INTERNATIONAL STANDARD

ISO 16845

First edition
2004-03-15

Road vehicles — Controller area network (CAN) — Conformance test plan

Véhicules routiers — Gestionnaire de réseau de communication (CAN) — Plan d'essai de conformité

Reference number
ISO 16845:2004(E)

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

Contents

Page

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

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

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

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

ISO 16845 was prepared by Technical Committee ISO/TC 22, *Road vehicles*, Subcommittee SC 3, *Electrical and electronic equipment*.

Road vehicles — Controller area network (CAN) — Conformance test plan

1 Scope

This International Standard provides the methodology and abstract test suite necessary for checking the conformance of any CAN implementation of the CAN specified in ISO 11898-1.

2 Normative references

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

ISO/IEC 9646-1:1994, *Information technology — Open Systems interconnection — Conformance testing methodology and framework — Part 1: General concepts*

ISO 11898-1:2003, *Road vehicles — Controller area network (CAN) — Part 1: Data link layer and physical signalling*

ISO 11898-2:2003, *Road vehicles — Controller area network (CAN) — Part 2: High-speed medium access unit*

3 Terms and definitions

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

3.1
ACK delimiter
second bit of the ACK field

3.2
ACK field
last field before the EOF, used for message validation

3.3
acknowledgement error
error condition of the transmitter when it does not detect a dominant bit on the ACK slot

3.4
ACK slot
first bit of the ACK field

3.5
active error flag
first field of an active error frame

3.6
active error frame
error frame that starts with an active (dominant) error flag

3.7
active state
state of a node when it can transmit an active error frame

3.8
arbitration field
field starting after the SOF bit and finished with the RTR bit

3.9
bit error
error condition encountered when the received bit does not correspond to the transmitted or expected bit

3.10
conformance testing
application of the test plan to an IUT

3.11
CRC delimiter
last bit of the CRC field

3.12
CRC error
error condition of a receiver when the received CRC code does not match the calculated CRC code

3.13
CRC field
field preceding the ACK field, consisting of the CRC code and the CRC delimiter

3.14
end of frame
last field of a data or remote frame before the intermission field

3.15
error delimiter
second field of an error frame

3.16
error flag
first field of an error frame

3.17
error frame
formatted sequence of bits indicating an error condition

3.18
form error
error condition encountered in a fixed form field

3.19
idle state
CAN bus state where no frame is started after intermission field

3.20
intermission field
field after EOF, error delimiter, or overload delimiter

3.21
lower tester
tester that supervises the test suite

3.22
overload delimiter
second field of an overload frame

3.23
overload flag
first field of an overload frame

3.24
overload frame
formatted sequence of bits indicating an overload condition

3.25
passive error flag
first part of a passive error frame

3.26
passive state
state of the device when the value of the REC or the TEC has reached the error passive limit

3.27
REC passive state
state of the device when the value of the REC has reached the error passive limit

3.28
recessive state
state of the CAN bus when no CAN node drives a dominant value on the line

3.29
stuff bit
bit inserted into the bit stream to increase the number of edges for synchronization purpose

3.30
stuff error
error condition encountered when an expected stuff bit is missing

3.31
suspend transmission field
waiting time added after the intermission field for an error passive transmitter, before it can start another transmission

3.32
TEC passive state
state of the device when the value of the TEC has reached the error passive limit

3.33
test case
specificly numbered and named test in the test suite

3.34
test frame
CAN frame containing the test pattern specified

3.35
test suite
check of the behaviour of the IUT for particular parameters of the CAN specification

3.36
test type
specification of the direction of the test frames

EXAMPLE Specification of the behaviour of the IUT receiving and/or transmitting messages.

3.37
time quantum
elementary time unit of the CAN bit time derived from the oscillator clock and the prescaler

3.38
upper tester
tester that acts as an user of the IUT

4 Abbreviated terms

ACK Acknowledgement

BRP Bit rate prescaler

CAN Controller area network

CRC Cyclic redundancy check

CTRL Control

DLC Data length code

EOF End of frame

DIE Identifier extension bit

IDEN CAN identifier

IPT Information processing time

IUT Implementation under test

LLC Logical link control

LME Layer management entity

LT Lower tester

MAC Medium access control

MDI Medium dependent interface

NDATA Network data

NTQ Number of Time Quanta

PCO Point of control and observation

PLS Physical layer signalling

PMA Physical medium attachment

REC Receive error counter

RTR Remote transmission request

RX Receiver signal

SJW Re-Synchronization jump width

SLIO Serial linked input/output

SOF Start of frame

SRR Substitute remote request

TEC Transmit error counter

TP Test plan

TQ Time quantum

TSYS System clock time (of the IUT)

UT Upper tester

5 General

5.1 Architecture of the test plan (TP)

The architecture of the TP plan is as shown in Figure 1.

The TP is a specific application of ISO 9646-1 and is restricted to the single party testing mode. Since the upper service boundary of a CAN implementation is not standardized and in some cases may not be observed and controlled, because of an application specific behaviour embedded in this implementation, (e.g. CAN SLIO), the TP shall rely either on the coordinated test method or the remote test method.

Depending on the test method applied, the TP shall involve up to three test functions:

— an LT operating in way similar to the CAN IUT, running test suite and granting test verdict;

— an UT acting as user of the IUT (IUT-dependant);

— a test management protocol between the IUT and the LT, consisting of test coordination procedures.

The last two functions are only applicable to the coordinated test procedure.

During testing, the LT may observe and control the standardized lower service boundary of the IUT (PCO) through the two service primitives provided by the PLS sub-layer, i.e. PLS Data.indicate and PLS Data.request, in most cases.

The environment that implements the TP is described in Figure 2.

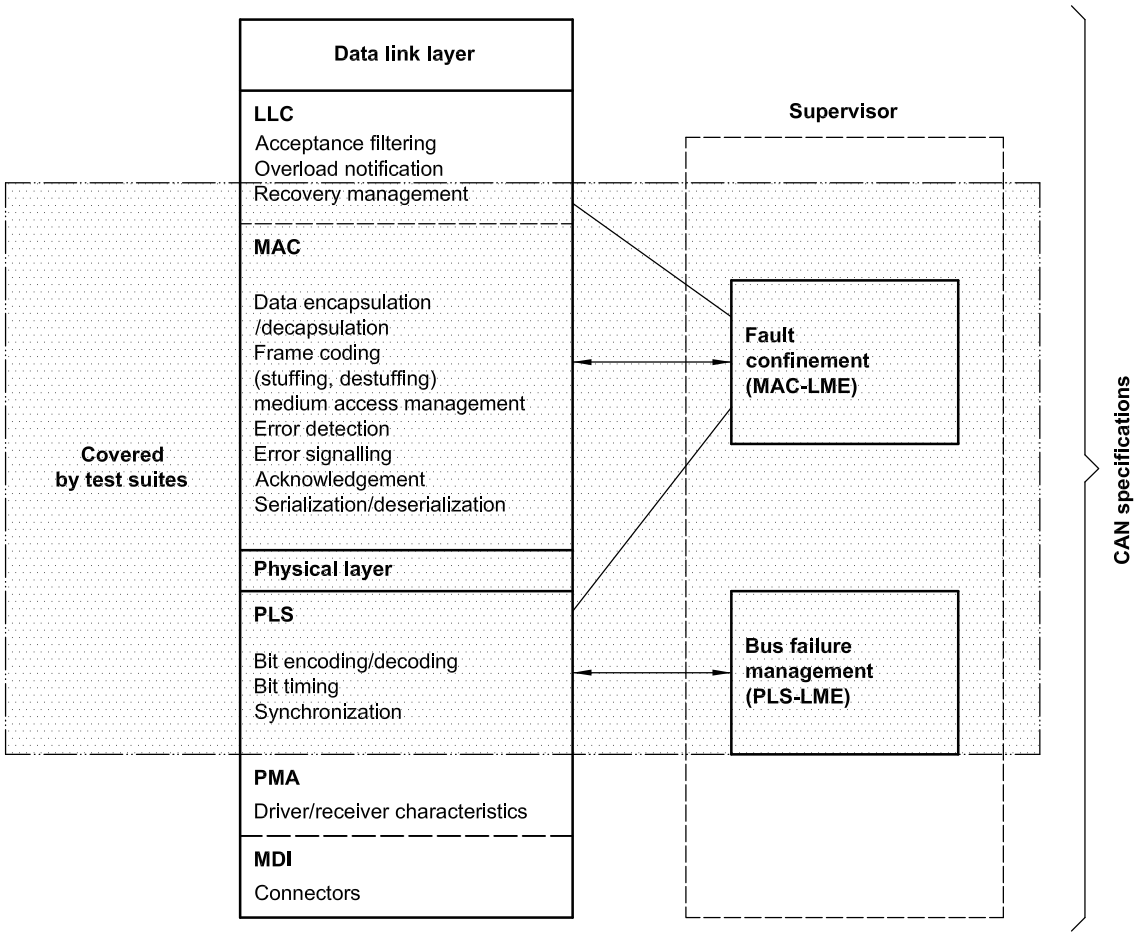

Figure 1 — Architecture of the test plan

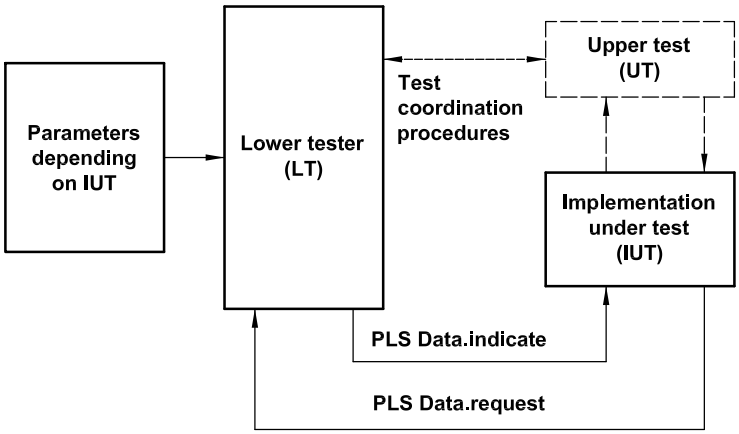

Figure 2 — CAN conformance TP environment

Using the network interface, the LT indicates to the UT the actions to be performed and the UT provides the LT with information concerning the internal behaviour of the IUT.

In order to allow the LT and the UT to communicate, it is necessary to specify some test coordination procedures between them. These procedures use the network to the exclusion of any other physical link. They are used to set up the UT and to verify the test results.

5.2 TP organization

5.2.1 General organization

The LT verifies if the IUT complies with the MAC, LLC, and PLS sub-layers of the CAN specification. The LT points out differences between what is expected according to the specification and the actual behaviour of the IUT.

The test suites of the TP are independent of one another. Each test suite may be used to check the behaviour of the IUT for a particular parameter of the CAN specification. Tests may be performed in sequence or separately.

Tests requiring variations of individual parameters (identifier, number of data, etc.) shall be repeated for each value of the parameter, these repetitions being known as elementary tests. A test including different elementary tests is valid only if all tests are passed.

5.2.2 Test organization

5.2.2.1 Elementary tests

5.2.2.1.1 Description

Each elementary test shall consist of three states:

— set-up;

— test;

— verification.

At the PCO, these states involve interchanges of valid sequences of PLS service primitives [CAN frame(s)] or invalid sequences of PLS primitives (invalid CAN frames or noise).

Before the first elementary test is started the IUT shall be initialized into the default state.

5.2.2.1.2 Set-up state

The set-up state is the state the IUT shall be in before entering the test state.

5.2.2.1.3 Test state

The test state is the part of the elementary test in which the parameter or protocol feature is checked. This state needs one or several interchanges of frames, called test frames.

5.2.2.1.4 Verification state

The verification state is made up of the data-reading frames, which verify that the data have been handled in accordance with the CAN specification.

For tests belonging to Classes 1 to 6 according to 5.3.3, the LT shall be able to detect the correct value of the bit.

For bit timing tests (Class 7 according to 5.3.3), the LT shall be able to detect a faulty synchronization of one time quantum.

5.2.2.2 Default state

The default state is characterized by the following default values:

— both REC and TEC shall be equal to 0;

— no pending transmission shall be present;

— IUT shall be in idle state;

— PLS data.indicate and PLS data.request shall be recessive.

After the end of each elementary test, the default state shall be applied.

5.3 Hierarchical structure of tests

5.3.1 Overview

The tests are grouped in categories in order to aid planning, development, understanding or execution of each test. Three levels of categories are specified test types, test classes and test cases.

5.3.2 Test types

Test types specify the direction of the frames. There are three types:

— **Type 1, received frame,** includes all tests evaluating the behaviour of the IUT for data frames and remote frames received by the IUT;

— **Type 2, transmitted frame,** includes all tests evaluating the behaviour of the IUT for data frames and remote frames transmitted by the IUT;

— **Type 3, bi-directional frame,** includes all tests with data frames or remote frames both received and transmitted by the IUT.

5.3.3 Test classes

Each of the three test types given in 5.3.2 is divided into seven classes, grouping tests:

— **Class 1, valid frame format,** includes the tests involving only error free data or remote frames;

— **Class 2, error detection,** includes the tests that corrupt data or remote frames, which are used to check correct error detection by the IUT;

— **Class 3, active error frame management,** includes the tests verifying the IUT correct management of error-free and of corrupted active error frames;

— **Class 4, overload frame management,** includes the tests verifying the IUT correct management of error-free and of corrupted overload frames;

— **Class 5, passive error state and bus-off,** includes the tests verifying the IUT behaviour during passive error state and bus-off state;

— **Class 6, error counter management,** includes the tests verifying the correct management of the TEC and REC by the IUT in both active and passive error state;

— **Class 7, bit timing,** includes the tests verifying the correct management of bit timing by the IUT, and shall be applied only to those components performing recessive to dominant edge synchronization — if the dominant to recessive edge synchronization exists, it shall be disabled.

5.3.4 Test cases

Each and every basic entry of the test list is intended for checking a particular parameter of the harmonized CAN specification in the IUT.

Each test case is specified by a number and a particular name in order to differentiate the test cases and to easily summarize the goal of the test case. Some test cases may be subdivided into elementary tests that are repetitions of the test case for several values of the parameter tested.

6 LT parameters

6.1 Overview

The CAN specification allows several IUT implementations. Consequently, the LT shall be provided with parameters in order to indicate which kind of IUT is to be tested. These parameters are classified in two categories:

— **communication parameters,** specifying which tests can be executed for the IUT, and which test method shall be applied;

— **application parameters,** which specifies the features of the frames used for each test case selected according to the previous parameters.

NOTE LT applies to IUT performing only recessive to dominant edge synchronization and operating in single sampling mode.

6.2 Description of parameters

6.2.1 Communication parameters

6.2.1.1 Categories of communication parameter

Communication parameters are subdivided in three categories: implementation, timing and NDATA parameters.

6.2.1.2 Implementation parameters

Implementation parameters dependant on the IUT shall be specified in order to allow the LT to fit on the IUT. These implementation parameters are as follows.

a) **CAN_VERSION** indicates the version implemented in the IUT and may take three values.

A: IUT handes 11 bit identifiers.

B: IUT handles 11 and 29 bit identifiers.

BP: IUT handes 11 identifiers and tolerates 29 bit identifiers.

b) **Open/specific,** which indicates whether the IUT is open regarding the application layers or includes a specific application, and may be of two types.

OPEN: open IUT allowing the test coordination procedure to be implemented in an UT.

These IUT shall be tested with the coordinated test method according to ISO 9646-1.

SPECIFIC: IUT that can be tested only with the help of a specific configuration procedure.

These IUT shall be tested with the remote test method according to ISO 9646-1.

6.2.1.3 Timing parameters

The LT also requires that some timing parameters be in accordance with the IUT and the UT characteristics. These parameters are as follows.

a) Timeout indicates the minimum duration time for which the LT shall wait in order to respect the following three conditions.

— The UT shall have enough time to put the IUT into the set-up state.

— The IUT shall have enough time to transmit a response frame after a remote frame.

— The LT shall consider an optional additional waiting time after the end of the minimum bus-off recovery sequence before the IUT enters error active state again.

b) **TSYS** indicates the duration of the IUT system clock (clock used as input of the prescaler).

c) **BRP** indicates the value of the prescaler (the duration of a TQ is T_Q = TSYS × BRP).

d) **NTQ** indicates the number of time quanta per bit.

e) **Phase_Seg2** indicates the number of time quanta for the phase buffer segment 2.

f) **SJW** indicates the number of time quanta for the re-synchronization jump width. In all tests, the re-synchronization jump width shall be programmed to its full range, up to its maximum value which is the minimum of Phase_Seg1 and 4 TQ.

g) **IPT** indicates the information processing time.

h) **IUT delay time** shall be considered for bit timing class tests. It indicates the time difference between the response of the IUT and the response of an ideal IUT (without internal delays) to an edge causing synchronization. The IUT delay time is the sum of the IUT input and output delay time periods, measured according to ISO 11898-2.

6.2.1.4 NDATA parameter, a set of DLC values which an IUT accepts for data exchange with higher layers.

6.2.2 Application parameters

Except for tests for which a particular profile of application parameters is specified by the TP, the content of the application parameters used during the test shall be chosen by the user.

7 Test type 1, received frame type

7.1 Test class 1, valid frame format

7.1.1 Identifier and number of data in standard format

7.1.1.1 Purpose and limits of test case

This test case is applicable to CAN_VERSION ∈ {A, B, BP}.

It is used to verify the behaviour of the IUT when receiving a correct data frame with different identifiers and different numbers of data bytes in a standard format frame.

Tested identifiers: ∈ [000h, 7EFh] ∪ [7F0h, 7FFh],

Tested number of data bytes: ∈ [0, 8]

7.1.1.2 Test case organization

Test case organization shall be in accordance with Table 1.

Table 1 — Identifier and number of data in standard format — Test case organization

State	Description
Set-up	No action required, the IUT is left in the default state.
Test	A single test frame is used for each elementary test.
Verification	The IUT shall not generate any error flag during the test. The IUT shall acknowledge the test frame. The data received by the IUT during the test state shall match the data sent in the test frame.

7.1.2 Identifier and number of data in extended format — Test case 1

7.1.2.1 Purpose and limits of this test case

This test case is applicable to CAN_VERSION $\in$ {B}.

It is used to verify the behaviour of the IUT when receiving a correct data frame with different identifiers and different numbers of data bytes in a extended format frame.

Tested identifiers: $\in$ [00000000, 1FFFFFFFh]

Tested number of data bytes: $\in$ [0, 8]

7.1.2.2 Test case organization

Test case organization shall be as shown in Table 2.

Table 2 — Identifier and number of data in extended format — Test case 1 organization

State	Description
Set-up	No action required, the IUT is left in the default state.
Test	A single test frame is used for each elementary test.
Verification	The IUT shall not generate any error flag during the test. The IUT shall acknowledge the test frame. The data received by the IUT during the test state shall match the data sent in the test frame.

7.1.3 Identifier and number of data in extended format — Test case 2

7.1.3.1 Purpose and limits of this test case

This test is applicable to CAN_VERSION $\in$ {BP}.

It is used to verify the behaviour of the IUT when receiving a correct data frame with different identifiers and different numbers of data bytes in a extended format frame.

Tested identifiers: $\in$ [00000000, 1FFFFFFFh]

Tested number of data bytes: $\in$ [0, 8]

7.1.3.2 Test case organization

Test case organization shall be in accordance with Table 3.

Table 3 — Identifier and number of data in extended format — Test case 2 organization

State	Description
Set-up	No action required, the IUT is left in the default state.
Test	A single test frame is used for each elementary test.
Verification	The IUT shall not generate any error flag during the test. The IUT shall acknowledge the test frame.

7.1.4 Acceptance of non-nominal r1, r0 combination in standard format

7.1.4.1 Purpose and limits of this test case

This test is applicable to CAN_VERSION $\in$ {A}.

Its purpose is to verify that the IUT accepts the non-nominal value of «r1, r0» bits in a valid standard frame.

Three (3) values shall be tested, as given in Table 4.

Table 4 — Values of r1 and r0 bits

r1	r0
1	1
1	0
0	1

7.1.4.2 Test case organization

Test case organization shall be in accordance with Table 5.

Table 5 — Acceptance of non-nominal r1, r0 combination in standard format — Test case organization

State	Description
Set-up	No action required, the IUT is left in the default state.
Test	A single test frame is used for each of the three elementary tests.
Verification	The IUT shall not generate any error flag during the test. The IUT shall acknowledge the test frame. The data received by the IUT during the test state shall match the data sent in the test frame.

7.1.5 Acceptance of non-nominal IDE, r0 combination in standard format

7.1.5.1 Purpose and limits of test case

This test is applicable to CAN_VERSION $\in$ {B, BP}.

Its purpose is to verify that the IUT accepts the non-nominal value of the IDE and r0 bits in a valid standard frame.

One (1) value shall be tested as specified in Table 6.

Table 6 — Non-nominal IDE

IDE	r0
0	1

7.1.5.2 Test case organization

Test case organization shall be in accordance with Table 7.

Table 7 — Acceptance of non-nominal IDE, r0 combination in standard format — Test case organization

State	Description
Set-up	No action required, the IUT is left in the default state.
Test	A single test frame is used for the elementary test.
Verification	The IUT shall not generate any error flag in this test frame. The IUT shall acknowledge the test frame. The data received by the IUT during the test state shall match the data sent in the test frame.

7.1.6 Acceptance of non-nominal values of SRR, r1, r0 in extended format — Test case 1

7.1.6.1 Purpose and limits of test case

This test is applicable to CAN_VERSION $\in$ {B}.

Its purpose is to verify that the IUT accepts the non-nominal values of the SRR, r1 and r0 bits in a valid extended frame.

Seven (7) values shall be tested, in accordance with Table 8.

Table 8 — Non-nominal values of the SRR, r1 and r0 bits

SRR	r1	r0
1	1	1
1	1	0
1	0	1
0	1	1
0	1	0
0	0	1
0	0	0

7.1.6.2 Test case organization

Test case organization shall be in accordance with Table 9.

Table 9 — Acceptance of non-nominal values of SRR, r1, r0 in extended format — Test case 1 organization

State	Description
Set-up	No action required, the IUT is left in the default state.
Test	A single test frame is used for each of the seven elementary tests.
Verification	The IUT shall not generate any error flag during the test. The IUT shall acknowledge the test frame. The data received by the IUT during the test state shall match the data sent in the test frame.

7.1.7 Acceptance of non-nominal values of SRR, r1 and r0 in extended format — Test case 2

7.1.7.1 Purpose and limits of test case

This test is applicable to CAN_VERSION $\in$ {BP}.

Its purpose is to verify that the IUT accepts the non-nominal value of the SRR, r1 and r0 bits in a valid extended frame.

Seven (7) values to test shall be tested, in accordance with Table 10.

Table 10 — Non-nominal values of SRR, r1, r0 bits

SRR	r1	r0
1	1	1
1	1	0
1	0	1
0	1	1
0	1	0
0	0	1
0	0	0

7.1.7.2 Test case organization

Test case organization shall be in accordance with Table 11.

Table 11 — Acceptance of non-nominal values of SRR, r1 and r0 in extended format — Test case 2 organization

State	Description
Set-up	No action required, the IUT is left in the default state.
Test	A single test frame is used for each of the 7 elementary tests.
Verification	The IUT shall not generate any error flag during the test. The IUT shall acknowledge the test frame.

7.1.8 DLC greater than 8

7.1.8.1 Purpose and limits of test case

This test is applicable to CAN_VERSION $\in$ {A, B, BP}.

It is used to verify the behaviour of the IUT when receiving a correct frame with a DLC field greater than eight (8).

Seven elementary tests shall be performed, for which DLC ∈ [9, Fh].

7.1.8.2 Test case organization

Test case organization shall be in accordance with Table 12.

Table 12 — DLC greater than 8 — Test case organization

State	Description
Set-up	No action required, the IUT is left in the default state.
Test	A single test frame is used for each of the elementary tests.
Verification	The IUT shall not generate any error flag during the test. The IUT shall acknowledge the test frame. If 8 is an element of NDATA, the data received by the IUT during the test state shall match the data sent in the test frame.

7.1.9 Absent bus idle

7.1.9.1 Purpose and limits of test case

This test is applicable to CAN_VERSION ∈ {A, B, BP}.

It is used to verify the behaviour of the IUT when receiving two consecutive frames not separated by a bus idle state.

Two cases shall be tested:

— the second frame starts after the second intermission bit of the first frame;

— the second frame starts after the third intermission bit of the first frame.

7.1.9.2 Test case organization

Test case organization shall be in accordance with Table 13.

Table 13 — Absent bus idle — Test case organization

State	Description
Set-up	No action required, the IUT is left in the default state.
Test	Two test frames are used for each of the two elementary tests.
Verification	The IUT shall not generate any error flag during the test. The IUT shall acknowledge the test frames.

7.1.10 Stuff acceptance — Test case 1

7.1.10.1 Purpose and limits of test case

This test is applicable to CAN_VERSION ∈ {A, B, BP}.

It is used to verify the behaviour of the IUT when receiving a correct standard frame with particular data containing critical stuffing bit profiles in the different fields of the frame.

The fields of the tested frame shall be in accordance with Table 14.

Table 14 — Stuff acceptance — Test case 1 — Fields of the tested frame

Frame	IDEN	RTR	CTRL	Data
1	078h	0	8	01h, E1h, E1h, E1h, E1h, E1h, E1h, E1h
2	41Fh	0	1	00
3	707h	0	1Fh	0Fh, 0Fh, 0Fh, 0Fh, 0Fh, 0Fh, 0Fh, 0Fh
4	360h	0	10h	—
5	730h	0	10h	—
6	47Fh	0	01h	1Fh
7	758h	0	00h	—
8	777h	0	01h	1Fh
9	7Efh	1	02h	—
10	3Eah	1	1Fh	—

7.1.10.2 Test case organization

Test case organization shall be in accordance with Table 15.

Table 15 — Stuff acceptance — Test case 1 organization

State	Description
Set-up	No action required, the IUT is left in the default state.
Test	A single test frame is used for each elementary test.
Verification	The IUT shall not generate any error flag during the test. The IUT shall acknowledge the test frame. The data received by the IUT during the test state shall match the data sent in the test frame.

7.1.11 Stuff acceptance — Test case 2

7.1.11.1 Purpose and limits of this test case

This test is applicable to CAN_VERSION $\in$ {B, BP}.

It is used to verify the behaviour of the IUT when receiving a correct extended frame with particular data containing critical stuffing bit profiles in the different fields of the frame.

The fields of the tested frame shall be as given in the Table 16.

Table 16 — Stuff acceptance — Test case 2 — Fields of the tested frame

Frame	IDEN 1	SRR	IDEN 2	RTR	CTRL	Data
1	1F0h	1	30F0Fh	0	8	3Ch, 3Ch, 3Ch, 3Ch, 3Ch, 3Ch, 3Ch, 3Ch
2	1F0h	1	0F0F0h	0	1	00
3	078h	1	31717h	0	1Fh	0Fh, 0Fh, 0Fh, 0Fh, 0Fh, 0Fh, 0Fh, 0Fh
4	078h	1	00FF0h	0	3Ch	1Fh, 0Fh, E0h, F0h, 7Fh, E0h, FFh, 20h
5	7EEh	1	0	0	01h	A0h
6	02Fh	1	0540Fh	1	20h	—
7	557h	1	15557h	1	3Fh	—

7.1.11.2 Test case organization

Test case organization shall be in accordance with Table 17.

Table 17 — Stuff acceptance — Test case 2 organization

State	Description
Set-up	No action required, the IUT is left in the default state.
Test	A single test frame is used for each of the n elementary tests.
Verification	The IUT shall not generate any error flag during the test. The IUT shall acknowledge the test frame. The data received by the IUT during the test state shall match the data sent in the test frame.

7.1.12 Message validation

7.1.12.1 Purpose and limits of test case

This test is applicable to CAN_VERSION $\in$ {A, B, BP}.

Its purpose is to verify the point of time at which a message is taken to be valid by the IUT.

7.1.12.2 Test case organization

Test case organization shall be in accordance with Table 18.

Table 18 — Message validation — Test case organization

State	Description
Set-up	The IUT has to be initialized with data different from those used in the test frame.
Test	A single test frame is used for the elementary test. The last bit of the EOF is forced to dominant state.
Verification	The IUT shall not generate any error flag during the test. The IUT shall acknowledge the test frame. The IUT shall generate an overload frame. The data received by the IUT during the test state shall match the data sent in the test frame.

7.1.13 DLC not belonging to NDATA

7.1.13.1 Purpose and limits of test case

This test is applicable to CAN_VERSION $\in$ {A, B, BP}.

It is used to verify the behaviour of the IUT when receiving a correct frame with a DLC not belonging to NDATA and lower than a value of nine (9).

7.1.13.2 Test case organization

Test case organization shall be in accordance with Table 19.

Table 19 — DLC not belonging to NDATA — Test case organization

State	Description
Set-up	No action required, the IUT is left in the default state.
Test	A single test frame is used for each of the elementary tests.
Verification	The IUT shall not generate any error flag during the test. The IUT shall acknowledge the test frame.

7.2 Test class 2, error detection

7.2.1 Bit error in data frame

7.2.1.1 Purpose and limits of test case

This test is applicable to CAN_VERSION ∈ {A, B, BP}.

It is used to verify that the IUT detects a bit error when the dominant ACK slot is forced to recessive state by LT.

7.2.1.2 Test case organization

Test case organization shall be in accordance with Table 20.

Table 20 — Bit error in data frame — Test case organization

State	Description
Set-up	No action required, the IUT is left in the default state.
Test	A single test frame is used for the elementary test. The dominant acknowledgement bit sent by the IUT is forced to recessive state.
Verification	The IUT shall generate an active error frame starting at the bit position following the bit error.

7.2.2 Stuff error — Test case 1

7.2.2.1 Purpose and limits of test case

This test is applicable to CAN_VERSION ∈ {A, B, BP}.

It is used to verify that the IUT detects a stuff error whenever it receives six consecutive bits of the same value until the position of the CRC delimiter in a standard frame.

The frames used in this test shall be as given in Table 21.

Table 21 — Stuff error — Test case 1 — Frame specification

Frame	IDEN	RTR	CTRL	Data
1	078h	0	8	01h, E1h, E1h, E1h, E1h, E1h, E1h, E1h
2	41Fh	0	1	00
3	707h	0	1Fh	0Fh, 0Fh, 0Fh, 0Fh, 0Fh, 0Fh, 0Fh, 0Fh
4	360h	0	10h	—
5	730h	0	10h	—
6	47Fh	0	01h	1Fh
7	758h	0	00h	—
8	777h	0	01h	1Fh
9	7EFh	1	02h	—
10	3EAh	1	1Fh	—

7.2.2.2 Test case organization

Test case organization shall be in accordance with Table 22.

Table 22 — Stuff error — Test case 1 organization

State	Description
Set-up	No action required, the IUT is left in the default state.
Test	A single test frame is used for each elementary test. In each elementary test the LT forces one of the stuff bits to its complement.
Verification	The IUT shall generate an active error frame starting at the bit position following the stuff error.

7.2.3 Stuff error — Test case 2

7.2.3.1 Purpose and limits of test case

This test is applicable to CAN_VERSION $\in$ {B, BP}.

It is used to verify that the IUT detects a stuff error whenever it receives 6 consecutive bits of the same value until the position of the CRC delimiter in an extended frame.

The frames used in this test shall be as given in Table 23.

Table 23 — Stuff error — Test case 2 — Frame specification

Frame	IDEN 1	SRR	IDEN 2	RTR	CTRL	Data
1	1F0h	1	30F0Fh	0	8	3Ch, 3Ch, 3Ch, 3Ch, 3Ch, 3Ch, 3Ch, 3Ch
2	1F0h	1	0F0F0h	0	1	00
3	078h	1	31717h	0	1Fh	0Fh, 0Fh, 0Fh, 0Fh, 0Fh, 0Fh, 0Fh, 0Fh
4	078h	1	00FF0h	0	3Ch	1Fh, 0Fh, E0h, F0h, 7Fh, E0h, FFh, 20h
5	7EEh	1	0	0	01h	A0h
6	02Fh	1	0540Fh	1	20h	—
7	557h	1	15557h	1	3Fh	—

7.2.3.2 Test case organization

Test case organization shall be in accordance with Table 24.

Table 24 — Stuff error — Test case 2 organization

State	Description
Set-up	No action required, the IUT is left in the default state.
Test	A single test frame is used for each elementary test. In each elementary test the LT forces one of the stuff bits to its complement.
Verification	The IUT shall generate an active error frame starting at the bit position following the stuff error.

7.2.4 CRC error

7.2.4.1 Purpose and limits of this test case

This test is applicable to CAN_VERSION ∈ {A, B, BP}

Its purpose is to verify

— that the IUT uses the specific CRC mechanism as in the CAN specification, and

— that an IUT detecting a CRC error generates an error frame at the correct position.

Two elementary tests shall be performed:

a) a dominant bit in the CRC field is changed in a recessive one;

b) a recessive bit in the CRC field is changed in a dominant one.

7.2.4.2 Test case organization

Test case organization shall be in accordance with Table 25.

Table 25 — CRC error — Test case organization

State	Description
Set-up	No action required, the IUT is left in the default state.
Test	A single test frame is used for each elementary test. In each elementary test the LT modifies the frame according to 7.2.4.1.
Verification	The IUT shall not acknowledge the test frame The IUT shall generate an active error frame starting at the bit position following the ACK delimiter.

7.2.5 Combination of CRC error and form error

7.2.5.1 Purpose and limits of this test case

This test is applicable to CAN_VERSION ∈ {A, B, BP}.

Its purpose is to verify that an IUT detecting a CRC error and a form error on the CRC delimiter in the same frame generates only a single, 6-bit long error flag starting on the bit following the CRC delimiter.

One elementary test shall be performed.

7.2.5.2 Test case organization

Test case organization shall be in accordance with Table 26.

Table 26 — Combination of CRC error and form error — Test case organization

State	Description
Set-up	No action required, the IUT is left in the default state.
Test	A single test frame is used for the elementary test.
Verification	The IUT shall generate an active error frame.

7.2.6 Form error in data frame — Test case 1

7.2.6.1 Purpose and limits of test case

This test is applicable to CAN_VERSION ∈ {A, B, BP}

It is used to verify that the IUT detects a form error when the recessive bit of CRC delimiter is forced to dominant state by LT.

One elementary test shall be performed.

7.2.6.2 Test case organization

Test case organization shall be in accordance with Table 27.

Table 27 — Form error in data frame — Test case 1 organization

State	Description
Set-up	No action required, the IUT is left in the default state.
Test	A single test frame is used for the elementary test.
Verification	The IUT shall generate an active error frame at the bit position following the CRC delimiter.

7.2.7 Form error in data frame — Test case 2

7.2.7.1 Purpose and limits of test case

This test is applicable to CAN_VERSION ∈ {A, B, BP}.

It is used to verify that the IUT detects a form error when the recessive ACK delimiter is forced to dominant state by LT.

One elementary test shall be performed.

7.2.7.2 Test case organization

Test case organization shall be in accordance with Table 28.

Table 28 — Form error in data frame — Test case 2 organization

State	Description
Set-up	No action required, the IUT is left in the default state.
Test	A single test frame is used for the elementary test.
Verification	The IUT shall generate an active error frame at the bit position following the ACK delimiter.

7.2.8 Form error in data frame — Test case 3

7.2.8.1 Purpose and limits of test case

This test is applicable to CAN_VERSION ∈ {A, B, BP}.

It is used to verify that the IUT detects a form error when one of six first recessive bits of EOF is forced to dominant state by LT.

Three elementary tests shall be performed, corrupting the first, last and 3rd bit position.

7.2.8.2 Test case organization

Test case organization shall be in accordance with Table 29.

Table 29 — Form error in data frame — Test case 3 organization

State	Description
Set-up	No action required, the IUT is left in the default state.
Test	A single test frame is used for the elementary test.
Verification	The IUT shall generate an active error frame at the bit position following the corrupted bit.

7.2.9 Message non-validation

7.2.9.1 Purpose and limits of test case

This test is applicable to CAN_VERSION $\in$ {A, B, BP}.

Its purpose is to verify the point of time at which a message is still considered as non-valid by the IUT.

One elementary test shall be performed.

7.2.9.2 Test case organization

Test case organization shall be in accordance with Table 30.

Table 30 — Message non-validation — Test case organization

State	Description
Set-up	The IUT has to be initialized with data different from those used in the test frame.
Test	A single test frame is used for the elementary test. The sixth bit of the EOF is forced to dominant. A single test frame is used for the elementary test.
Verification	The IUT shall generate an active error frame. The data initialized during the set-up state shall remain unchanged. No frame reception is indicated to the upper layers of the IUT.

7.3 Test class 3, active error frame management

7.3.1 Error flag longer than 6 bits

7.3.1.1 Purpose and limits of test case

This test is applicable to CAN_VERSION $\in$ {A, B, BP}.

It is used to verify that the IUT tolerates up to seven consecutive dominant bits after sending an active error flag.

Three elementary tests shall be performed, lengthening the error flag by one, four and seven dominant bits.

7.3.1.2 Test case organization

Test case organization shall be in accordance with Table 31.

Table 31 — Error flag longer than 6 bits — Test case organization

State	Description
Set-up	No action required, the IUT is left in the default state.
Test	The LT causes the IUT to generate an error frame. The LT lengthens the error flag generated by the IUT.
Verification	After sending the active error flag, the IUT sends recessive bits.

7.3.2 Data frame starting on third bit of intermission field

7.3.2.1 Purpose and limits of test case

This test is applicable to CAN_VERSION $\in$ {A, B, BP}.

The purpose of this test is to verify that an IUT accepts a frame starting after the second bit of the intermission following the error frame it has transmitted.

One elementary test shall be performed.

7.3.2.2 Test case organization

Test case organization shall be in accordance with Table 32.

Table 32 — Data frame starting on third bit of intermission field — Test case organization

State	Description
Set-up	No action required, the IUT is left in the default state.
Test	The LT causes the IUT to generate an error frame. A single test frame is started two (2) bits after the end of the error delimiter.
Verification	The IUT shall acknowledge the test frame. The data received by the IUT during the test state shall match the data sent in the test frame.

7.3.3 Bit error in error flag

7.3.3.1 Purpose and limits of test case

This test is applicable to CAN_VERSION $\in$ {A, B, BP}.

It is used to verify that the IUT detects a bit error when one of the six dominant bits of the error flag it transmits is forced to recessive state by LT.

Three elementary tests shall be performed, corrupting the first, third and sixth bit of the error flag.

7.3.3.2 Test case organization

Test case organization shall be in accordance with Table 33.

Table 33 — Bit error in error flag — Test case organization

State	Description
Set-up	No action required, the IUT is left in the default state.
Test	The LT causes the IUT to generate an error frame. The LT forces one of the above mentioned bits of the error frame generated by the IUT to recessive state.
Verification	The IUT shall restart with an active error frame at the bit position following the corrupted bit.

7.3.4 Form error in error delimiter

7.3.4.1 Purpose and limits of test case

This test is applicable to CAN_VERSION ∈ {A, B, BP}.

It is used to verify that the IUT detects a form error when receiving an invalid error delimiter.

The LT replaces one of the eight recessive bits of the error delimiter by a dominant bit.

Three elementary tests shall be performed, corrupting the second, fourth and seventh bit of the error delimiter.

7.3.4.2 Test case organization

Test case organization shall be in accordance with Table 34.

Table 34 — Form error in error delimiter — Test case organization

State	Description
Set-up	No action required, the IUT is left in the default state.
Test	The LT causes the IUT to generate an error frame. The LT replaces one of the above mentioned recessive bits of the error delimiter with a dominant bit.
Verification	The IUT shall restart with an active error frame at the bit position following the replaced bit.

7.4 Test class 4, overload frame management

7.4.1 MAC overload generation during intermission field

7.4.1.1 Purpose and limits of test case

This test is applicable to CAN_VERSION ∈ {A, B, BP}.

It is used to verify that the IUT generates an overload frame when detecting a dominant bit on one of the two first recessive bits of the intermission field.

Two elementary tests shall be performed.

7.4.1.2 Test case organization

Test case organization shall be in accordance with Table 35.

Table 35 — MAC overload generation during intermission field — Test case organization

State	Description
Set-up	No action required, the IUT is left in the default state.
Test	One test frame is used for each of the two elementary tests. The LT forces one of the two first bits of the intermission field of the test frame to a dominant value.
Verification	The IUT generates an overload frame at the bit position following the dominant bit.

7.4.2 Last bit of EOF

7.4.2.1 Purpose and limits of test case

This test is applicable to CAN_VERSION ∈ {A, B, BP}.

It is used to verify that the IUT generates an overload frame when detecting a dominant state on the last bit of EOF.

One elementary test shall be performed.

7.4.2.2 Test case organization

Test case organization shall be in accordance with Table 36.

Table 36 — Last bit of EOF — Test case organization

State	Description
Set-up	No action required, the IUT is left in the default state.
Test	The LT forces the last bit of the EOF to a dominant state.
Verification	The IUT generates an overload frame at the bit position following the dominant bit.

7.4.3 Eighth bit of an error and overload delimiter

7.4.3.1 Purpose and limits of test case

This test is applicable to CAN_VERSION $\in$ {A, B, BP}.

It is used to verify that the IUT generates an overload frame when detecting a dominant bit on the eighth bit of an error and overload delimiter it is transmitting.

Two elementary tests shall be performed.

7.4.3.2 Test case organization

Test case organization shall be in accordance with Table 37.

Table 37 — Eighth bit of an error and overload delimiter — Test case organization

State	Description
Set-up	No action required, the IUT is left in the default state.
Test	The LT causes the IUT to generate an error frame or an overload frame. The LT forces to a dominant state the eighth bit of the delimiter.
Verification	The IUT generates an overload frame starting at the bit position following the dominant bit forced by the LT.

7.4.4 Bit error in overload flag

7.4.4.1 Purpose and limits of test case

This test is applicable to CAN_VERSION $\in$ {A, B, BP}

It is used to verify that the IUT detects a bit error when one of the six dominant bits of the overload flag it transmits is forced to recessive state by LT.

Three elementary tests shall be performed, corrupting the first, third and sixth bit of the overload flag.

7.4.4.2 Test case organization

Test case organization shall be in accordance with Table 38.

Table 38 — Bit error in overload flag — Test case organization

State	Description
Set-up	No action required, the IUT is left in the default state.
Test	The LT causes the IUT to generate an overload frame. The LT forces one of the above mentioned bits of the overload flag to the recessive state.
Verification	The IUT shall generate an error frame at the bit position following the corrupted bit.

7.4.5 Form error in overload delimiter

7.4.5.1 Purpose and limits of test case

This test is applicable to CAN_VERSION $\in$ {A, B, BP}.

It is used to verify that the IUT detects a form error when receiving an invalid overload delimiter.

The LT replaces one of the eight recessive bits of the overload delimiter by a dominant bit.

Three elementary tests shall be performed, corrupting the second, fourth and seventh bit of the overload delimiter.

7.4.5.2 Test case organization

Test case organization shall be in accordance with Table 39.

Table 39 — Form error in overload delimiter — Test case organization

State	Description
Set-up	No action required, the IUT is left in the default state.
Test	The LT causes the IUT to generate an overload frame. The LT replaces one of the above mentioned recessive bits of the overload delimiter with a dominant bit.
Verification	The IUT generates an error frame starting at the bit position following the replaced bit.

7.5 Test class 5, passive-error state and bus-off

7.5.1 Passive-error flag completion — Test case 1

7.5.1.1 Purpose and limits of test case

This test is applicable to CAN_VERSION $\in$ {A, B, BP}.

Its purpose is to verify that an error-passive IUT considers the passive-error flag as completed after the detection of six consecutive bits of the same value.

Three elementary tests shall be performed, superimposing the passive-error flag by an active-error flag starting at the first, the third and the sixth bits.

7.5.1.2 Test case organization

Test case organization shall be in accordance with Table 40.

Table 40 — Passive-error flag completion — Test case 1 organization

State	Description
Set-up	The IUT is set in passive state.
Test	The LT causes the IUT to generate a passive-error frame. During the passive-error flag sent by the IUT, the LT sends an active-error flag. At the end of the active-error flag the LT waits for (8 + 2) bit time before sending a valid test frame.
Verification	The IUT shall acknowledge the test frame.

7.5.2 Data frame acceptance after passive-error frame transmission

7.5.2.1 Purpose and limits of test case

This test is applicable to CAN_VERSION $\in$ {A, B, BP}.

Its purpose is to verify that an error-passive IUT accepts a frame starting after the second bit of the intermission following the error frame it has transmitted.

One elementary test shall be performed.

7.5.2.2 Test case organization

Test case organization shall be in accordance with Table 41.

Table 41 — Data frame acceptance after passive-error frame transmission — Test case organization

State	Description
Set-up	The IUT is set in passive state.
Test	The LT causes the IUT to generate a passive-error frame. At the end of the passive-error flag the LT waits for (8 + 2) bit time before sending a valid test frame.
Verification	The IUT shall acknowledge the test frame.

7.5.3 Acceptance of 7 consecutive dominant bits after passive-error flag

7.5.3.1 Purpose and limits of this test case

This test is applicable to CAN_VERSION $\in$ {A, B, BP}.

Its purpose is to verify that an error-passive IUT does not detect any error when detecting up to 7 consecutive dominant bits starting at the bit position following the last bit of the passive-error flag.

Three elementary tests shall be performed, transmitting one, four or seven consecutive dominant bits.

7.5.3.2 Test case organization

Test case organization shall be in accordance with Table 42.

Table 42 — Acceptance of 7 consecutive dominant bits after passive-error flag — Test case organization

State	Description
Set-up	The IUT is set in passive state.
Test	The LT causes the IUT to generate a passive-error frame. After the passive-error flag, the LT starts transmitting dominant bits according to 7.5.3.1. After the dominant bit sequence the LT waits for (8 + 2) bit time before sending a valid test frame.
Verification	The IUT shall acknowledge the test frame.

7.5.4 Passive state unchanged on further errors

7.5.4.1 Purpose and limits of this test case

This test is applicable to CAN_VERSION ∈ {A, B, BP}.

It is used to verify that an error-passive IUT does not become error active on any error detection.

One elementary test shall be performed.

7.5.4.2 Test case organization

Test case organization shall be in accordance with Table 43.

Table 43 — Passive state unchanged on further errors — Test case organization

State	Description
Set-up	The IUT is set to passive state.
Test	The LT sends at least nine invalid test frames.
Verification	The IUT shall not generate any active error frame.

7.5.5 Passive-error flag completion — Test case 2

7.5.5.1 Purpose and limits of test case

This test is applicable to CAN_VERSION ∈ {A, B, BP}.

Its purpose is to verify that an error-passive IUT restarts the passive-error flag when detecting up to five consecutive dominant bits during its own passive-error flag.

Three elementary tests shall be performed, superimposing the passive-error flag by the sequence of five dominant bits starting at the first, third and sixth bit of the passive-error flag.

7.5.5.2 Test case organization

Test case organization shall be in accordance with Table 44.

Table 44 — Passive-error flag completion — Test case 2 organization

State	Description
Set-up	The IUT is set in passive state.
Test	The LT causes the IUT to generate a passive-error frame. During the passive-error flag sent by the IUT the LT sends a sequence of five dominant bits according to 7.5.5.1. After this sequence, the LT waits for (6 + 7) bit time before sending a dominant bit, corrupting the last bit of the error delimiter.
Verification	Following the dominant bit sent by the LT the IUT shall generate an overload frame.

7.5.6 Form error in passive-error delimiter

7.5.6.1 Purpose and limits of test case

This test is applicable to CAN_VERSION ∈ {A, B, BP}.

Its purpose is to verify that an error-passive IUT detects a form error when receiving an invalid error delimiter.

The LT replaces one of the eight recessive bits of the error delimiter by a dominant bit.

Three elementary tests shall be performed, corrupting the second, fourth and seventh bit of the error delimiter.

7.5.6.2 Test case organization

Test case organization shall be in accordance with Table 45.

Table 45 — Form error in passive-error delimiter — Test case organization

State	Description
Set-up	The IUT is set in passive state.
Test	The LT causes the IUT to generate a passive-error frame. During reception of the error delimiter, the LT creates a form error according to 7.5.6.1. After creating the form error, the LT waits for (6 + 7) bit time before sending a dominant bit, corrupting the last bit of the error delimiter.
Verification	The IUT shall generate an overload frame starting at the bit position following the last dominant bit sent by the LT.

7.6 Test class 6, error counter management

7.6.1 REC increment on bit error in active-error flag

7.6.1.1 Purpose and limits of test case

This test is applicable to CAN_VERSION ∈ {A, B, BP}.

It is used to verify that the IUT increases its REC by 8, when detecting a bit error during the transmission of an active-error flag.

Three elementary tests shall be performed, corrupting the first, third and sixth bit of the active-error flag.

7.6.1.2 Test case organization

Test case organization shall be in accordance with Table 46.

Table 46 — REC increment on bit error in active-error flag — Test case organization

State	Description
Set-up	No action required, the IUT is left in the default state.
Test	The LT causes the IUT to generate an active error frame. The LT corrupts one of the above mentioned dominant bits of the error flag.
Verification	The REC is increased by 8 on the corrupted bit.

7.6.2 REC increment on bit error in overload flag

7.6.2.1 Purpose and limits of test case

This test is applicable to CAN_VERSION $\in$ {A, B, BP}.

It is used to verify that the IUT increases its REC by 8, when detecting a bit error during the transmission of an overload flag.

Three elementary tests shall be performed, corrupting the first, fourth and sixth bit of the overload flag.

7.6.2.2 Test case organization

Test case organization shall be in accordance with Table 47.

Table 47 — REC increment on bit error in overload flag — Test case organization

State	Description
Set-up	No action required, the IUT is left in the default state.
Test	The LT causes the IUT to generate an overload frame. The LT corrupts one of the above-mentioned dominant bits of the overload flag.
Verification	The REC is increased by 8 on the corrupted bit.

7.6.3 REC increment when active-error flag is longer than 13 bits

7.6.3.1 Purpose and limits of test case

This test is applicable to CAN_VERSION $\in$ {A, B, BP}.

It is used to verify that the IUT increases its REC by 8, when detecting the eighth consecutive dominant bit following the transmission of its active-error flag and after each sequence of additional eight consecutive dominant bits.

One elementary test shall be performed.

7.6.3.2 Test case organization

Test case organization shall be in accordance with Table 48.

Table 48 — REC increment when active-error flag longer than 13 bits — Test case organization

State	Description
Set-up	No action required, the IUT is left in the default state.
Test	The LT causes the IUT to generate an active error frame. After the error flag sent by the IUT, the LT sends a sequence of 16 dominant bits.
Verification	The REC is increased by 8 on each eighth dominant bit after the error flag.

7.6.4 REC increment when overload flag is longer than 13 bits

7.6.4.1 Purpose and limits of test case

This test is applicable to CAN_VERSION $\in$ {A, B, BP}.

It is used to verify that the IUT increases its REC by 8, when detecting the eighth consecutive dominant bit following the transmission of its overload flag and after each sequence of additional eight consecutive dominant bits.

One elementary test shall be performed.

7.6.4.2 Test case organization

Test case organization shall be in accordance with Table 49.

Table 49 — REC increment when overload flag longer than 13 bits — Test case organization

State	Description
Set-up	No action required, the IUT is left in the default state.
Test	The LT causes the IUT to generate an overload frame. After the overload flag sent by the IUT, the LT sends a sequence of 16 dominant bits.
Verification	The REC is increased by 8 on each eighth dominant bit after the overload flag.

7.6.5 REC increment on bit error in the ACK field

7.6.5.1 Purpose and limits of test case

This test is applicable to CAN_VERSION $\in$ {A, B, BP}.

It is used to verify that the IUT increases its REC by 1, when detecting a bit error on the ACK slot it transmits.

One elementary test shall be performed.

7.6.5.2 Test case organization

Test case organization shall be in accordance with Table 50.

Table 50 — REC increment on bit error in the ACK field — Test case organization

State	Description
Set-up	No action required, the IUT is left in the default state.
Test	The LT causes the IUT to send a dominant acknowledgement, the ACK slot is corrupted by LT.
Verification	The REC is increased by 1 on the corrupted bit.

7.6.6 REC increment on form error at CRC delimiter

7.6.6.1 Purpose and limits of test case

This test is applicable to CAN_VERSION ∈ {A, B, BP}.

It is used to verify that the IUT increases its REC by 1, when detecting a form error at CRC delimiter.

One elementary test shall be performed.

7.6.6.2 Test case organization

Test case organization shall be in accordance with Table 51.

Table 51 — REC increment on form error at CRC delimiter — Test case organization

State	Description
Set-up	No action required, the IUT is left in the default state.
Test	The LT sends a frame with the CRC delimiter changed to a dominant value.
Verification	The REC is increased by 1 on the dominant CRC delimiter.

7.6.7 REC increment on form error at ACK delimiter

7.6.7.1 Purpose and limits of test case

This test is applicable to CAN_VERSION ∈ {A, B, BP}.

It is used to verify that the IUT increases its REC by 1, when detecting a form error on ACK delimiter.

One elementary test shall be performed.

7.6.7.2 Test case organization

Test case organization shall be in accordance with Table 52.

Table 52 — REC increment on form error at ACK delimiter — Test case organization

State	Description
Set-up	No action required, the IUT is left in the default state.
Test	The LT sends a frame with the ACK delimiter changed to a dominant value.
Verification	The REC is increased by 1 on the dominant ACK delimiter.

7.6.8 REC increment on form error in EOF field

7.6.8.1 Purpose and limits of test case

This test is applicable to CAN_VERSION ∈ {A, B, BP}.

It is used to verify that the IUT increases its REC by 1, when detecting a form error on the EOF field during reception of a data frame.

Three elementary tests shall be performed, corrupting the second, third and fifth bit of the EOF.

7.6.8.2 Test case organization

Test case organization shall be in accordance with Table 53.

Table 53 — REC increment on form error in EOF field — Test case organization

State	Description
Set-up	No action required, the IUT is left in the default state.
Test	The LT sends a frame with the EOF modified according to 7.6.8.1.
Verification	The REC is increased by 1 on the replaced bit of the EOF.

7.6.9 REC increment on stuff error

7.6.9.1 Purpose and limits of test case

This test is applicable to CAN_VERSION ∈ {A, B, BP}.

It is used to verify that the IUT increases its REC by 1, when detecting a stuff error.

Eight elementary tests shall be performed, testing both errors on recessive and dominant stuff bits in the arbitration, control, data and CRC fields.

7.6.9.2 Test case organization

Test case organization shall be in accordance with Table 54.

Table 54 — REC increment on stuff error — Test case organization

State	Description
Set-up	No action required, the IUT is left in the default state.
Test	The LT sends a sequence of six consecutive bits according to 7.6.9.1.
Verification	The REC is increased by 1 on the sixth consecutive bit.

7.6.10 REC increment on CRC error

7.6.10.1 Purpose and limits of test case

This test is applicable to CAN_VERSION ∈ {A, B, BP}.

It is used to verify that the IUT increases its REC by 1, when detecting a CRC error during reception of a frame.

One elementary test shall be performed.

7.6.10.2 Test case organization

Test case organization shall be in accordance with Table 55.

Table 55 — REC increment on CRC error — Test case organization

State	Description
Set-up	No action required, the IUT is left in the default state.
Test	The LT sends a frame containing a CRC error.
Verification	The IUT sends a recessive acknowledge. The IUT starts the transmission of an active error frame after the ACK delimiter. The REC is increased by 1 after the ACK delimiter.

7.6.11 REC increment on dominant bit after end of error flag

7.6.11.1 Purpose and limits of test case

This test is applicable to CAN_VERSION ∈ {A, B, BP}.

It is used to verify that an error active IUT increases its REC by 8, when detecting a dominant bit as the first bit after sending an error flag.

One elementary test shall be performed.

7.6.11.2 Test case organization

Test case organization shall be in accordance with Table 56.

Table 56 — REC increment on dominant bit after end of error flag — Test case organization

State	Description
Set-up	No action required, the IUT is left in the default state.
Test	The LT causes the IUT to generate an active-error flag. The LT sends a dominant bit at the bit position following the end of the error flag sent by the IUT.
Verification	The REC is increased by 8 after reception of the dominant bit sent by the LT.

7.6.12 REC increment on form error in error delimiter

7.6.12.1 Purpose and limits of test case

This test is applicable to CAN_VERSION ∈ {A, B, BP}.

It is used to verify that a receiver increases its REC by 1, when detecting a form error on a bit of the error delimiter it is transmitting.

Two elementary tests shall be performed, the second and the last bit of the error delimiter.

7.6.12.2 Test case organization

Test case organization shall be in accordance with Table 57.

Table 57 — REC increment on form error in error delimiter — Test case organization

State	Description
Set-up	No action required, the IUT is left in the default state.
Test	The LT causes the IUT to generate an active error frame. The LT corrupts 1 bit of the error delimiter according to 7.6.12.1.
Verification	The REC is increased by 1 after reception of the dominant bit sent by the LT.

7.6.13 REC increment on form error in overload delimiter

7.6.13.1 Purpose and limits of test case

This test is applicable to CAN_VERSION ∈ {A, B, BP}.

It is used to verify that a receiver increases its REC by 1, when detecting a form error on a bit of the overload delimiter it is transmitting.

Two elementary tests shall be performed, corrupting the second, and last bit of the overload delimiter.

7.6.13.2 Test case organization

Test case organization shall be in accordance with Table 58.

Table 58 — REC increment on form error in overload delimiter — Test case organization

State	Description
Set-up	No action required, the IUT is left in the default state.
Test	The LT causes the IUT to generate an overload frame. The LT corrupts 1 bit of the overload delimiter according to 7.6.13.1.
Verification	The REC is increased by 1 after reception of the dominant bit sent by the LT.

7.6.14 REC decrement on valid frame reception

7.6.14.1 Purpose and limits of test case

This test is applicable to CAN_VERSION $\in$ {A, B, BP}.

It is used to verify that the IUT decreases its REC by 1, when receiving a valid frame.

One elementary test shall be performed.

7.6.14.2 Test case organization

Test case organization shall be in accordance with Table 59.

Table 59 — REC decrement on valid frame reception — Test case organization

State	Description
Set-up	The LT forces the IUT to increase its REC.
Test	The LT sends one valid test frame.
Verification	The REC is decreased by 1 after the successful transmission of the ACK slot.

7.6.15 REC decrement on valid frame reception during passive state

7.6.15.1 Purpose and limits of test case

This test is applicable to CAN_VERSION $\in$ {A, B, BP}.

It is used to verify that the IUT set its REC to a value between 119 and 127, when receiving a valid frame while being error-passive.

One elementary test shall be performed.

7.6.15.2 Test case organization

Test case organization shall be in accordance with Table 60.

Table 60 — REC decrement on valid frame reception during passive state — Test case organization

State	Description
Set-up	The LT causes the IUT's REC value to be at error-passive level.
Test	The LT sends one valid test frame.
Verification	The REC shall be decremented to a value between 119 and 127 after the successful transmission of the ACK slot.

7.6.16 REC non-increment on last bit of EOF field

7.6.16.1 Purpose and limits of test case

This test is applicable to CAN_VERSION ∈ {A, B, BP}.

It is used to verify that the IUT does not change the value of its REC, when detecting a dominant bit at the last bit of the EOF it is receiving, and also that the REC is not decremented below zero.

One elementary test shall be performed.

7.6.16.2 Test case organization

Test case organization shall be in accordance with Table 61.

Table 61 — REC non-increment on last bit of EOF field — Test case organization

State	Description
Set-up	No action required, the IUT is left in the default state.
Test	The LT sends one valid test frame with a dominant bit at the last bit of EOF.
Verification	The REC value is zero.

7.6.17 REC non-increment on 13-bit length overload flag

7.6.17.1 Purpose and limits of test case

This test is applicable to CAN_VERSION ∈ {A, B, BP}.

It is used to verify that the IUT does not change the value of its REC when receiving a 13-bit length overload flag.

One elementary test shall be performed.

7.6.17.2 Test case organization

Test case organization shall be in accordance with Table 62.

Table 62 — REC non-increment on 13-bit length overload flag — Test case organization

State	Description
Set-up	No action required, the IUT is left in the default state.
Test	The LT causes the IUT to generate an overload frame. After the overload flag sent by the IUT, the LT sends a sequence of seven (7) dominant bits.
Verification	The REC value is zero.

7.6.18 REC non-increment on 13-bit length error flag

7.6.18.1 Purpose and limits of test case

This test is applicable to CAN_VERSION ∈ {A, B, BP}.

It is used to verify that the IUT does not increase its REC after the seventh bit of the received error flag.

One elementary test shall be performed.

7.6.18.2 Test case organization

Test case organization shall be in accordance with Table 63.

Table 63 — REC non-increment on 13-bit length error flag — Test case organization

State	Description
Set-up	No action required, the IUT is left in the default state.
Test	The LT causes the IUT to generate an active error frame. After the error flag sent by the IUT, the LT sends additional seven (7) consecutive dominant bits.
Verification	The REC is not further incremented after the increment, owing to the dominant bit which followed the error flag sent by the IUT.

7.6.19 REC non-increment on last bit of error and overload delimiter

7.6.19.1 Purpose and limits of test case

This test is applicable to CAN_VERSION ∈ {A, B, BP}.

It is used to verify that the IUT does not change the value of its REC when detecting a dominant bit at the last bit of an error and overload delimiter it is transmitting.

Two elementary tests shall be performed.

7.6.19.2 Test case organization

Test case organization shall be in accordance with Table 64.

Table 64 — REC non-increment on last bit of error and overload delimiter — Test case organization

State	Description
Set-up	No action required, the IUT is left in the default state.
Test	The LT causes the IUT to generate an error or an overload frame. The LT corrupts the last bit of the error or the overload delimiter.
Verification	The REC value is zero.

7.7 Test class 7, bit timing class

7.7.1 Sample point

7.7.1.1 Purpose and limits of test case

This test is applicable to CAN_VERSION ∈ {A, B, BP}.

Its purpose is to verify the position of the sample point of an IUT.

One elementary test shall be performed.

7.7.1.2 Test case organization

Test case organization shall be in accordance with Table 65.

Table 65 — Sample point — Test case organization

State	Description
Set-up	No action required, the IUT is left in the default state.
Test	The LT shortens a dominant stuff bit by an amount of Phase_Seg2 and then later shortens another dominant stuff bit by an amount of (Phase_Seg2 + 1TQ).
Verification	The IUT shall generate an error frame on the bit position following the second shortened stuff bit.

7.7.2 Hard synchronization on SOF reception

7.7.2.1 Purpose and limits of test case

This test is applicable to CAN_VERSION $\in$ {A, B, BP}.

Its purpose is to verify that the IUT makes a hard synchronization when receiving a SOF delayed by e, $e \in$ [1, NTQ].

NTQ elementary tests shall be performed.

7.7.2.2 Test case organization

Test case organization shall be in accordance with Table 66.

Table 66 — Hard synchronization on SOF reception — Test case organization

State	Description
Set-up	No action required, the IUT is left in the default state.
Test	The LT sends a first test frame and after the second bit of the intermission field it sends a SOF delayed by e TQ depending on the elementary test. The SOF is followed by a sequence of 5 dominant bits.
Verification	The IUT shall respond with an error frame exactly 6 bit times after the recessive to dominant edge at the beginning of the SOF.

7.7.3 Synchronization when $e > 0$ and $e \leqslant$ SJW

7.7.3.1 Purpose and limits of test case

This test is applicable to CAN_VERSION $\in$ {A, B, BP}.

Its purpose is to verify the behaviour of an IUT detecting a positive phase error, e, on a recessive to dominant edge with $e \leqslant$ SJW.

The values tested for e are in TQ with $e \in$ [1, SJW].

There is one elementary test shall be performed for each possible value of e.

7.7.3.2 Test case organization

Test case organization shall be in accordance with Table 67.

Table 67 — Synchronization when e > 0 and $e \leqslant$ SJW — Test case organization

State	Description
Set-up	No action required, the IUT is left in the default state.
Test	The LT delays a dominant stuff bit by an amount of e TQ and shortens the same bit by an amount of (Phase_Seg2 + 1TQ − e).
Verification	The IUT shall generate an error frame 1 bit time after the recessive to dominant edge of the delayed stuff bit.

7.7.4 Synchronization when e > 0 and e > SJW

7.7.4.1 Purpose and limits of test case

This test is applicable to CAN_VERSION ∈ {A, B, BP}.

Its purpose is to verify the behaviour of an IUT detecting a positive phase error, e, on a recessive to dominant edge with $e >$ SJW.

The values tested for e are in TQ with $e \in$ [(SJW + 1), (NTQ − (Phase_Seg2 + 1))].

One elementary test shall be performed for each possible value of e.

7.7.4.2 Test case organization

Test case organization shall be in accordance with Table 68.

Table 68 — Synchronization when e > 0 and e > SJW — Test case organization

State	Description
Set-up	No action required, the IUT is left in the default state.
Test	The LT delays a dominant stuff bit by an amount of e TQ and shortens the same bit by an amount of Phase_Seg2 + 1TQ − e).
Verification	The IUT shall generate an error frame 1 bit time + (e − SJW) time quanta after the recessive to dominant edge of the delayed stuff bit.

7.7.5 Synchronization when e < 0 and |e| ⩽ SJW

7.7.5.1 Purpose and limits of test case

This test is applicable to CAN_VERSION ∈ {A, B, BP}.

Its purpose is to verify the behaviour of an IUT detecting a negative phase error, e, on a recessive to dominant edge with |e| ⩽ SJW.

The values tested for e are in time quantum with |e| ∈ [1, SJW].

One elementary test shall be performed for each possible value of e.

7.7.5.2 Test case organization

Test case organization shall be in accordance with Table 69.

Table 69 — Synchronization when $e < 0$ and $|e| \leqslant$ SJW — Test case organization

State	Description
Set-up	No action required, the IUT is left in the default state.
Test	The LT shortens the last recessive bit before an expected dominant stuff bit by an amount of $\|e\|$ TQ and then sends a dominant value for 1 TQ followed by a recessive state.
Verification	The IUT shall generate an error frame 1 bit time after the last recessive to dominant edge.

7.7.6 Synchronization when $e < 0$ and $|e| >$ SJW

7.7.6.1 Purpose and limits of test case

This test is applicable to CAN_VERSION $\in$ {A, B, BP}.

Its purpose is to verify the behaviour of an IUT detecting a negative phase error, e, on a recessive to dominant edge with $|e| >$ SJW.

The values tested for e are in time quantum with $|e| \in$ [(SJW + 1), Phase_Seg2].

One elementary test shall be performed for each possible value of e.

7.7.6.2 Test case organization

Test case organization shall be in accordance with Table 70.

Table 70 — Synchronization when $e < 0$ and $|e| >$ SJW — Test case organization

State	Description
Set-up	No action required, the IUT is left in the default state.
Test	The LT shortens the last recessive bit before an expected dominant stuff bit by an amount of $\|e\|$ TQ and then sends a dominant value for 1 TQ followed by a recessive state.
Verification	The IUT shall generate an error frame 1 bit time $-$ ($\|e\| -$ SJW) TQ after the last recessive to dominant edge.

7.7.7 Glitch filtering test on positive phase error

7.7.7.1 Purpose and limits of test case

This test is applicable to CAN_VERSION $\in$ {A, B, BP}.

Its purpose is to verify that there is only one synchronization within one bit time if there are two dominant-to-recessive edges between synchronization segment and sample point. The test is also used to verify that an IUT is able to synchronize on a minimum duration pulse obeying the synchronization rules.

One elementary test shall be performed.

7.7.7.2 Test case organization

Test case organization shall be in accordance with Table 71.

Table 71 — Glitch filtering test on positive phase error — Test case organization

State	Description
Set-up	No action required, the IUT is left in the default state.
Test	The LT sends a frame containing a dominant stuff bit. After the first 2 TQ of dominant value, it changes 1 TQ to recessive value. This dominant stuff bit is followed by 6 recessive bits.
Verification	The IUT shall respond with an error frame exactly 7 bit times after the first recessive to dominant edge of the stuff bit.

7.7.8 Glitch filtering test on negative phase error

7.7.8.1 Purpose and limits of test case

This test is applicable to CAN_VERSION $\in$ {A, B, BP}.

Its purpose is to verify that there is only one synchronization within one bit time if there are two dominant to recessive edges between two sample points, where the first edge comes before the synchronization segment. It is also used to verify that an IUT is able to synchronize on a minimum duration pulse obeying to the synchronization rules.

One elementary test shall be performed.

7.7.8.2 Test case organization

Test case organization shall be in accordance with Table 72.

Table 72 — Glitch filtering test on negative phase error — Test case organization

State	Description
Set-up	No action required, the IUT is left in the default state.
Test	The LT sends a frame containing a dominant stuff bit. The recessive bit before the stuff bit is shortened by 1 TQ. After the first 2 TQ of dominant value it changes 1 TQ to recessive value. This dominant stuff bit is followed by 6 recessive bits.
Verification	The IUT shall respond with an error frame exactly 7 bit times after the first recessive to dominant edge of the stuff bit.

7.7.9 Glitch filtering in idle state

7.7.9.1 Purpose and limits of test case

This test is applicable to CAN_VERSION $\in$ {A, B, BP}.

Its purpose is to verify that an IUT will not detect a SOF when detecting a dominant level shorter than (propagation segment + phase buffer segment 1 – 1 T_Q).

One elementary test shall be performed.

7.7.9.2 Test case organization

Test case organization shall be in accordance with Table 73.

Table 73 — Glitch filtering in idle state — Test case organization

State	Description
Set-up	No action required, the IUT is left in the default state.
Test	The LT sends a dominant glitch according to 7.7.8.1. Then the LT waits for 8 bit times.
Verification	The IUT shall remain in the idle state.

7.7.10 Non-resynchronization after dominant sampled bit

7.7.10.1 Purpose and limits of test case

This test is applicable to CAN_VERSION $\in$ {A, B, BP}.

Its purpose is to verify that no edge will be used for synchronization if the value detected at the previous sample point is the same as the bus value immediately after the edge.

One elementary test shall be performed.

7.7.10.2 Test case organization

Test case organization shall be in accordance with Table 74.

Table 74 — Non-resynchronization after dominant sampled bit — Test case organization

State	Description
Set-up	No action required, the IUT is left in the default state.
Test	The LT sends a frame containing a dominant stuff bit. At the position (NTQ − Phase_Seg2 + 1) TQ after the edge at the beginning of the stuff bit, the LT changes the value to recessive for 1 TQ. The stuff bit is followed by 5 additional dominant bits.
Verification	The IUT shall respond with an error frame exactly 6 bit times after the recessive to dominant edge at the beginning of the stuff bit.

8 Test type 2, transmitted frame

8.1 Test class 1, valid frame format

8.1.1 Identifier and number of data bytes in standard format

8.1.1.1 Purpose and limits of test case

This test is applicable to CAN_VERSION $\in$ {A, B, BP}.

It is used to verify the capacity of the IUT to transmit a frame with different identifiers and different numbers of data in a standard format frame.

Tested identifiers: IDEN $\in$ [000h, 7EFh] $\cup$ [7F0h, 7FFh],

Tested number of data bytes: $\in$ [0, 8].

8.1.1.2 Test case organization

Test case organization shall be in accordance with Table 75.

Table 75 — Identifier and number of data bytes in standard format — Test case organization

State	Description
Set-up	No action required. The IUT is left in the default state.
Test	A single test frame is used for each elementary test. The LT causes the IUT to transmit a data frame with the parameters listed in 8.1.1.1.
Verification	The IUT shall not generate any error flag during the test. The content of the frame shall match the LT request. The number of data bytes shall match the DLC if the DLC is less than 9. Not more than 8 data bytes may be transmitted.

8.1.2 Identifier and number of data bytes in extended format

8.1.2.1 Purpose and limits of test case

This test is applicable to CAN_VERSION $\in$ {B}.

It is used to verify the capacity of the IUT to transmit a data frame with different identifiers and different numbers of data in an extended format frame.

Tested identifiers: IDEN $\in$ [0, 1FFFFFFFh]

Tested number of data bytes: $\in$ [0, 8]

8.1.2.2 Test case organization

Test case organization shall be in accordance with Table 76.

Table 76 — Identifier and number of data bytes in extended format — Test case organization

State	Description
Set-up	No action required. The IUT is left in the default state.
Test	A single test frame is used for each elementary test. The LT causes the IUT to transmit a data frame with the parameters listed in 8.1.2.1.
Verification	The IUT shall not generate any error flag during the test. The content of the frame shall match the LT request. The number of data bytes shall match the DLC if the DLC is less than 9. Not more than 8 data bytes may be transmitted.

8.1.3 Arbitration in standard format frame

8.1.3.1 Purpose and limits of test case

This test is applicable to CAN_VERSION $\in$ {A, B, BP}.

It is used to verify the capability of the IUT to manage the arbitration mechanism on every bit position in a standard format frame it is transmitting.

For an open device, the cases given in Table 77 shall be tested.

Table 77 — Arbitration in standard format frame — Cases to be tested for an open device

Transmitted frame			Description of the concerned arbitration bit(s)	Number of elementary test
IDEN	RTR	Data field		
7EFh	0	No data	Collision on all bits equal to 1	10
010h	1	No data	Collision on all bits equal to 1	2

For a specific device, all possible cases of transmitting a recessive arbitration bit shall be considered.

At most, 12 elementary tests shall be performed.

8.1.3.2 Test case organization

Test case organization shall be in accordance with Table 78.

Table 78 — Arbitration in standard format frame — Test case organization

State	Description
Set-up	No action required. The IUT is left in the default state.
Test	The LT causes the IUT to transmit a frame. Then the LT forces a recessive bit in the arbitration field to the dominant state according to Table 77 and continues to send a valid frame.
Verification	The IUT shall become receiver when sampling the dominant bit sent by the LT. The data received by the IUT shall match the data sent by the LT. As soon as the bus is idle the IUT shall restart the transmission of the frame. The IUT shall not generate any error flag during the test. The content of the frame shall match the LT request.

8.1.4 Arbitration in extended format frame

8.1.4.1 Purpose and limits of test case

This test is applicable to CAN_VERSION $\in$ {B}.

It is used to verify the capacity of the IUT to manage the arbitration mechanism on every bit position in an extended format frame it is transmitting.

For an open device, the cases given in Table 79 shall be tested.

Table 79 — Arbitration in extended format frame — Cases to be tested for an open device

Transmitted frame			Description of the concerned bit(s)	Number of elementary test
IDEN	RTR	Data field		
1FBFFFFFh	0	No data	Collision on all bits equal to 1	28
00400000h	1	No data	Collision on all bits equal to 1	2
00400000h	0	No data	Collision on SRR and IDE bit	2

For a specific device, all possible cases of transmitting a recessive arbitration bit shall be considered.

At most, 32 elementary tests shall be performed.

8.1.4.2 Test case organization

Test case organization shall be in accordance with Table 80.

Table 80 — Arbitration in extended format frame — Test case organization

State	Description
Set-up	No action required. The IUT is left in the default state.
Test	The LT causes the IUT to transmit a frame. The LT forces a recessive bit in the arbitration field to the dominant state according to Table 79 and continues to send a valid frame.
Verification	The IUT shall become receiver when sampling the dominant bit sent by the LT. The data received by the IUT shall match the data sent by the LT. As soon as the bus is idle the IUT shall restart the transmission of the frame. The IUT shall not generate any error flag during the test. The content of the frame shall match the LT request.

8.1.5 Message validation

8.1.5.1 Purpose and limits of test case

This test is applicable to CAN_VERSION $\in$ {A, B, BP}.

Its purpose is to verify the point of time at which a message transmitted by the IUT is taken to be valid.

One elementary test shall be performed.

8.1.5.2 Test case organization

Test case organization shall be in accordance with Table 81.

Table 81 — Message validation — Test case organization

State	Description
Set-up	No action required. The IUT is left in the default state.
Test	The LT causes the IUT to transmit a data frame. On the first bit of the intermission field of the frame sent by the IUT, the LT starts a SOF making the IUT generating an overload frame.
Verification	The IUT shall not generate any error flag during the test. The IUT shall not restart any frame after the overload frame.

8.1.6 Stuff bit generation capability in standard frame

8.1.6.1 Purpose and limits of test case

This test is applicable to CAN_VERSION $\in$ {A, B, BP}.

Its purpose is to verify that an IUT correctly generates the stuff bits in a standard frame.

For an open device, the cases given in Table 82 shall be tested.

For a specific device, all possible dominant and recessive stuff bits inside and at the end of each stuffed field shall be considered.

Six elementary tests shall be performed.

Table 82 — Stuff bit generation capability in standard frame — Cases to be tested for an open device

Frame	IDEN	RTR	CTRL	Data
1	078h	0	8	01h, E1h, E1h, E1h, E1h, E1h, E1h, E1h
2	41Fh	0	1	00
3	47Fh	0	01h	1Fh
4	758h	0	00h	—
5	777h	0	01h	1Fh
6	7EFh	1	02h	—

8.1.6.2 Test case organization

Test case organization shall be in accordance with Table 83.

Table 83 — Stuff bit generation capability in standard frame — Test case organization

State	Description
Set-up	No action required. The IUT is left in the default state.
Test	The LT causes the IUT to transmit a frame according to Table 82.
Verification	The IUT shall not generate any error flag during the test. The IUT shall correctly generate all stuff bits.

8.1.7 Stuff bit generation capability in extended frame

8.1.7.1 Purpose and limits of test case

This test is applicable to CAN_VERSION $\in$ {B}.

Its purpose is to verify that an IUT correctly generates the stuff bits in an extended frame.

For an open device, the cases given in Table 84 shall be tested.

Table 84 — Stuff bit generation capability in extended frame — Cases to be tested for an open device

Frame	IDEN 1	IDEN 2	RTR	CTRL	Data
1	1F0h	30F0Fh	0	8	3Ch, 3Ch, 3Ch, 3Ch, 3Ch, 3Ch, 3Ch, 3Ch
2	1F0h	0F0F0h	0	1	00
3	7EEh	0	0	01h	A0h

For a specific devic, all possible dominant and recessive stuff bits inside and at the end of each stuffed field shall be considered.

Three elementary tests shall be performed.

8.1.7.2 Test case organization

Test case organization shall be in accordance with Table 85.

Table 85 — Stuff bit generation capability in extended frame — Test case organization

State	Description
Set-up	No action required. The IUT is left in the default state.
Test	The LT causes the IUT to transmit a frame according to Table 84.
Verification	The IUT shall not generate any error flag during the test. The IUT shall correctly generate all stuff bits.

8.2 Test class 2, error detection

8.2.1 Bit error in standard frame test

8.2.1.1 Purpose and limits of test case

This test is applicable to CAN_VERSION $\in$ {A, B, BP}.

It is used to verify that the IUT detects a bit error when the bit it is transmitting in a standard frame is different with the bit it receives.

The test shall, at minimum, modify at least one dominant and one recessive bit in each field of the frame except for the arbitration field for which only dominant bits shall be modified. The ACK slot is not tested.

8.2.1.2 Test case organization

Test case organization shall be in accordance with Table 86.

Table 86 — Bit error in standard frame test — Test case organization

State	Description
Set-up	No action required. The IUT is left in the default state.
Test	The LT causes the IUT to transmit the frames and creates a bit error according to 8.2.1.1.
Verification	The IUT shall generate an active error frame starting at the bit position following the corrupted bit. The IUT shall restart the transmission of the data frame as soon as the bus is idle.

8.2.2 Bit error in extended frame test

8.2.2.1 Purpose and limits of test case

This test is applicable to CAN_VERSION $\in$ {B}.

It is used to verify that the IUT detects a bit error when the bit it is transmitting in an extended frame is different with the bit it receives.

The test shall modify at least one dominant extended identifier bit and the two reserved bits.

8.2.2.2 Test case organization

Test case organization shall be in accordance with Table 87.

Table 87 — Bit error in extended frame test — Test case organization

State	Description
Set-up	No action required. The IUT is left in the default state.
Test	The LT causes the IUT to transmit the frames and creates a bit error according to 8.2.2.1.
Verification	The IUT shall generate an active error frame starting at the bit position following the corrupted bit. The IUT shall restart the transmission of the data frame as soon as the bus is idle.

8.2.3 Stuff error test in standard frame

8.2.3.1 Purpose and limits of test case

This test is applicable to CAN_VERSION $\in$ {A, B, BP}.

It is used to verify that the IUT detects an error when after the transmission of five identical bits it receives a sixth bit identical to the five precedents. This test is made of a standard format frame.

The frames given in Table 88 shall be used for the elementary tests.

Table 88 — Stuff error test in standard frame — Frames for elementary tests

Frame	IDEN	RTR	CTRL	Data
1	078h	0	8	01h, E1h, E1h, E1h, E1h, E1h, E1h, E1h
2	41Fh	0	1	00
3	47Fh	0	01h	1Fh
4	758h	0	00h	—
5	777h	0	01h	1Fh
6	7EFh	1	02h	—

For an open device, at least one stuff error shall be generated at each stuffed field.

For a specific device, at least one stuff error shall be generated at each stuffed field, where a stuff bit can occur.

8.2.3.2 Test case organization

Test case organization shall be in accordance with Table 89.

Table 89 — Stuff error test in standard frame — Test case organization

State	Description
Set-up	No action required. The IUT is left in the default state.
Test	The LT causes the IUT to transmit the frames and create a stuff error according to 8.2.3.1.
Verification	The IUT shall generate an error frame at the bit position following the corrupted stuff bit. The IUT shall restart the transmission of the data frame as soon as the bus is idle.

8.2.4 Stuff error test in extended frame

8.2.4.1 Purpose and limits of test case

This test is applicable to CAN_VERSION ∈ {B}.

It is used to verify that the IUT detects an error when after the transmission of five identical bits it receives a sixth bit identical to the five preceding bits. This test is made on an extended frame.

The frames given in Table 90 shall be used for the elementary tests.

Table 90 — Stuff error test in extended frame — Frames used at elementary tests

Frame	IDEN 1	IDEN 2	RTR	CTRL	Data
1	1F0h	30F0Fh	0	8	3Ch, 3Ch, 3Ch, 3Ch, 3Ch, 3Ch, 3Ch, 3Ch
2	1F0h	0F0F0h	0	1	00
3	7EEh	0	0	01h	A0h

For an open device, at least one stuff error shall be generated at each stuffed field.

For a specific device, at least one stuff error shall be generated at each stuffed field, where a stuff bit can occur.

8.2.4.2 Test case organization

Test case organization shall be in accordance with Table 91.

Table 91 — Stuff error test in extended frame — Test case organization

State	Description
Set-up	No action required. The IUT is left in the default state.
Test	The LT causes the IUT to transmit the frames and create a stuff error according to 8.2.4.1.
Verification	The IUT shall generate an error frame at the bit position following the corrupted stuff bit. The IUT shall restart the transmission of the data frame as soon as the bus is idle.

8.2.5 Form error

8.2.5.1 Purpose and limits of test case

This test is applicable to CAN_VERSION ∈ {A, B, BP}.

It is used to verify that the IUT detects a form error if one of the following fields, transmitted by the IUT, contains a dominant bit:

— CRC delimiter;

— ACK delimiter;

— EOF (the first, the fourth and the last one).

Five elementary tests shall be performed.

8.2.5.2 Test case organization

Test case organization shall be in accordance with Table 92.

Table 92 — Form error — Test case organization

State	Description
Set-up	No action required. The IUT is left in the default state.
Test	The LT causes the IUT to transmit a frame. Then the LT creates a form error on the fields listed in 8.2.5.1 by corrupting the recessive bit of these fields.
Verification	The IUT shall generate an error frame at the bit position following the corrupted bit. The IUT shall restart the transmission of the frame as soon as the bus is idle.

8.2.6 Acknowledgement error

8.2.6.1 Purpose and limits of test case

This test is applicable to CAN_VERSION $\in$ {A, B, BP}.

It is used to verify that the IUT detects an acknowledgement error when the received ACK slot is recessive.

One elementary test shall be performed.

8.2.6.2 Test case organization

Test case organization shall be in accordance with Table 93.

Table 93 — Acknowledgement error — Test case organization

State	Description
Set-up	No action required. The IUT is left in the default state.
Test	The LT causes the IUT to transmit a standard frame. Then the LT does not send a dominant bit in the ACK slot.
Verification	The IUT shall generate an error frame starting at the bit position following the ACK slot. The IUT shall restart the transmission of the frame as soon as the bus is idle.

8.3 Test class 3, active error frame management

8.3.1 Error flag longer than 6 bits

8.3.1.1 Purpose and limits of test case

This test is applicable to CAN_VERSION $\in$ {A, B, BP}.

It is used to verify that an IUT acting as a transmitter tolerates up to seven dominant bits after sending its own error flag.

Three elementary tests shall be performed, lengthening the error flag by one, four and seven dominant bits.

8.3.1.2 Test case organization

Test case organization shall be in accordance with Table 94.

Table 94 — Error flag longer than 6 bits — Test case organization

State	Description
Set-up	No action required. The IUT is left in the default state.
Test	The LT causes the IUT to transmit a frame. The LT corrupts this frame causing the IUT to send an active error frame. The LT prolongs the error flag send by IUT according to 8.3.1.1.
Verification	The IUT shall generate only 1 error frame. The IUT shall restart the transmission after the intermission field following the error frame.

8.3.2 Transmission on the third bit of intermission field

8.3.2.1 Purpose and limits of test case

This test is applicable to CAN_VERSION $\in$ {A, B, BP}.

Its purpose is to verify that an IUT is able to transmit a frame on reception of a SOF starting at the third bit of the intermission field following the error frame it has transmitted.

For open devices, the identifier shall start with four dominant bits.

For a specific device which can not send such an identifier, any other value may be used.

One elementary test shall be performed.

8.3.2.2 Test case organization

Test case organization shall be in accordance with Table 95.

Table 95 — Transmission on the third bit of intermission field — Test case organization

State	Description
Set-up	No action required. The IUT is left in the default state.
Test	The LT causes the IUT to transmit a frame according to 8.3.2.1. The LT corrupts this frame causing the IUT to send an active error frame. At the end of the error flag sent by the IUT, the LT waits for $(8 + 2)$ bit times before sending a SOF.
Verification	The IUT shall repeat the frame starting with the identifier without transmitting any SOF.

8.3.3 Bit error in error flag

8.3.3.1 Purpose and limits of test case

This test is applicable to CAN_VERSION $\in$ {A, B, BP}.

It is used to verify that an IUT acting as a transmitter detects a bit error when one of the six dominant bits of the error flag it transmits is forced to recessive state by LT.

Three elementary tests shall be performed, corrupting the first, fourth and sixth bit of the error flag.

8.3.3.2 Test case organization

Test case organization shall be in accordance with Table 96.

Table 96 — Bit error in error flag — Test case organization

State	Description
Set-up	No action required. The IUT is left in the default state.
Test	The LT causes the IUT to transmit a frame. The LT corrupts this frame causing the IUT to send an active error frame. Then the LT forces one of the 6 bits of the active-error flag sent by the IUT to recessive state.
Verification	The IUT shall restart its active-error flag at the bit position following the corrupted bit.

8.3.4 Form error in error delimiter

8.3.4.1 Purpose and limits of test case

This test is applicable to CAN_VERSION $\in$ {A, B, BP}.

It is used to verify that an IUT acting as a transmitter detects a form error when it receives an invalid error delimiter.

The LT replaces one of the eight recessive bits of the error delimiter by a dominant bit.

Three elementary tests shall be performed, corrupting the second, fourth and seventh bit of the error delimiter.

8.3.4.2 Test case organization

Test case organization shall be in accordance with Table 97.

Table 97 — Form error in error delimiter — Test case organization

State	Description
Set-up	No action required. The IUT is left in the default state.
Test	The LT causes the IUT to transmit a frame. The LT corrupts this frame causing the IUT to send an active error frame. Then the LT forces one recessive bit of the error delimiter to the dominant state according to 8.3.4.1.
Verification	The IUT shall restart the error frame at the bit position following the corrupted bit.

8.4 Test class 4, overload frame management

8.4.1 MAC overload generation in intermission field

8.4.1.1 Purpose and limits of test case

This test is applicable to CAN_VERSION $\in$ {A, B, BP}.

It is used to verify that an IUT acting as a transmitter generates an overload frame when detecting a dominant bit on one of the two first recessive bits of the intermission field following a data frame it is transmitting.

Two elementary tests shall be performed.

8.4.1.2 Test case organization

Test case organization shall be in accordance with Table 98.

Table 98 — MAC overload generation in intermission field — Test case organization

State	Description
Set-up	The IUT is set to the TEC passive state.
Test	The LT causes the IUT to transmit a frame. Then the LT forces one of the first 2 bits of the intermission field to the dominant state.
Verification	The IUT shall generate an overload frame starting at the bit position following the dominant bit generated by the LT.

8.4.2 Eighth bit of an error and overload delimiter

8.4.2.1 Purpose and limits of test case

This test is applicable to CAN_VERSION $\in$ {A, B, BP}.

It is used to verify that an IUT acting as a transmitter generates an overload frame when detecting a dominant bit on the eighth bit of an error or an overload delimiter it is transmitting.

Two elementary tests shall be performed.

8.4.2.2 Test case organization

Test case organization shall be in accordance with Table 99.

Table 99 — Eighth bit of an error and overload delimiter — Test case organization

State	Description
Set-up	The IUT is set to the TEC passive state.
Test	The LT causes the IUT to transmit a frame. Then the LT causes the IUT to generate an error frame or overload frame. Then the LT forces the eighth bit of the delimiter to a dominant state.
Verification	The IUT shall generate an overload frames starting at the bit position following the dominant bit sent by the LT.

8.4.3 Transmission on third bit of intermission field

8.4.3.1 Purpose and limits of test case

This test is applicable to CAN_VERSION $\in$ {A, B, BP}.

Its purpose is to verify that an IUT is able to transmit a data frame starting with the identifier and without transmitting SOF, when detecting a dominant bit on the third bit of the intermission field following an overload frame.

For open devices, the identifier shall start with four dominant bits.

For a specific device which can not send such an identifier, any other value may be used.

One elementary test shall be performed.

8.4.3.2 Test case organization

Test case organization shall be in accordance with Table 100.

Table 100 — Transmission on third bit of intermission field — Test case organization

State	Description
Set-up	No action required. The IUT is left in the default state.
Test	The LT causes the IUT to transmit a frame according to 8.4.3.1. Then the LT causes the IUT to generate an overload frame. Then the LT forces the third bit of the intermission following the overload delimiter to dominant state.
Verification	The IUT shall repeat the frame starting with the identifier without transmitting any SOF.

8.4.4 Bit error in overload flag

8.4.4.1 Purpose and limits of test case

This test is applicable to CAN_VERSION $\in$ {A, B, BP}.

It is used to verify that an IUT acting as a transmitter detects a bit error when one of the six dominant bits of the overload flag it transmits is forced to recessive state by LT.

Three elementary tests shall be performed, corrupting the first, second and sixth bit of the overload flag.

8.4.4.2 Test case organization

Test case organization shall be in accordance with Table 101.

Table 101 — Bit error in overload flag — Test case organization

State	Description
Set-up	No action required. The IUT is left in the default state.
Test	The LT causes the IUT to transmit a frame. Then the LT causes the IUT to generate an overload frame. Then the LT corrupts one of the 6 dominant bits of the overload flag to the recessive state according to 8.4.4.1.
Verification	The IUT shall generate an error frame starting at the bit position after the corrupted bit.

8.4.5 Form error in overload delimiter

8.4.5.1 Purpose and limits of test case

This test is applicable to CAN_VERSION $\in$ {A, B, BP}.

It is used to verify that an IUT acting as a transmitter detects a form error when it receives an invalid overload delimiter.

The LT replaces one of the eight recessive bits of the overload delimiter by a dominant bit.

Three elementary tests shall be performed, corrupting the second, fourth and seventh bit of the overload delimiter.

8.4.5.2 Test case organization

Test case organization shall be in accordance with Table 102.

Table 102 — Form error in overload delimiter — Test case organization

State	Description
Set-up	No action required. The IUT is left in the default state.
Test	The LT causes the IUT to transmit a frame. Then the LT causes the IUT to generate an overload frame. The LT corrupts the overload delimiter according to 8.4.5.1.
Verification	The IUT shall generate an error frame starting at the bit position following the corrupted bit.

8.5 Test class 5, passive-error state and bus-off

8.5.1 Acceptance of active-error flag overwriting passive-error flag

8.5.1.1 Purpose and limits of test case

This test is applicable to CAN_VERSION $\in$ {A, B, BP}.

Its purpose is to verify that a passive state IUT acting as a transmitter does not detect any error when detecting an active-error flag during its own passive-error flag.

Three elementary tests shall be performed, superposing the passive-error flag by an active-error flag starting at the first, third and sixth bit.

8.5.1.2 Test case organization

Test case organization shall be in accordance with Table 103.

Table 103 — Acceptance of active-error flag overwriting passive-error flag — Test case organization

State	Description
Set-up	The IUT is set to the TEC passive state.
Test	The LT causes the IUT to transmit a frame. Then the LT causes the IUT to send a passive-error flag. During the passive-error flag sent by the IUT, the LT sends an active-error flag according to 8.4.1.1. At the end of the error flag the LT waits for (8 + 3) bit time before sending a frame.
Verification	The IUT shall acknowledge the last frame transmitted by the LT.

8.5.2 Frame acceptance after passive-error frame transmission

8.5.2.1 Purpose and limits of test case

This test is applicable to CAN_VERSION $\in$ {A, B, BP}.

Its purpose is to verify that a passive state IUT acting as a transmitter accepts to receive a frame starting after the second bit of the intermission following the error frame it has transmitted.

One elementary test shall be performed.

8.5.2.2 Test case organization

Test case organization shall be in accordance with Table 104.

Table 104 — Frame acceptance after passive-error frame transmission — Test case organization

State	Description
Set-up	The IUT is set to the TEC passive state.
Test	The LT causes the IUT to transmit a frame. Then the LT causes the IUT to send a passive-error flag. During the passive-error flag sent by the IUT, the LT sends an active-error flag. At the end of the error flag the LT waits for (8 + 2) bit time before sending a frame.
Verification	The IUT shall acknowledge the frame.

8.5.3 Acceptance of 7 consecutive dominant bits after passive-error flag

8.5.3.1 Purpose and limits of test case

This test is applicable to CAN_VERSION ∈ {A, B, BP}.

Its purpose is to verify that a passive state IUT acting as a transmitter does not detect any error when detecting dominant bits during the seven first bits of the error delimiter.

Three elementary tests shall be performed, transmitting one, four or seven consecutive dominant bits.

8.5.3.2 Test case organization

Test case organization shall be in accordance with Table 105.

Table 105 — Acceptance of 7 consecutive dominant bits after passive-error flag — Test case organization

State	Description
Set-up	The IUT is set to the TEC passive state.
Test	The LT causes the IUT to transmit a data frame. Then the LT causes the IUT to send a passive-error flag. At the end of error flag, the LT continues transmitting dominant bits according to 8.5.3.1. At this step the LT waits for (8+3) bit time before sending a frame.
Verification	The IUT shall acknowledge the frame transmitted by the LT. The IUT shall restart the transmission of the corrupted frame (1+7+3+8) bit time after its ACK bit.

8.5.4 Reception of frame during suspend transmission field

8.5.4.1 Purpose and limits of test case

This test is applicable to CAN_VERSION ∈ {A, B, BP}.

Its purpose is to verify that a passive state IUT acting as a transmitter is able to receive a frame during the suspend transmission field.

Three elementary tests shall be performed, for which the received frame starts on the first, fourth and eighth bit of the suspend transmission field.

8.5.4.2 Test case organization

Test case organization shall be in accordance with Table 106.

Table 106 — Reception of frame during suspend transmission field — Test case organization

State	Description
Set-up	The IUT is set to the TEC passive state.
Test	The LT causes the IUT to transmit a frame. At the end of the EOF and intermission fields, the LT sends a frame according to 8.5.4.1.
Verification	The IUT shall acknowledge the last frame transmitted by the LT.

8.5.5 Transmission of frame after suspend transmission field — Test case 1

8.5.5.1 Purpose and limits of test case

This test is applicable to CAN_VERSION $\in$ {A, B, BP}.

Its purpose is to verify that a passive state IUT acting as a transmitter does not transmit any frame before the end of the Suspend Transmission field following an error frame.

One elementary test shall be performed.

8.5.5.2 Test case organization

Test case organization shall be in accordance with Table 107.

Table 107 — Transmission of frame after suspend transmission field — Test case 1 organization

State	Description
Set-up	The IUT is set to the TEC passive state.
Test	The LT causes the IUT to transmit a frame. Then the LT causes the IUT to send a passive-error frame.
Verification	After the intermission following the error frame, the LT verifies that the IUT waits 8 more bits before re-transmitting the frame.

8.5.6 Transmission of frame after suspend transmission field — Test case 2

8.5.6.1 Purpose and limits of test case

This test is applicable to CAN_VERSION $\in$ {A, B, BP}.

Its purpose is to verify that a passive state IUT being transmitter does not transmit any data frame before the end of the suspend transmission field following an overload frame.

One elementary test shall be performed.

8.5.6.2 Test case organization

Test case organization shall be in accordance with Table 108.

Table 108 — Transmission of frame after suspend transmission field — Test case 2 organization

State	Description
Set-up	The IUT is set to the TEC passive state.
Test	The LT causes the IUT to transmit 2 frames. The LT lets the IUT transmit the first frame and causes the IUT to generate an overload frame.
Verification	After the intermission following the overload frame, the LT verifies that the IUT waits 8 more bits before transmitting the second frame.

8.5.7 Transmission of frame after suspend transmission field — Test case 3

8.5.7.1 Purpose and limits of test case

This test is applicable to CAN_VERSION $\in$ {A, B, BP}.

Its purpose is to verify that a passive state IUT acting as a transmitter does not transmit any frame before the end of the suspend transmission field following a frame.

One elementary test shall be performed.

8.5.7.2 Test case organization

Test case organization shall be in accordance with Table 109.

Table 109 —Transmission of frame after suspend transmission field — Test case 3 organization

State	Description
Set-up	The IUT is set to the TEC passive state.
Test	The LT causes the IUT to transmit 2 frames. The LT lets the IUT transmit the first frame. This frame shall end with EOF field followed by the intermission field.
Verification	After the intermission following the first frame, the LT verifies that the IUT waits eight more bits before transmitting the second frame.

8.5.8 Transmission of frame without suspend transmission field

8.5.8.1 Purpose and limits of test case

This test is applicable to CAN_VERSION $\in$ {A, B, BP}.

Its purpose is to verify that a passive state IUT, after losing arbitration, repeats the frame without inserting any suspend transmission field.

One elementary test shall be performed.

8.5.8.2 Test case organization

Test case organization shall be in accordance with Table 110.

Table 110 — Transmission of frame without suspend transmission field — Test case organization

State	Description
Set-up	The IUT is set to the TEC passive state.
Test	The LT causes the IUT to transmit a frame. The LT causes the IUT to loose arbitration by sending a frame of higher priority.
Verification	The LT verifies that the IUT re-transmits its frame (1 + 7 + 3) bit times after acknowledging the received frame.

8.5.9 No-transmission of frame on the third bit of intermission field

8.5.9.1 Purpose and limits of test case

This test is applicable to CAN_VERSION ∈ {A, B, BP}.

Its purpose is to verify that a passive state IUT does not transmit a frame starting with an identifier and without transmitting SOF, when detecting a dominant bit on the third bit of the intermission field.

One elementary test shall be performed.

8.5.9.2 Test case organization

Test case organization shall be in accordance with Table 111.

Table 111 — No-transmission of frame on the third bit of intermission field — Test case organization

State	Description
Set-up	The IUT is set to the TEC passive state.
Test	The LT causes the IUT to transmit 2 frames. The LT lets the IUT transmit the first frame. This frame shall end with EOF field followed by the intermission field. At the third bit of the intermission field, the LT starts sending a frame with the lowest priority.
Verification	The IUT shall not start the second transmission before the end of the frame sent by the LT.

8.5.10 Bus-off state

8.5.10.1 Purpose and limits of test case

This test is applicable to CAN_VERSION ∈ {A, B, BP}.

Its purpose is to verify that an IUT switching to bus-off state no longer sends dominant bits.

One elementary test shall be performed.

8.5.10.2 Test case organization

Test case organization shall be in accordance with Table 112.

Table 112 — Bus-off state — Test case organization

State	Description
Set-up	No action required. The IUT is left in the default state.
Test	The LT causes the IUT to transmit a frame twice. The LT causes the IUT to generate an error frame. During the error flag transmitted by the IUT, the LT forces recessive state during 16 bit times and then dominant state for 112 bit times. Then the IUT transmits its first frame. The LT acknowledges the frame and immediately causes the IUT to generate an overload frame. The LT forces the first bit of this overload flag to recessive state creating a bit error. (6 + 7) bit times later, the LT generates a dominant bit to cause the IUT to generate a new overload frame; the LT forces the first bit of this new overload flag to recessive state causing the IUT to increment its TEC to the bus-off limit. (6 + 8 + 3 + 8) bit times later, the LT sends a frame.
Verification	Only 1 frame shall be transmitted by the IUT. The IUT shall not acknowledge the frame sent by the LT.

8.5.11 Bus-off recovery

8.5.11.1 Purpose and limits of test case

This test is applicable to CAN_VERSION $\in$ {A, B, BP}.

Its purpose is to verify that an IUT which is bus-off is not permitted to become error active (no longer bus off) before 128 occurrences of 11 consecutive recessive bits.

Two elementary tests shall be performed. In the first, the LT sends recessive bus level for at least 1408 bit times until the IUT becomes active again. In the second, the LT sends 1 group of 10 recessive bits, and 1 group of 21 recessive bits followed by at least 127 groups of 11 recessive bits, each group being separated by one dominant bit.

8.5.11.2 Test case organization

Test case organization shall be in accordance with Table 113.

Table 113 — Bus-off recovery — Test case organization

State	Description
Set-up	No action required. The IUT is left in the default state.
Test	The LT ask the IUT to send a frame and sets it in the Bus-off state The LT sends the profiles specified in 8.5.11.1.
Verification	The IUT shall not transmit the frame before the end of the profiles sent by the LT and shall send it before the end of the timeout.

8.5.12 Completion condition for passive-error flag

8.5.12.1 Purpose and limits of test case

This test is applicable to CAN_VERSION $\in$ {A, B, BP}.

Its purpose is to verify that a passive state IUT acting as a transmitter waits for six consecutive identical bits to complete its passive-error flag.

One elementary test shall be performed.

8.5.12.2 Test case organization

Test case organization shall be in accordance with Table 114.

Table 114 — Completion condition for passive-error flag — Test case organization

State	Description
Set-up	The IUT is set to the TEC passive state.
Test	The LT causes the IUT to transmit a frame. Then the LT causes the IUT to generate a passive-error flag. During the error flag, the LT sends 5 dominant bits, 5 recessive bits and then 6 dominant bits. After the 6 dominant bits, the LT waits for 8 bit time before sending a dominant bit.
Verification	The IUT shall generate an overload frame starting at the bit position following the last dominant bit generated by the LT.

8.5.13 Form error in passive-error delimiter

8.5.13.1 Purpose and limits of test case

This test is applicable to CAN_VERSION $\in$ {A, B, BP}.

Its purpose is to verify that an error-passive IUT acting as a transmitter detects a form error when monitoring a corruption in the error delimiter.

The LT forces one of the eight recessive bits of the error delimiter to a dominant state.

Three elementary tests shall be performed, corrupting the second, fourth and seventh bit of the error delimiter.

8.5.13.2 Test case organization

Test case organization shall be in accordance with Table 115.

Table 115 — Form error in passive-error delimiter — Test case organization

State	Description
Set-up	The IUT is set to the TEC passive state.
Test	The LT causes the IUT to transmit a data frame. Then the LT causes the IUT to generate a passive-error frame. The LT creates a form error according to 8.5.13.1. After the form error, the LT waits for (6 + 7) bit time before sending a dominant bit.
Verification	The IUT shall generate an overload frame starting at the bit position following the last dominant bit generated by the LT.

8.5.14 Maximum recovery time after corrupted frame

8.5.14.1 Purpose and limits of test case

This test is applicable to CAN_VERSION $\in$ {A, B, BP}.

Its purpose is to verify that the recovery time of an error-passive IUT detecting an error is at most 31 bit times.

One elementary test shall be performed.

8.5.14.2 Test case organization

Test case organization shall be in accordance with Table 116.

Table 116 — Maximum recovery time after corrupted frame — Test case organization

State	Description
Set-up	The IUT is set to the passive state.
Test	The LT causes the IUT to transmit a frame. Then the LT corrupts a dominant bit of this frame, causing the IUT to generate a passive-error flag. At the bit position following the end of the passive-error flag, the LT starts to send 6 dominant bits.
Verification	The IUT shall re-transmit the same frame 31 bit times after the detection of the corrupted bit.

8.6 Test class 6, error counter management

8.6.1 TEC increment on bit error during active-error flag

8.6.1.1 Purpose and limits of test case

This test is applicable to CAN_VERSION $\in$ {A, B, BP}.

It is used to verify that an IUT acting as a transmitter increases its TEC by 8 when detecting a bit error during the transmission of an active-error flag.

Three elementary tests shall be performed, corrupting the first, third and sixth bit of the active-error flag.

8.6.1.2 Test case organization

Test case organization shall be in accordance with Table 117.

Table 117 — TEC increment on bit error during active-error flag — Test case organization

State	Description
Set-up	No action required. The IUT is left in the default state.
Test	The LT causes the IUT to transmit a frame. Then the LT causes the IUT to generate an active error frame. The LT corrupts 1 of the above mentioned dominant bits of the error flag.
Verification	The TEC is increased by 8 on the corrupted bit.

8.6.2 TEC increment on bit error during overload flag

8.6.2.1 Purpose and limits of test case

This test is applicable to CAN_VERSION $\in$ {A, B, BP}.

It is used to verify that an IUT acting as a transmitter increases its TEC by 8 when detecting a bit error during the transmission of an overload flag.

Three elementary tests shall be performed, corrupting the first, fourth and sixth bit of the overload flag.

8.6.2.2 Test case organization

Test case organization shall be in accordance with Table 118.

Table 118 — TEC increment on bit error during overload flag — Test case organization

State	Description
Set-up	No action required. The IUT is left in the default state.
Test	The LT causes the IUT to transmit a frame. Then the LT causes the IUT to generate an overload frame. The LT corrupts 1 of the above mentioned dominant bits of the overload flag.
Verification	The TEC is increased by 8 at the corrupted bit.

8.6.3 TEC increment when active-error flag followed by dominant bits

8.6.3.1 Purpose and limits of test case

This test is applicable to CAN_VERSION $\in$ {A, B, BP}.

It is used to verify that an IUT acting as a transmitter increases its TEC by 8 when detecting eight consecutive dominant bits following the transmission of its active-error flag and after each sequence of additional eight consecutive dominant bits.

One elementary test shall be performed.

8.6.3.2 Test case organization

Test case organization shall be in accordance with Table 119.

Table 119 — TEC increment when active-error flag followed by dominant bits — Test case organization

State	Description
Set-up	No action required. The IUT is left in the default state.
Test	The LT causes the IUT to transmit a frame. Then the LT causes the IUT to generate an active error frame. After the error flag sent by the IUT, the LT sends a sequence of 16 dominant bits.
Verification	The TEC is increased by 8 on each eighth dominant bit after the error flag.

8.6.4 TEC increment when passive-error flag followed by dominant bits

8.6.4.1 Purpose and limits of this test case

This test is applicable to CAN_VERSION $\in$ {A, B, BP}.

It is used to verify that an IUT acting as a transmitter increases its TEC by 8 when detecting eight consecutive dominant bits following the transmission of its passive-error flag and after each sequence of additional eight consecutive dominant bits.

One elementary test shall be performed.

8.6.4.2 Test case organization

Test case organization shall be in accordance with Table 120.

Table 120 — TEC increment when passive-error flag followed by dominant bits — Test case organization

State	Description
Set-up	The IUT is set to the TEC passive state.
Test	The LT causes the IUT to transmit a frame. Then the LT causes the IUT to generate a passive-error frame. After the error flag sent by the IUT, the LT sends a sequence of 16 dominant bits.
Verification	The TEC is increased by 8 on each eighth dominant bit after the error flag.

8.6.5 TEC increment when overload flag followed by dominant bits

8.6.5.1 Purpose and limits of test case

This test is applicable to CAN_VERSION $\in$ {A, B, BP}.

It is used to verify that an IUT acting as a transmitter increases its TEC by 8 when detecting eight consecutive dominant bits following the transmission of its overload flag and after each sequence of additional eight consecutive dominant bits.

One elementary test shall be performed.

8.6.5.2 Test case organization

Test case organization shall be in accordance with Table 121.

Table 121 — TEC increment when overload flag followed by dominant bits — Test case organization

State	Description
Set-up	No action required. The IUT is left in the default state.
Test	The LT causes the IUT to transmit a frame. Then the LT causes the IUT to generate an overload frame. After the overload flag sent by the IUT, the LT sends a sequence of 23 dominant bits.
Verification	The TEC is increased by 8 on each eighth dominant bit after the error flag.

8.6.6 TEC increment on bit error in data frame

8.6.6.1 Purpose and limits of test case

This test is applicable to CAN_VERSION $\in$ {A, B, BP}.

It is used to verify that an IUT acting as a transmitter increases its TEC by 8 when detecting a bit error in a data frame on one of the following fields it transmits: SOF, arbitration, control, data and CRC fields.

In the arbitration field, only bit error on dominant bits shall be considered.

Five elementary tests shall be performed.

8.6.6.2 Test case organization

Test case organization shall be in accordance with Table 122.

Table 122 — TEC increment on bit error in data frame — Test case organization

State	Description
Set-up	No action required. The IUT is left in the default state.
Test	The LT causes the IUT to transmit a frame. Then the LT corrupts a bit according to 8.6.6.1.
Verification	The TEC is increased by 8 at the bit error detection.

8.6.7 TEC increment on form error in a frame

8.6.7.1 Purpose and limits of test case

This test is applicable to CAN_VERSION $\in$ {A, B, BP}.

It is used to verify that an IUT acting as a transmitter increases its TEC by 8 when detecting a form error in a frame on one of the following fields it transmits: CRC delimiter, acknowledge delimiter, EOF (first, fourth and last bit).

Five elementary tests shall be performed.

8.6.7.2 Test case organization

Test case organization shall be in accordance with Table 123.

Table 123 — TEC increment on form error in a frame — Test case organization

State	Description
Set-up	No action required. The IUT is left in the default state.
Test	The LT causes the IUT to transmit a frame. Then the LT corrupts a bit according to 8.6.7.1.
Verification	The TEC is increased by 8 at the form error detection.

8.6.8 TEC increment on acknowledgement error

8.6.8.1 Purpose and limits of test case

This test is applicable to CAN_VERSION $\in$ {A, B, BP}.

It is used to verify that an active state IUT acting as a transmitter increases its TEC by 8 when detecting an acknowledgement error in a frame.

One elementary test shall be performed.

8.6.8.2 Test case organization

Test case organization shall be in accordance with Table 124.

Table 124 — TEC increment on acknowledgement error — Test case organization

State	Description
Set-up	No action required. The IUT is left in the default state.
Test	The LT causes the IUT to transmit a frame. Then the LT does not acknowledge the frame.
Verification	The TEC is increased by 8 at the Acknowledgement error detection.

8.6.9 TEC increment on form error in error delimiter

8.6.9.1 Purpose and limits of test case

This test is applicable to CAN_VERSION $\in$ {A, B, BP}.

It is used to verify that an IUT acting as a transmitter increases its TEC by 8 when detecting a form error on a bit of the error delimiter it is transmitting.

Three elementary tests shall be performed, corrupting the first, fourth and sixth bit of the error delimiter.

8.6.9.2 Test case organization

Test case organization shall be in accordance with Table 125.

Table 125 — TEC increment on form error in error delimiter — Test case organization

State	Description
Set-up	No action required. The IUT is left in the default state.
Test	The LT causes the IUT to transmit a frame. Then the LT causes the IUT to generate an error frame. The LT corrupts the error delimiter according to 8.6.9.1.
Verification	The TEC is increased by 8 at the corrupted bit.

8.6.10 TEC increment on form error in overload delimiter

8.6.10.1 Purpose and limits of this test case

This test is applicable to CAN_VERSION $\in$ {A, B, BP}.

It is used to verify that an IUT acting as a transmitter increases its TEC by 8 when detecting a form error during the transmission of an overload delimiter.

Two elementary tests shall be performed, corrupting the second and seventh bit of the overload delimiter.

8.6.10.2 Test case organization

Test case organization shall be in accordance with Table 126.

Table 126 — TEC increment on form error in overload delimiter — Test case organization

State	Description
Set-up	No action required. The IUT is left in the default state.
Test	The LT causes the IUT to transmit a frame. Then the LT causes the IUT to generate an overload frame. The LT corrupts 1 of the above-mentioned recessive bits of the overload delimiter.
Verification	The TEC is increased by 8 at the corrupted bit.

8.6.11 TEC decrement on successful frame transmission for TEC $<$ 128

8.6.11.1 Purpose and limits of test case

This test is applicable to CAN_VERSION $\in$ {A, B, BP}.

It is used to verify that an active state IUT decreases its TEC by 1 when transmitting a valid frame.

One elementary test shall be performed.

8.6.11.2 Test case organization

Test case organization shall be in accordance with Table 127.

Table 127 — TEC decrement on successful frame transmission for TEC < 128 — Test case organization

State	Description
Set-up	The LT forces the IUT to increase its TEC
Test	The LT causes the IUT to transmit a frame. The LT acknowledges this frame.
Verification	The TEC is decreased by 1 after the frame is completed.

8.6.12 TEC decrement on successful frame transmission for TEC $>$ 127

8.6.12.1 Purpose and limits of test case

This test is applicable to CAN_VERSION $\in$ {A, B, BP}.

It is used to verify that a passive state IUT decreases its TEC by 1 when transmitting a valid data frame.

One elementary test shall be performed.

8.6.12.2 Test case organization

Test case organization shall be in accordance with Table 128.

Table 128 — TEC decrement on successful frame transmission for TEC > 127 **— Test case organization**

State	Description
Set-up	The IUT is set to the TEC passive state.
Test	The LT causes the IUT to transmit a frame. The LT acknowledges this frame.
Verification	The TEC is decreased by 1 after the frame is completed.

8.6.13 TEC non-increment on 13 bit long overload flag

8.6.13.1 Purpose and limits of test case

This test is applicable to CAN_VERSION ∈ {A, B, BP}.

It is used to verify that an IUT acting as a transmitter does not change the value of its TEC when receiving a 13-bit long overload flag.

One elementary test shall be performed.

8.6.13.2 Test case organization

Test case organization shall be in accordance with Table 129.

Table 129 — TEC non-increment on 13-bit long overload flag — Test case organization

State	Description
Set-up	The LT forces the IUT to increase its TEC.
Test	The LT causes the IUT to transmit a frame. After the last bit of the EOF, the LT sends a sequence of 14 dominant bits.
Verification	The TEC shall equal the set-up value decreased by 1.

8.6.14 TEC non-increment on 13 bit long error flag

8.6.14.1 Purpose and limits of test case

This test is applicable to CAN_VERSION ∈ {A, B, BP}.

It is used to verify that an IUT acting as a transmitter does not change the value of its TEC when monitoring an error flag with 13 bit length.

One elementary test shall be performed.

8.6.14.2 Test case organization

Test case organization shall be in accordance with Table 130.

Table 130 — TEC non-increment on 13 bit long error flag — Test case organization

State	Description
Set-up	No action required. The IUT is left in the default state.
Test	The LT causes the IUT to transmit a frame and causes the IUT to send an active-error flag. After the last bit of the error flag, the LT sends a sequence of 7 dominant bits.
Verification	The TEC value shall be 8.

8.6.15 TEC non-increment on form error at last bit of overload delimiter

8.6.15.1 Purpose and limits of test case

This test is applicable to CAN_VERSION ∈ {A, B, BP}

It is used to verify that an IUT acting as a transmitter does not change the value of its TEC when detecting a form error on the last bit of the overload delimiter it is transmitting.

One elementary test shall be performed.

8.6.15.2 Test case organization

Test case organization shall be in accordance with Table 131.

Table 131 — TEC non-increment on form error at last bit of overload delimiter — Test case organization

State	Description
Set-up	The LT forces the IUT to increase its TEC.
Test	The LT causes the IUT to transmit a frame. Then the LT causes the IUT to generate an overload frame. At the last bit of the overload delimiter, the LT sends 1 dominant bit.
Verification	The TEC shall equal the set-up value.

8.6.16 TEC non-increment on form error at last bit of error delimiter

8.6.16.1 Purpose and limits of test case

This test is applicable to CAN_VERSION ∈ {A, B, BP}

It is used to verify that an IUT acting as a transmitter does not change the value of its TEC when detecting a form error on the last bit of the error delimiter it is transmitting.

One elementary test shall be performed.

8.6.16.2 Test case organization

Test case organization shall be in accordance with Table 132.

Table 132 — TEC non-increment on form error at last bit of error delimiter — Test case organization

State	Description
Set-up	The LT forces the IUT to increase its TEC.
Test	The LT causes the IUT to transmit a frame. Then the LT causes the IUT to generate an error frame. At the last bit of the error delimiter, the LT sends one dominant bit.
Verification	The TEC shall equal the Set-up value increased by 8.

8.6.17 TEC non-increment on acknowledgement error in passive state

8.6.17.1 Purpose and limits of test case

This test is applicable to CAN_VERSION $\in$ {A, B, BP}

It is used to verify that a passive state IUT acting as a transmitter does not increase its TEC when detecting an acknowledgement error followed by a passive-error flag.

One elementary test shall be performed.

8.6.17.2 Test case organization

Test case organization shall be in accordance with Table 133.

Table 133 — TEC non-increment on acknowledgement error in passive state — Test case organization

State	Description
Set-up	The IUT is set to the TEC passive state.
Test	The LT causes the IUT to transmit a frame. The LT does not acknowledge this frame. After the acknowledgement error the LT sends a passive-error frame.
Verification	The TEC shall equal the set-up value.

8.6.18 TEC increment on acknowledgement error in passive state

8.6.18.1 Purpose and limits of test case

This test is applicable to CAN_VERSION $\in$ {A, B, BP}.

It is used to verify that a passive state IUT acting as a transmitter increases its TEC when detecting an acknowledgement error followed by at least one dominant bit during the passive-error flag.

One elementary test shall be performed.

8.6.18.2 Test case organization

Test case organization shall be in accordance with Table 134.

Table 134 — TEC increment on acknowledgement error in passive state — Test case organization

State	Description
Set-up	The IUT is set to the TEC passive state.
Test	The LT causes the IUT to transmit a frame. The LT does not acknowledge this frame. After the Acknowledgement error the LT sends a dominant bit at the sixth bit position of the passive-error flag.
Verification	The TEC shall equal the set-up value increased by 8.

8.6.19 TEC non-increment on stuff error during arbitration

8.6.19.1 Purpose and limits of test case

This test is applicable to CAN_VERSION ∈ {A, B, BP}.

It is used to verify that an IUT acting as a transmitter does not increase its TEC when detecting a stuff error during arbitration when monitoring a dominant bit.

One elementary test shall be performed.

8.6.19.2 Test case organization

Test case organization shall be in accordance with Table 135.

Table 135 — TEC non-increment on stuff error during arbitration — Test case organization

State	Description
Set-up	No action required. The IUT is left in the default state.
Test	The LT causes the IUT to transmit a frame and forces a recessive stuff bit to a dominant state.
Verification	The TEC shall equal the Set-up value.

8.7 Test class 7, bit timing

8.7.1 Sample point

8.7.1.1 Purpose and limits of test case

This test is applicable to CAN_VERSION ∈ {A, B, BP}.

Its purpose is to verify the sample point of an IUT acting as a transmitter.

One elementary test shall be performed.

8.7.1.2 Test case organization

Test case organization shall be in accordance with Table 136.

Table 136 — Sample point — Test case organization

State	Description
Set-up	No action required. The IUT is left in the default state.
Test	The LT causes the IUT to transmit a frame. The LT shortens a dominant bit preceded by a recessive bit by an amount of (Phase_Seg2 - TSYS) and later shortens another dominant bit preceded by a recessive bit by an amount of (Phase_Seg2 + 1TQ + 1TSYS).
Verification	The IUT shall generate an error frame on the bit position following the second shortened bit.

8.7.2 Hard synchronization on SOF reception before sample point

8.7.2.1 Purpose and limits of test case

This test is applicable to CAN_VERSION $\in$ {A, B, BP}.

Its purpose is to verify that the IUT, with a pending transmission, makes a hard synchronization when detecting a dominant bit before the sample point of the third bit of the intermission field.

One elementary test shall be performed.

8.7.2.2 Test case organization

Test case organization shall be in accordance with Table 137.

Table 137 — Hard synchronization on SOF reception before sample point — Test case organization

State	Description
Set-up	No action required. The IUT is left in the default state.
Test	The LT causes the IUT to transmit a frame. While the IUT's transmission is pending the LT generates a dominant bit starting (1 TQ + 1 TSYS) before the sample point of the third bit of the intermission field.
Verification	The IUT shall start transmitting the first bit of the identifier one bit time after the recessive-to-dominant edge of the SOF. The first edge inside the arbitration field has to be transmitted an integer number of bit time after the SOF edge.

8.7.3 Hard synchronization on SOF Reception after sample point

8.7.3.1 Purpose and limits of test case

This test is applicable to CAN_VERSION $\in$ {A, B, BP}.

Its purpose is to verify that the IUT, with a pending transmission, makes a hard synchronization when detecting a dominant bit after the sample point of the third bit of the intermission field.

One elementary test shall be performed.

8.7.3.2 Test case organization

Test case organization shall be in accordance with Table 138.

Table 138 — Hard synchronization on SOF Reception after sample point — Test case organization

State	Description
Set-up	No action required. The IUT is left in the default state.
Test	The LT causes the IUT to transmit a frame. While the IUT's transmission is pending the LT generates a dominant bit starting IPT after the sample point of the third bit of the intermission field.
Verification	The IUT shall start transmitting its SOF at the next TQ following the dominant-to-recessive edge.

8.7.4 Synchronization when $e < 0$ and $|e| \leqslant$ SJW

8.7.4.1 Purpose and limits of test case

This test is applicable to CAN_VERSION $\in$ {A, B, BP}.

Its purpose is to verify the behaviour of an IUT, acting as a transmitter, detecting a negative phase error, e, on a recessive-to-dominant edge with $|e| \leqslant$ SJW.

The value tested for e are in time quantum with $|e| \in$ {1, min [SJW, (Phase_Seg2 − IPT)]}.

At least [SJW, (Phase_Seg2 − IPT)] elementary tests shall be performed.

8.7.4.2 Test case organization

Test case organization shall be in accordance with Table 139.

Table 139 — Synchronization when $e < 0$ and $|e| \leqslant$ SJW — Test case organization

State	Description
Set-up	No action required. The IUT is left in the default state.
Test	The LT causes the IUT to transmit a frame. The LT shortens a recessive bit preceding a dominant bit by an amount of $\|e\|$ inside the arbitration field.
Verification	The next edge sent by the IUT occurs an integer number of bit times after the edge applied by the LT.

8.7.5 Synchronization for $e < 0$ and $|e| >$ SJW

8.7.5.1 Purpose and limits of test case

This test is applicable to CAN_VERSION $\in$ {A, B, BP}.

Its purpose is to verify the behaviour of an IUT acting as a transmitter detecting a negative phase error, e, on a recessive to dominant bit with $|e| >$ SJW.

The value tested for e are in TQ $|e| \in$ [(SJW + 1), (Phase_Seg2 − IPT)].

There are [(Phase_Seg2 − IPT) − SJW] elementary tests to perform.

8.7.5.2 Test case organization

Test case organization shall be in accordance with Table 140.

Table 140 — Synchronization for e < 0 and |e| > SJW — Test case organization

State	Description
Set-up	No action required. The IUT is left in the default state.
Test	The LT causes the IUT to transmit a frame. The LT shortens a recessive bit preceding a dominant bit by an amount of |e| inside the arbitration field.
Verification	The next edge sent by the IUT occurs an integer number of bit times plus an amount (|e| − SJW) after the edge applied by the LT.

8.7.6 Glitch filtering on negative phase error

8.7.6.1 Purpose and limits of test case

This test is applicable to CAN_VERSION ∈ {A, B, BP}.

Its purpose is to verify that there is only one synchronization within one bit time if there are two dominant to recessive edges between two sample points where the first edge comes before the synchronization segment.

The test is also used to verify that an IUT is able to synchronize on a minimum duration pulse obeying to the synchronization rules.

One elementary test shall be performed.

8.7.6.2 Test case organization

Test case organization shall be in accordance with Table 141.

Table 141 — Glitch filtering on negative phase error — Test case organization

State	Description
Set-up	No action required, the IUT is left in the default state.
Test	The LT causes the IUT to transmit a frame. A recessive bit which is followed by a dominant bit is shortened by 1 TQ. After 1 TQ of dominant value the LT forces 1 TQ to recessive value.
Verification	The IUT shall send a dominant-to-recessive edge an integer number of bit time after the first recessive-to-dominant edge applied by the LT.

8.7.7 Non-synchronization on dominant bit transmission

8.7.7.1 Purpose and limits of test case

This test is applicable to CAN_VERSION ∈ {A, B, BP}.

Its purpose is to verify that an IUT transmitting a dominant bit does not perform any re-synchronization as a result of a recessive-to-dominant edge with a positive phase error.

One elementary test shall be performed.

8.7.7.2 Test case organization

Test case organization shall be in accordance with Table 142.

Table 142 — Non-synchronization on dominant bit transmission — Test case organization

State	Description
Set-up	No action required. The IUT is left in the default state.
Test	The LT causes the IUT to transmit a frame. While the IUT is transmitting the frame, the LT delays each recessive to dominant edge by 2 TQ.
Verification	The IUT shall continue transmitting frame without any re-synchronization.

8.7.8 Synchronization before information processing time

8.7.8.1 Purpose and limits of test case

This test is applicable to CAN_VERSION $\in$ {A, B, BP}.

Its purpose is to verify that an IUT transmitting will synchronise correctly in case of a re-synchronization as a result of a recessive-to-dominant edge that occurs immediately after the sample point.

One elementary test shall be performed.

8.7.8.2 Test case organization

Test case organization shall be in accordance with Table 143.

Table 143 — Synchronization before information processing time — Test case organization

State	Description
Set-up	No action required. The IUT is left in the default state.
Test	The LT causes the IUT to transmit a frame that contains a sequence of 4 alternating recessive and dominant bits. While the IUT is transmitting the frame, the LT shortens the first recessive bit of the alternating sequence by an amount of Phase_Seg 2 and sends a dominant bit.
Verification	The next edge from recessive to dominant sent by the IUT shall occur two CAN bit times + (Phase_Seg 2 – SJW) after the edge applied by the LT and the IUT shall continue transmitting the frame.

8.7.9 Synchronization after sample point while sending dominant bit

8.7.9.1 Purpose and limits of test case

This test is applicable to CAN_VERSION $\in$ { B, BP}.

Its purpose is to verify the synchronization of a IUT, acting as a transmitter, on a recessive to dominant edge after the sample point, while sending a dominant bit. The edge is caused by a disturbance of the dominant bit.

8.7.9.2 Test case organization

Test case organization shall be in accordance with Table 144.

Table 144 — Synchronization after sample point while sending dominant bit — Test case organization

State	Description
Set-up	No action required. The IUT is left in the default state.
Test	The LT causes the IUT to transmit a frame with a dominant stuff bit in the identifier. While the IUT sends the dominant stuff bit, the LT forces the IUT RX signal 2 TQ to recessive state starting 1 TQ in advance of the sample point.
Verification	The IUT has to send an error frame as the reaction to the falsified dominant stuff bit. The dominant to recessive edge of the error delimiter sent by the IUT has to occur 6 bit times + (Phase_Seg2 – SJW) after the recessive-to-dominant edge applied by the LT after the sample point of the dominant stuff bit.

9 Test type 3, bi-directional frame

9.1 Test class 1, valid frame format

No test is specified.

9.2 Test class 2, error detection

No test is specified.

9.3 Test class 3, active error frame management

No test is specified.

9.4 Test class 4, overload frame management

No test is specified.

9.5 Test class 5, passive-error state and bus-off

No test is specified.

9.6 Test class 6, error counter management

9.6.1 REC unaffected when increasing TEC

9.6.1.1 Purpose and limits of test case

This test is applicable to CAN_VERSION $\in$ {A, B, BP}.

It is used to verify that increasing REC and TEC are independent operations.

One elementary test shall be performed.

9.6.1.2 Test case organization

Test case organization shall be in accordance with Table 145.

Table 145 — REC unaffected when increasing TEC — Test case organization

State	Description
Set-up	No action required, the IUT is left in the default state.
Test	The LT causes the IUT to increase its REC up to 127. Then LT causes the IUT to increase its TEC up to 128. Then the LT sends a frame containing a stuff error.
Verification	Each increment of the TEC shall be responded by an active-error flag. The IUT responds to the stuff error with a passive-error flag.

9.6.2 TEC unaffected when increasing REC

9.6.2.1 Purpose and limits of test case

This test is applicable to CAN_VERSION $\in$ {A, B, BP}.

It is used to verify that increasing REC and TEC are independent operations.

One elementary test shall be performed.

9.6.2.2 Test case organization

Test case organization shall be in accordance with Table 146.

Table 146 — TEC unaffected when increasing REC — Test case organization

State	Description
Set-up	No action required, the IUT is left in the default state.
Test	The LT causes the IUT to increase its TEC up to 127. Then LT causes the IUT to increase its REC up to 128. Then the LT causes the IUT to send a frame and corrupts this frame.
Verification	Each increment of the REC shall be responded by an active-error flag. The IUT responds to the corrupted bit with a passive-error flag.

9.7 Test class 7, bit timing

No test case is specified.

ICS 43.040.15

Price based on 77 pages