readiness to receive the next I-block from A.

Rule 2.2: I($N_a(S)$,1) sent by A

is acknowledged by

R($N_b(R)$) sent by B [where $N_b(R)$ is not equal to $N_a(S)$]

to indicate that the received block was correct and the readiness to receive the next I-block from A.

NOTE - Chaining is only possible in one direction at a time.

Rule 3: If the ICC requires more than BWT to process the previously received I-block, it sends S(WTX request) where INF is a one byte binary integer multiplier of the BWT value. The IFD acknowledges by S(WTX response) with the same INF.

The time allocated starts at the leading edge of the last character of the S(WTX response) block.

Rule 4: The ICC sends S(IFS request) to indicate a new IFSC it can support and this shall be acknowledged by S(IFS response) with the same INF. The IFD assumes the new IFSC is valid as long as no other IFSC is indicated by another S(IFS request).

The IFD sends S(IFS request) to indicate a new IFSD it can support and this shall be acknowledged by S(IFS response) with the same INF. The ICC assumes the new IFSD is valid as long as no other IFSD is indicated by another S(IFS request).

IFSC and IFSD are each coded as a byte binary integer in the INF of these S-blocks.

Rule 5: Chaining is indicated by the M-bit, where I($N(S)$,0) is a non chained block or the last block of a chain. I($N(S)$,1) is part of a chain and will be followed by at least one chained block.

R($N(R)$) requests transmission of the next chained I-block I($N(S)=N(R)$,...) and acknowledges the received chained I-block I(NOT $N(R)$,1).

9.6.2.3 Error handling

9.6.2.3.1 Errors detected by the receiver of a block

The tasks of the block layer are to transmit blocks, to detect transmission and sequence errors, to handle such errors and to resynchronize the block transmission protocol. Therefore the protocol of the block layer should be able to handle the following errors

- BWT timeout: The delay between the leading edge of the last character of a block and the leading edge of the first character of the next block exceeds BWT.

- Reception of an invalid block. Examples are

- a) a parity error in one or more characters of a block, or an EDC error
- b) invalid PCB (due to unknown encoding);
- c) invalid LEN (transmission error or incompatibility with IFSC or IFSD);
- d) loss of synchronization (either an underrun when the receiver expects more than the number of received characters or an overrun when the sender sends more characters than the value indicated in the received length field);
- e) failure to receive S(... response) after having sent the relevant S(... request).

Resynchronization in this protocol is attempted at three consecutive levels. If one level is unsuccessful, then the next level is tried.

The three synchronization levels are

for the IFD

- a) Retransmission of blocks
- b) Use of S(RESYNCH request)
- c) Resetting or deactivation of the ICC

for the ICC

- a) Retransmission of blocks
- b) Use of S(RESYNCH response)
- c) Without action by the IFD, the card becomes unresponsive

9.6.2.3.2 Protocol rules for the error handling

Rule 6: S(RESYNCH request) may be sent only by the IFD to reach resynchronization and to initiate resetting the communication parameters of the block transmission protocol to its initial values.

Rule 6.1: If a loss of synchronization is detected by the receiver, the receiver gets back the right to send after a silence on I/O greater than the larger of CWT or BGT.

Rule 6.2: S(RESYNCH request) shall be responded to by S(RESYNCH response) from the ICC.

Rule 6.3: After the IFD has received S(RESYNCH response), the protocol is initiated.

Rule 6.4: After the IFD has failed a maximum of three times in succession to reach the intended resynchronization by sending S(RESYNCH request), it resets the ICC.

Rule 6.5: When an S(RESYNCH request) is received, the previously sent block is assumed not to have been received.

Rule 7.1: When an I-block was sent and an invalid block is received or a BWT timeout (with the IFD) occurs, an R-block is sent, which requests with its N(R) for the expected I-block with $N(S)=N(R)$.

Rule 7.2: When an R-block was sent and an invalid block is received or a BWT timeout (with the IFD) occurs, this R-block is retransmitted.

Rule 7.3: When an S(... request)-block was sent and the received answer is not S(... response)-block or a BWT timeout occurs (only IFD), the S(... request)-block is retransmitted.

When an S(... response)-block was sent and an invalid block is received or a BWT timeout occurs (only IFD), an R-block is sent.

Rule 7.4.1: After failing to receive an error free block at the beginning of the protocol, the IFD makes a maximum of two further attempts in succession before resetting or deactivating the ICC.

Rule 7.4.2: During the protocol, if the IFD fails to receive an error free block it makes a maximum of two further attempts in succession before sending S(RESYNCH request).

Rule 7.4.3: If the ICC fails to receive an error free block after a second attempt in succession, it remains in receive mode.

Rule 7.5: After the beginning of the protocol the ICC reacts on receiving an invalid first block by sending R(0).

Rule 7.6: If the first block sent by the IFD is not responded to within BWT, the IFD sends R(0).

Rule 8: When the ICC sends S(IFS request) and receives an invalid block, it retransmits a maximum of one more S(IFS request) block in order to elicit an S(IFS response). After the second failure it remains in receive mode.

Rule 9: The abortion of a chain can be initiated by either the sender or receiver of a chain sending an S(ABORT request).

The S(ABORT request) shall be answered by an S(ABORT response), after which an R-block may be sent depending on whether it is necessary to give back the right to send.

NOTE - Abortion of chaining may be due to physical error in the ICC, such as ICC memory error.

9.6.2.4 Coding of PCB

9.6.2.4.1 PCB of I-blocks

In all I-blocks the most significant bit (b8) of the PCB is set to zero.

The other 7 bits are used as shown in figure 13.

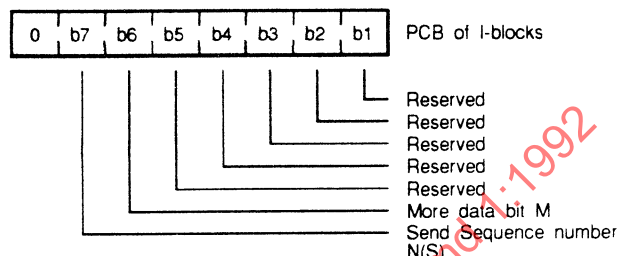


Figure 13 - Coding of I-block PCB

Bits b1 to b5 are reserved for future use and shall be set to zero.

9.6.2.4.2 PCB of R-blocks

In all R-blocks the most significant bit (b8) of PCB is set to one and the bit b7 of PCB is set to zero.

The other 6 bits are used as shown in figure 14.

1	0	b6	b5	b4	b3	b2	b1	PCB of R-blocks
0	N(R)	0	0	0	0	0	0	Errorfree
0	N(R)	0	0	0	0	1	0	EDC or parity error
0	N(R)	0	0	0	1	0	0	Other errors

(all other values reserved)

Figure 14 - Coding of R-block PCB

NOTE - The value of N(R) states whether the R-block is an error indicating block or not. Bits b1 to b4 may optionally be evaluated.

9.6.2.4.3 PCB of S-blocks

In all S-blocks both the most significant bits (b8 and b7) of PCB are set to one.

The other 6 bits are used as shown in figure 15.

1	1	b6	b5	b4	b3	b2	b1	PCB of S-blocks (b6 is the RESPONSE bit)
0	0	0	0	0	0	0	0	RESYNCH request
1	0	0	0	0	0	0	0	RESYNCH response
0	0	0	0	0	0	1	1	IFS request
1	0	0	0	0	0	1	1	IFS response
0	0	0	0	0	1	0	0	ABORT request
1	0	0	0	0	1	0	0	ABORT response
0	0	0	0	0	1	1	1	WTX request
1	0	0	0	0	1	1	1	WTX response
1	0	0	1	0	0	0	0	Vpp error response

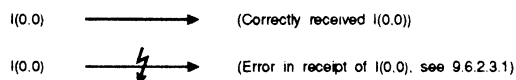
(all other values reserved)

Figure 15 - Coding of S-block PCB

Annex A (informative) Scenarios

NOTE - See 9.6.2.1 for notation

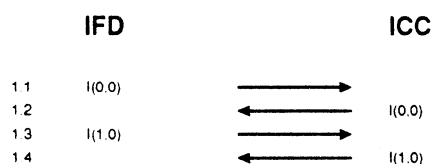
Additional examples of notation:



A.1 Error-free operation

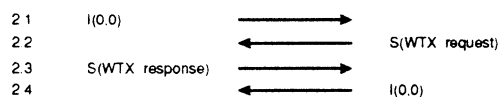
A.1.1 Exchange of I-blocks (see 9.6.2.2.3, Rule 1, Rule 2.1)

Scenario 1:



A.1.2 Request for waiting time extension (see 9.6.2.2.3, Rule 3)

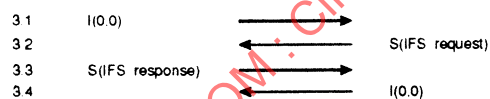
Scenario 2:



A.1.3 IFS adjustment (see 9.6.2.2.3, Rule 4)

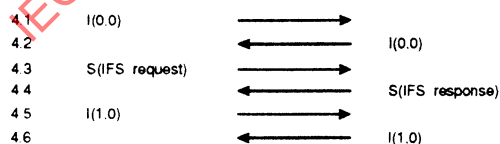
A.1.3.1 ICC is initiator

Scenario 3:



A.1.3.2 IFD is initiator

Scenario 4:



A.1.4 Chaining function (see 9.6.2.2.3, Rule 2.2, Rule 5)

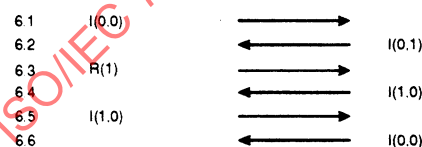
A.1.4.1 IFD sends chained I-blocks

Scenario 5:



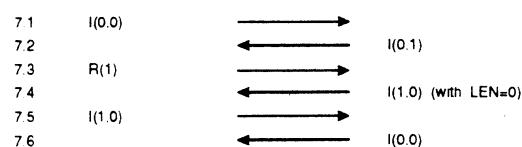
A.1.4.2 ICC sends chained I-blocks

Scenario 6:



A.1.4.3 ICC uses the M-bit to force an acknowledgement for a sent I-block (see NOTE at the end of 9.6.2.2.2)

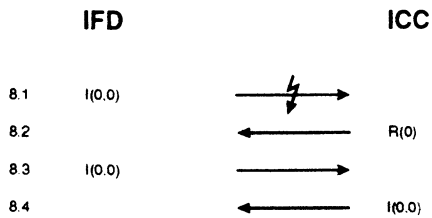
Scenario 7:



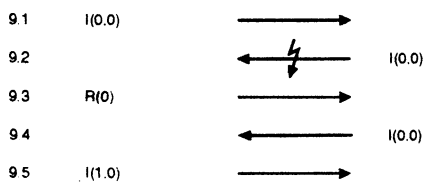
A.2 Error handling

A.2.1 Exchange of I-blocks

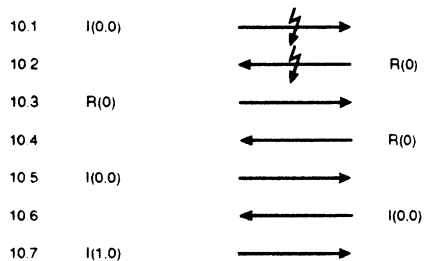
Scenario 8 (Starting protocol):
(see 9.6.2.3.2, Rule 7.5)



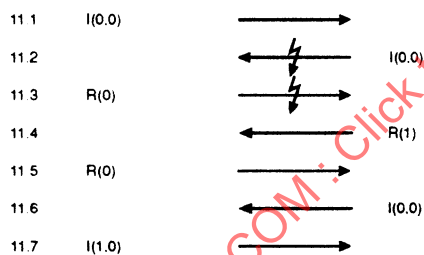
Scenario 9 (see 9.6.2.3.2, Rule 7.1, Rule 7.6):



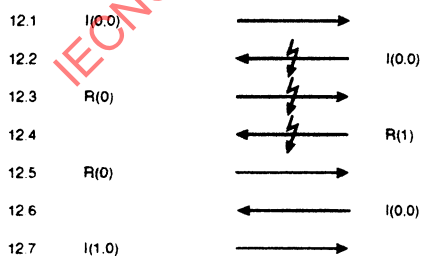
Scenario 10: (see 9.6.2.3.2, Rule 7.1, Rule 7.5, Rule 7.6)



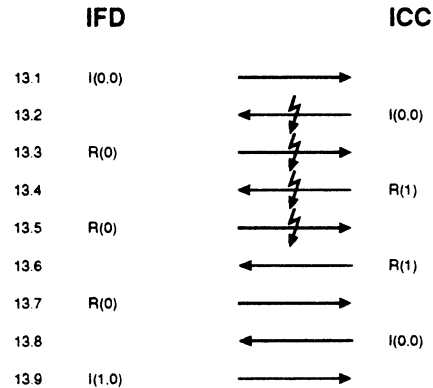
Scenario 11: (see 9.6.2.3.2, Rule 7.1, Rule 7.6)



Scenario 12: (see 9.6.2.3.2, Rule 7.1, Rule 7.2, Rule 7.6)

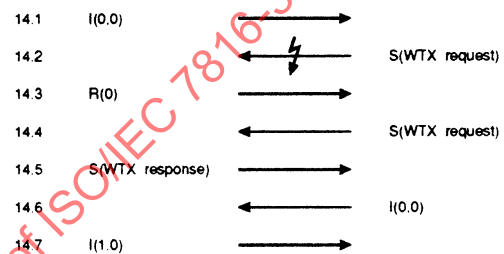


Scenario 13: (see 9.6.2.3.2, Rule 7.1, Rule 7.2, Rule 7.6)

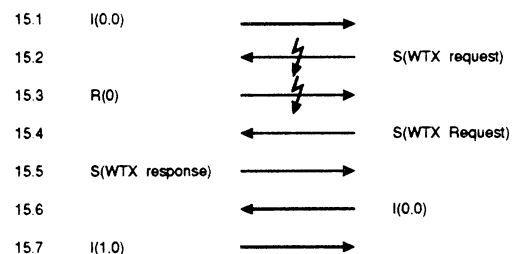


A.2.2 Request for waiting time extension (see 9.6.2.3.2, Rule 7.3)

Scenario 14:



Scenario 15:



A.2.3 Request for IFSI adjustment (see 9.6.2.3.2, Rule 7.3)

Scenario 16:

