ETSI TS 103 713 V18.0.0 (2023-07)

Smart Secure Platform (SSP);
SPI interface
(Release 18)

Reference
RTS/SET-T103713vi00

Keywords
M2M, MFF

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - APE 7112B
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° w061004871

Contents

Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to normative deliverables may have been declared to ETSI. The declarations pertaining to these essential IPRs, if any, are publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (https://ipr.etsi.org/).

Pursuant to the ETSI Directives including the ETSI IPR Policy, no investigation regarding the essentiality of IPRs, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

DECT™, **PLUGTESTS™**, **UMTS™** and the ETSI logo are trademarks of ETSI registered for the benefit of its Members. **3GPP™** and **LTE™** are trademarks of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners. **oneM2M™** logo is a trademark of ETSI registered for the benefit of its Members and of the oneM2M Partners. **GSM**® and the GSM logo are trademarks registered and owned by the GSM Association.

Foreword

This Technical Specification (TS) has been produced by ETSI Technical Committee Secure Element Technologies (SET).

The contents of the present document are subject to continuing work within TC SET and may change following formal TC SET approval. If TC SET modifies the contents of the present document, it will then be republished by ETSI with an identifying change of release date and an increase in version number as follows:

Version x.y.z

where:

- x the first digit:
  - 0 early working draft;
  - 1 presented to TC SET for information;
  - 2 presented to TC SET for approval;
  - 3 or greater indicates TC SET approved document under change control.
- y the second digit is incremented for all changes of substance, i.e. technical enhancements, corrections, updates, etc.
- z the third digit is incremented when editorial only changes have been incorporated in the document.

Modal verbs terminology

In the present document "**shall**", "**shall not**", "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the ETSI Drafting Rules (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

1 Scope

The present document describes the SPI interface for the communication of an SSP, as defined in ETSI TS 103 666-1 [1] using the SCL protocol.

2 References

2.1 Normative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

- In the case of a reference to a TC SET document, a non-specific reference implicitly refers to the latest version of that document in the same Release as the present document.

Referenced documents which are not found to be publicly available in the expected location might be found at https://docbox.etsi.org/Reference/.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are necessary for the application of the present document.

[1] ETSI TS 103 666-1: "Smart Secure Platform (SSP); Part 1: General characteristics".

[2] ETSI TS 102 613: "Smart Cards; UICC - Contactless Front-end (CLF) Interface; Physical and data link layer characteristics".

[3] ISO/IEC 13239: "Information Technology -- Telecommunications and information exchange between systems -- High-level Data Link Control (HDLC) procedures".

2.2 Informative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

- In the case of a reference to a TC SET document, a non-specific reference implicitly refers to the latest version of that document in the same Release as the present document.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

[i.1] ETSI TR 102 216: "Smart cards; Vocabulary for Smart Card Platform specifications".

3 Definition of terms, symbols and abbreviations

3.1 Terms

For the purposes of the present document, the terms given in ETSI TR 102 216 [i.1] and the following apply:

data transfer: information exchange during an SPI access between the master and the slave with SPI_MISO driven by the slave and SPI_MOSI driven by the master while the master is toggling the SPI_CLK signal

flow control: mechanism of the Data Link Layer that consists of methods applied by the transmitter in order to send at any time a number of logical data units that can be accepted by the receiver

frame: link layer data structure consisting of a prologue or frame header, payload and epilogue or trailer usually containing the CRC bytes

MAC access request: request from the slave to the master for a data transfer, i.e. a MAC phase initiated by the slave

MAC phase: initiation of a data transfer by the master and/or request for a data transfer by the slave

SPI access: SPI_NSS assertion by the master, if not already asserted in the MAC phase, followed by SPI_CLK start for transferring a certain number of bytes according to the SPI master configuration

NOTE: The number of bytes transferred during an SPI access is always the same in both directions on SPI_MISO and SPI_MOSI and is also referred to as access length.

window size: maximum number of logical data units that can be sent from the transmitter to the receiver without any link layer acknowledgements for any of these data units

window size slot: fixed space used by the slave in the receive buffer for the logical data units

NOTE: The length of a window size slot equals the Data Link Layer MTU.

3.2 Symbols

Void.

3.3 Abbreviations

For the purposes of the present document, the following abbreviations apply:

A	Asserted
AC	Alternating Current
ACT	ACTivation
CLF	ContactLess Frontend
CLT	ContactLess Tunnelling
CMD	Command
CPHA	Clock Phase
CPOL	Clock Polarity
CRC	Cyclic Redundancy Check
D	Driven (either Low Level or High Level)
DA	De-Asserted
DC	Direct Current
HiZ	High Impedance
II	Input Ignored
IL	Input Listened
IO	Input/Output
IOH	High Output Current (Output current corresponding to VOH)
IOL	Low Output Current (Output current corresponding to VOL)
ISO	International Organization for Standardization
LLC	Logical Link Control

LPDU	Link Protocol Data Unit
MAC	Medium Access Control
MAX	Maximum
MCT	MAC aCTivation
MIN	Minimum
MISO	Master Input Slave Output
MOSI	Master Output Slave Input
MSB	Most Significant Bit
MTU	Maximum Transmission Unit
NSD	Non-Significant Data
OD	Open Drain
OSI	Open System Interconnection
POT	Power-On Time
RFU	Reserved for Future Use
SCL	SSP Common Layer
SHDLC	Simplified High Level Data Link Control
SPI	Serial Peripheral Interface
SS_MI	Slave Select Master Input signal
SS_MO	Slave Select Master Output signal
SS_SI	Slave Select Slave Input signal
SS_SO	Slave Select Slave Output signal
SSP	Smart Secure Platform
SWP	Single Wire Protocol

NOTE: As defined in ETSI TS 102 613 [2].

UICC	Universal Integrated Circuit Card
VDD	Supply Voltage
VIH	High Input Voltage (Input Voltage for High Logic Level)
VIL	Low Input Voltage (Input Voltage for Low Logic Level)
VOH	High Output Voltage (Output Voltage for High Logic Level)
VOL	Low Output Voltage (Output Voltage for Low Logic Level)

4 Introduction

The Serial Peripheral Interface (SPI) is a serial synchronous full-duplex communication interface between a single master and one or more slaves present on the same SPI bus, each slave being selected at one time by a dedicated SPI_NSS signal. This clause defines the physical, MAC and data link layers for the SPI interface.

In this clause the terms master and slave refer respectively to the terms master SPI and slave SPI.

5 SCL Under-Layers Protocol Stack

Figure 5.1 illustrates the protocol stack below the SCL supporting the SPI interface.

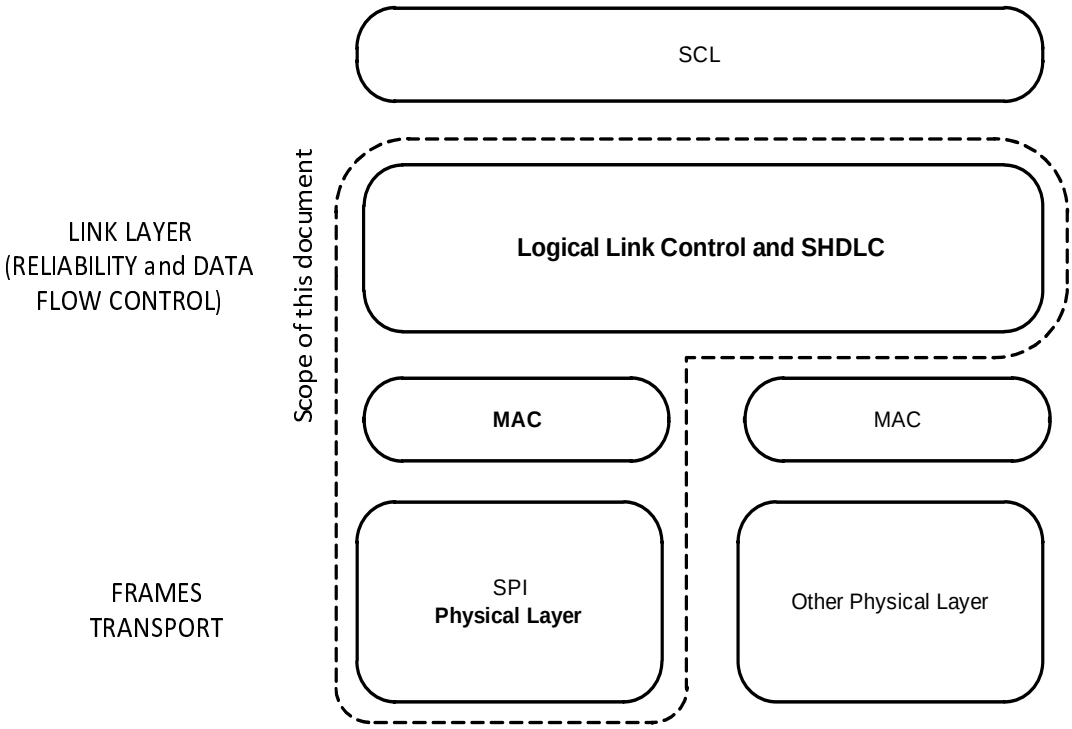


Figure 5.1: Protocol stack for SPI Interface

6 Electrical interfaces

6.1 Introduction

In the clauses below, different implementations of SPI interface are defined. These implementations allow bi-directional communication and the possibility for the slave to initiate communication with the master when it has data available thus avoiding the necessity for continuous polling to be performed by master.

Slave may initiate communication to send a command without a prior command from master.

6.2 Physical interface with 5 signals

Figure 6.1 illustrates the SPI electrical interface using 5 signals.

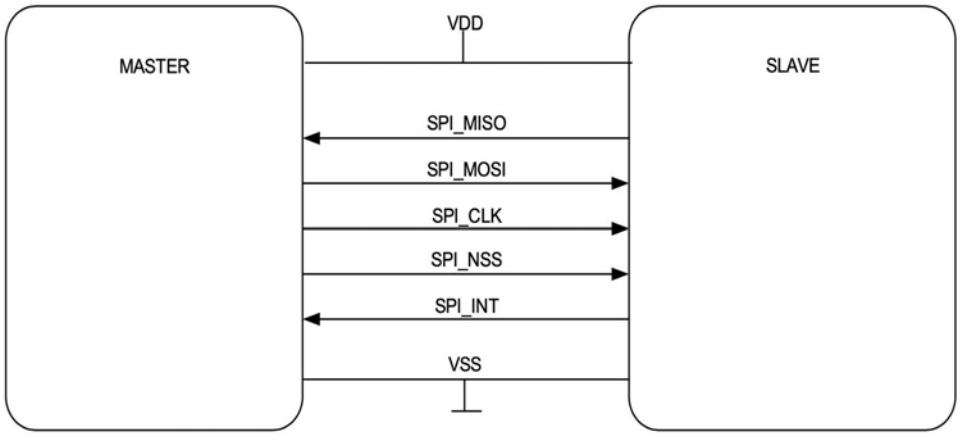


Figure 6.1: SPI electrical interface with 5 signals

This SPI interface describes two sets of signals:

- The generic and legacy SPI interface using the 4 signals:
  - SPI_MOSI (Master Output Slave Input);
  - SPI_MISO (Master Input Slave Output);
  - SPI_CLK (clock);
  - SPI_NSS signal used for the selection of a Slave Endpoint among N slaves sharing the same bus;

  SPI_MISO, SPI_MOSI and SPI_CLK can be shared between several SPI slaves present on the same SPI bus.
- The SPI_INT signal allows the slave to initiate a MAC access request in order to notify the master to start a data transfer.

SPI_INT is defined as an edge-triggered interrupt. It is asserted on the rising edge of the signal.

SPI_NSS is considered active or asserted at low voltage level.

6.3 Physical interface with 4 signals

Figure 6.2 illustrates the SPI interface using 4 signals, bi-directional SPI_NSS.

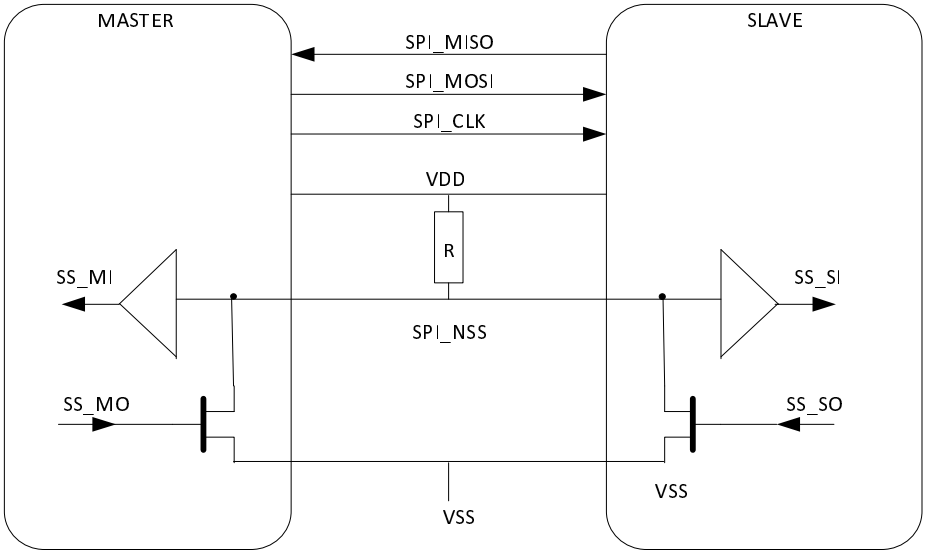


Figure 6.2: SPI electrical interface with 4 signals, bi-directional SPI_NSS

The SPI interface with 4 signals describes two sets of signals:

- The three generic and legacy SPI signals as SPI_MOSI (Master Output Slave Input), SPI_MISO (Master Input Slave Output) and SPI_CLK (clock). These signals can be shared between several SPI slaves as a bus.
- The SPI_NSS (Negative Slave Select) signal used for the selection of a slave endpoint among N slaves sharing the same bus and for the slave to initiate a MAC access request to notify the master to initiate a data transfer.

SPI_NSS is considered active or asserted at low voltage level. SPI_NSS requires a bidirectional IO implementing an Open Drain (OD) interface for both master and slave. This configuration allows driving the SPI_NSS signal to low voltage level by both master and slave without electrical conflict.

A pull-up resistor keeps SPI_NSS at high state level (i.e. idle state) when SS_MO and SS_SO are not asserted. The SPI_NSS signal is at low state when either SS_MO or SS_SO are asserted.

NOTE: The current industry de-facto SPI specification defines SPI_NSS signal as unidirectional, driven by the master. However, in the present document the SPI_NSS in the 4 signals configuration is bidirectional.

Table 6.1: Definition of the signals

Signal	Description
SS_MO	Internal master output signal for SPI_NSS assertion. SS_MO is at high state level for generating a SPI_NSS signal assertion (i.e. low level state)
SS_SO	Internal slave output signal for SPI_NSS assertion. SS_SO is at high state level for generating a SPI_NSS signal assertion (i.e. low level state)
SS_MI	Internal master input signal indicating SPI_NSS status. SS_MI is at high state level when the SPI_NSS signal is not asserted
SS_SI	Internal slave input signal indicating SPI_NSS status. SS_SI is at high state level when the SPI_NSS signal is not asserted
SPI_NSS	SPI_NSS signal: low state level when asserted

6.4 Electrical characteristics

6.4.1 DC characteristics

The SPI Electrical specification interface is defined for VDD operational voltage classes B, C and D as defined in ETSI TS 103 666-1 [1], clause 6.2.2.3. An implementation shall support at least one of these voltage classes.

NOTE: The negotiation of the voltage class between the master and the slave is not defined in the present document.

Table 6.2: DC characteristics for operational voltage class B

Parameter	Symbol	Min	Max	Unit	Note/Test condition
Input high voltage	VIH	0,7 × VDD	VDD + 0,3	V	
Input low voltage	VIL	-0,3	0,3 × VDD	V	
Output high voltage	VOH	0,9 × VDD	VDD (see note 2)	V	IOH = -100 uA
Output low voltage	VOL	0 (see note 2)	0,1 × VDD	V	IOL = 1,0 mA
SPI_NSS Low Level Output current (see note 1)	IOL	-1	-	mA	VOL = 0,3 V
Maximal SPI_NSS line capacitance (see note 1)	Cl	-	20	pF	
NOTE 1: Applicable for the physical interface with 4 signals. NOTE 2: To allow for overshoot, the voltage on I/O shall remain between -0,3 V and VDD + 0,3 V during dynamic operation.					

Table 6.3: DC characteristics for operational voltage class C

Parameter	Symbol	Min	Max	Unit	Note/Test condition
Input high voltage	VIH	0,7 × VDD	VDD + 0,3	V	
Input low voltage	VIL	-0,3	0,3 × VDD	V	
Output high voltage	VOH	0,9 × VDD	VDD (see note 2)	V	IOH = -100 uA
Output low voltage	VOL	0 (see note 2)	0,1 × VDD	V	IOL = 1,0 mA
SPI_NSS Low Level Output current (see note 1)	IOL	-1	-	mA	VOL = 0,3 V
Maximal SPI_NSS line capacitance (see note 1)	Cl	-	20	pF	
NOTE 1: Applicable for the physical interface with 4 signals. NOTE 2: To allow for overshoot, the voltage on I/O shall remain between -0,3 V and VDD + 0,3 V during dynamic operation.					

Table 6.3a: DC characteristics for operational voltage class D

Parameter	Symbol	Min	Max	Unit	Note/Test condition
Input high voltage	VIH	0,7 × VDD	VDD + 0,3	V	
Input low voltage	VIL	-0,3	0,3 × VDD	V	
Output high voltage	VOH	0,9 × VDD	VDD (see note 2)	V	IOH = -100 uA
Output low voltage	VOL	0 (see note 2)	0,1 × VDD	V	IOL = 0,6 mA
SPI_NSS Low Level Output current (see note 1)	IOL	-0,6	-	mA	VOL = 0,12 V
Maximal SPI_NSS line capacitance (see note 1)	Cl	-	20	pF	
NOTE 1: Applicable for the physical interface with 4 signals. NOTE 2: To allow for overshoot, the voltage on I/O shall remain between -0,3 V and VDD + 0,3 V during dynamic operation.					

The value of the resistor R in figure 6.2 shall be selected for a resultant maximum current lower than or equal to the minimum between the absolute IOL values of the master and the slave.

6.4.2 Data transfer mode, AC characteristics

The SPI interface shall implement the SPI mode 0 according to the industry de-facto SPI specification.

SPI mode 0 is determined by CPOL = 0 and CPHA = 0 where:

- CPOL: defines the SPI_CLK idle state.
- CPOL = 0 implies that the SPI_CLK is at input low voltage while it is idle.
- CPHA: defines the data sampling time.
- CPHA = 0 implies that data sampling is done on the rising edges of the SPI_CLK for both SPI_MISO and SPI_MOSI.

SPI_NSS is considered active or asserted at low voltage level.

Data availability timings with reference to SPI_CLK are shown in figure 6.3.

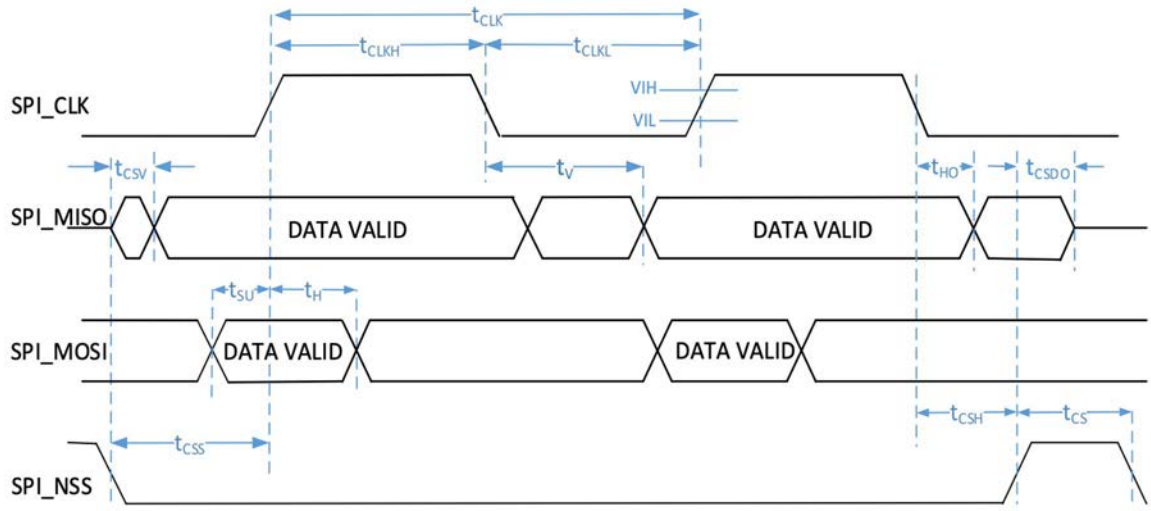


Figure 6.3: SPI timing diagram

Table 6.4: AC characteristics for voltage classes B, C and D (SPI Slave, Mode 0: CPOL = 0, CPHA = 0)

Symbol	Definition	Value (MIN values unless MAX specified)
fCLK	SPI_CLK frequency	Max SPI_CLK is specified by Slave at initialization in MCT_READY
tCLKL	SPI_CLK low time	0,45 × tCLK
tCLKH	SPI_CLK high time	0,45 × tCLK
tSU	Data setup time to clock rising edge	5 ns
tH	SPI_MOSI hold time/Data hold time to clock rising edge	3 ns
tHO	SPI_MISO hold time/Output hold time to clock falling edge	0 ns
tCSS	SPI_NSS setup time (class C, D)	63 ns
tCSS	SPI_NSS setup time (class B)	33 ns
tCSH	Hold time clock falling edge to SPI_NSS inactive	0,5 × tCLK
tCS	SPI_NSS inactive time (class C, D)	60 ns
tCS	SPI_NSS inactive time (class B)	30 ns
tCSV	SPI_MISO valid delay time from SPI_NSS active (class C, D)	58 ns (MAX)
tCSV	SPI_MISO valid delay time from SPI_NSS active (class B)	28 ns (MAX)
tV	SPI_MISO valid delay time from clock falling edge	0 ns (MIN) 0,7 × tCLKL (MAX)
tCSDO	SPI_MISO Output disable time from SPI_NSS inactive (class C, D)	0 ns (MIN) 60 ns (MAX)
tCSDO	SPI_MISO Output disable time from SPI_NSS inactive (class B)	0 ns (MIN) 30 ns (MAX)

The values indicated in table 6.4 are reference values for generic SPI slaves. If the concrete slave device supports better timing parameters, a system design may choose to configure the master for these timing values in order to achieve better performance.

6.5 Slave state

6.5.1 Slave state definitions

The slave shall be in one of the following states:

Initial state:

The slave enters this state as soon as it is powered on and VDD is valid or after a reset. In this state, the slave is not initialized and the SPI module is not ready to send or receive any data. The master shall not perform any SPI access while the slave is in this state.

After the POT time, following VDD valid or a reset, the slave shall transition to the state configured/de-selected.

Configured state:

The slave in configured state has two sub states:

- **De-selected sub-state:** In this state, the slave SPI module is enabled, i.e. it shall not ignore SPI_NSS assertion by master and shall be ready for an SPI access. SPI_NSS is not asserted by the master. As a consequence, the slave shall have SPI_MISO in high impedance and shall ignore both SPI_CLK and SPI_MOSI. In an implementation with 4 signal interface, SPI_NSS is not asserted by the slave.

  In this state, the slave is waiting for a master access. The slave will also be in this state after issuing a MAC access request, until the master generates an access for the data transfer.

  From this state, the slave shall enter into one of the states: configured/selected, pro-active or power saving mode; or initial in case of a reset.

- **Selected sub-state:** In this state, the master has asserted the SPI_NSS during the MAC phase and data transfer phase.

  The slave SPI module is enabled, i.e. it shall not ignore SPI_NSS assertion by the master and shall not ignore SPI_CLK and SPI_MOSI. SPI_MISO is not in high impedance.

  From this state, the slave shall enter into one of the states: configured/de-selected or pro-active; or initial in case of a reset.

Pro-active state:

This state is entered by the slave when data need to be sent. The slave issues a MAC access request by asserting the SPI_NSS or SPI_INT. The slave enters this state for the duration of T2, as described in clauses 7.2.3.2 or 7.2.4.3. The slave exits this state on its own after T2, regardless of the input signals states driven by the master:

- For the 4 signal interface:

  The slave shall have its SPI module disabled, i.e. all input signals shall be ignored and SPI_MISO shall be in high impedance. The slave asserts SPI_NSS.

  It is implementation-dependent if the slave SPI module keeps its internal configuration during T2 duration or requires a (partial) re-configuration when exiting this state, which has to be performed within T1 Slave Ready Time reported to the master.

- For the 5 signal interface:

  The slave SPI module needs not to be disabled.

  It is implementation-dependent if the slave SPI module keeps its internal configuration during T2 or needs a (partial) re-configuration when exiting the pro-active state, which has to be performed within T1 Slave Ready Time.

For both the 4 signal and 5 signal implementations, the slave shall transition from the pro-active state to:

- the state configured/de-selected if only the slave generated a MAC access request and the master did not initiate any MAC phase simultaneously;
- the state configured/selected when both the master and the slave generate a MAC phase simultaneously;
- the initial state in case of a reset.

Busy state:

This is an optional state applicable only for the case of the 4 wire interface. It is implementation-dependent if the slave SPI module keeps its internal configuration and/or stays enable during this state.

This state is entered by the slave when the slave performs the optional slave driven flow control by keeping SPI_NSS asserted following the configured/selected state, as described in clause 7.2.4.5.

From this state the slave shall enter the configured/de-selected state.

Power saving mode state:

This state is entered by the slave to reduce the power consumption. Entry and exit conditions are described in clause 7.8.

The slave shall transition from this state to the state configured/selected after the master has resumed the slave; or to the initial state in case of a reset.

6.5.2 Slave state diagram

Figure 6.4 shows the diagram of the slave states.

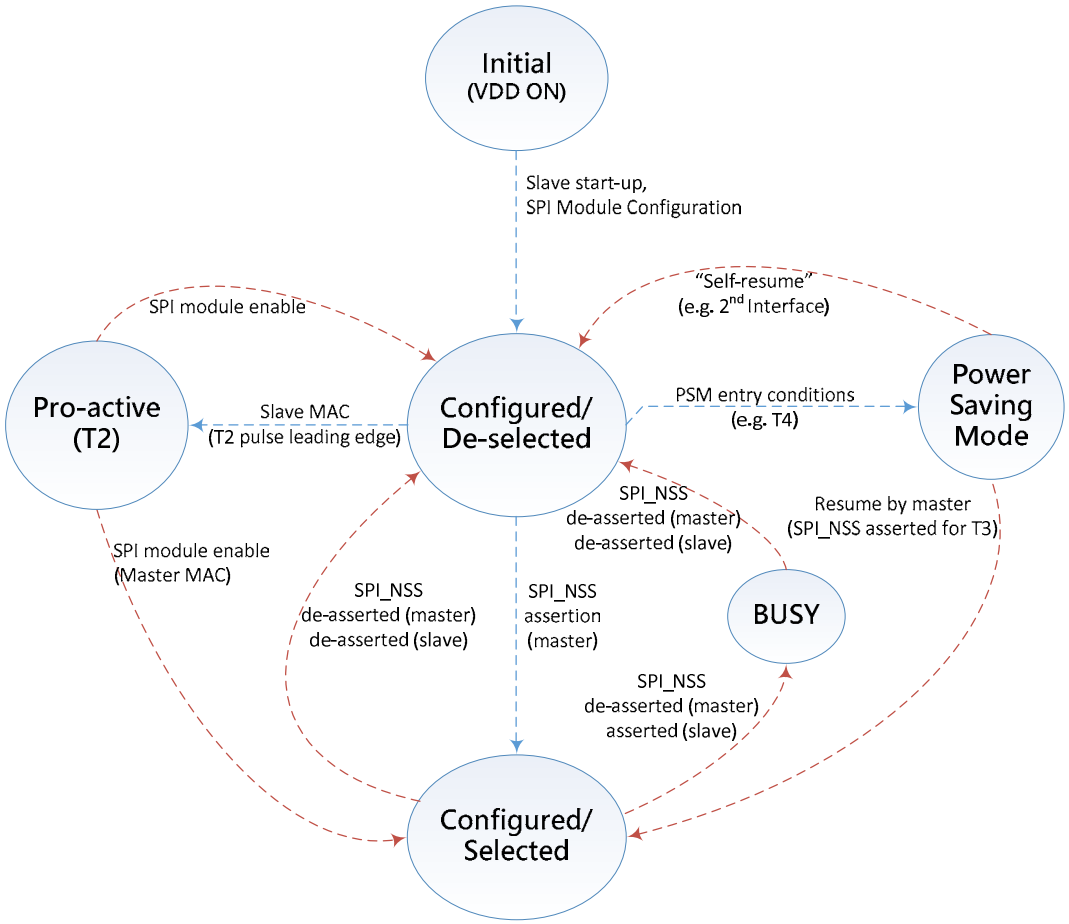


Figure 6.4: Diagram of the slave states

NOTE 1: For the sake of readability, the transitions to the initial state if a reset occurs are not shown.

NOTE 2: Additional information on slave state is provided in annex A.

7 Data Link Layer

7.1 Overview

Clause 9.1 in ETSI TS 102 613 [2] shall apply.

7.2 MAC Layer

7.2.1 Overview

The MAC phase is the initial handshake phase between the SPI master and the SPI slave followed by the SPI data transfer phase.

7.2.2 Timing

7.2.2.1 Timing definitions

Table 7.1 describes the timing parameters of the MAC layer. In addition to these MAC timings, the timing requirements listed in AC electrical characteristics shall apply for all MAC diagrams in the following clauses.

Table 7.1: MAC timing parameters

Symbol	Definition	Value (min)/ Reference	Description
T1	Slave Ready Time	Reported in MCT_READY	MAC phase time prior to data transfer start. This time is the maximum time needed for the SPI slave to be ready for the data transfer and is reported in MCT_READY.
T2	Slave Request Time	1 µs	SPI_NSS or SPI_INT assertion min pulse width.
T3	Slave Resume Time from power saving mode	Reported in MCT_READY	Time from SPI_NSS assertion by master for slave resume to data transfer start. T3 value shall be used by master in MAC phase instead of T1 when the slave is in power saving mode.
T4 (see note)	Inactivity time	Negotiated using the MCT_MASTER_REQ and MCT_READY	Inactivity period for entry into power saving mode (ms) as defined in clause 7.8.2.1.
T5	Master Ready Time	Reported in MCT_MASTER_REQ	The SPI master MAC phase time needed after T1 prior to the data transfer start. This additional delay time after T1 is needed by the SPI master to be ready for a data transfer and is reported in MCT_MASTER_REQ.
T6	Master Resume Time from power saving mode	Reported in MCT_MASTER_REQ	The SPI master time delay prior to the data transfer start, when the SPI master is resuming from power saving mode. This time period is needed by the SPI master to be ready for a data transfer and is reported in MCT_MASTER_REQ.
T7	Maximum delay time	Reported in MCT_READY	Maximum delay time requested by the slave after T1 until the SPI master starts the data transfer.
T8	Master MAC access request ready time	Reported in MCT_MASTER_REQ	Time required by the master from the point it de-asserts the SPI_NSS after completion of a link layer frame transfer to the time it accepts a slave MAC access request.
NOTE: T4 Inactivity time is not a MAC Timing. It is defined in clause 7.8.2.1.			

7.2.2.2 T1 = Slave Ready Time

T1 is the MAC phase time required by the slave to get configured and enabled at the end of the MAC phase, ready for the data transfer phase start (i.e. when the SPI_CLK can be started by master).

T1 is defined as the maximum time the slave needs to become ready for a data transfer. The slave requires the SPI master to allow at least a time T1 from the leading edge of the SPI_NSS or SPI_INT assertion by either master or slave to the data transfer phase start.

T1 becomes a minimum time value requirement from the master perspective. The SPI master starts the data transfer phase at the time Ts after the end of T1 by asserting SPI_NSS, if not already asserted depending on the prior MAC phase, followed by the SPI_CLK start.

T1 is slave implementation dependant.

T1 is determined by the slave and includes T2 and any slave-specific internal latencies. Slave shall provide a T1 value that covers the worst–case time it needs between the moment of sampling SPI_NSS to the time when slave becomes ready for the data transfer.

Master shall allow at least the time T1 requested by the slave between the start of the MAC phase initiated by either master or slave and the point in time when the data transfer phase starts. Master may use a Ts delay in addition to T1. The delay Ts shall be lower than T7 if T7 is provided. Ts may vary from one MAC phase to another.

7.2.2.3 T2 = Slave Request Time

T2 is the duration of an SPI_NSS or SPI_INT pulse generated by the slave for a MAC access request.

The minimum value of T2 is 1 µs.

In order to sense the interrupts originating either from slave SPI_NSS or SPI_INT assertion, the master shall be configured for edge-triggered interrupts.

NOTE: The leading edges of the MAC access request signals of the slave should assert the internal interrupt of the master.

7.2.2.4 T3 = Slave resume time from power saving mode

T3 is the slave resume time from the power saving mode. It is slave implementation dependant. Slave reports at initialization in MCT_READY the minimum value of T3 required for the slave to become ready for the SPI access. Whenever master needs to resume the slave from power saving mode it should use at least the time T3 during the MAC phase instead of the time T1.

7.2.2.5 T5 = Master Ready Time

T5 is the MAC phase time after T1, required by the SPI master to get configured and enabled at the end of the MAC phase, ready to start the data transfer phase (i.e. when the SPI_CLK can be started by the SPI master). T5 is the minimum value possible for Ts that the SPI master can enforce.

T5 is determined by the SPI master and includes any SPI master-specific internal latencies. When the SPI master is not able to enforce a value for SPI master ready time (i.e. T5), the SPI master shall send 'FFFFFF' as T5 in MCT_DATA. T5 with a value different to 'FFFFFF' indicates the worst–case time needed by the SPI master starting after the end of T1 to the time when the SPI master becomes ready for the data transfer and starts SPI_CLK.

T5 shall not include the SPI master resume time when resuming from power saving mode. If T5 was provided by the SPI master and T7 not provided by the slave or the slave sent 'FFFFFF' for T7, the SPI master should start a SPI access no later than T1+T5.

NOTE: To allow a slave to acknowledge an SHDLC frame in due time, it is recommended for the SPI master and for the SPI slave supporting SHDLC-based flow control to have T1+T5 set to a time shorter than the T1 defined in clause 10.6.1 of ETSI TS 102 613 [2] (e.g. 2,5 ms for a windows size of 2).

7.2.2.6 T6 = Master Resume Time from power saving mode

T6 is the SPI master resume time from a power saving mode, required by the SPI master to get configured and enabled at the end of the MAC phase, ready to start the data transfer phase (i.e. when the SPI_CLK can be started by the SPI master).

T6 is defined from the leading edge of the SPI_NSS or SPI_INT assertion by the slave to the data transfer phase start. T6 is SPI master implementation dependant.

T6 is determined by the SPI master and includes T2 and any SPI master-specific internal latencies. When the SPI master is not able to enforce a value for SPI master resume time from power saving mode (i.e. T6), the SPI master shall send 'FFFFFF' as T6 in MCT_DATA. T6 with a value different to 'FFFFFF' indicates the worst–case time needed by the SPI master from the start of the resume to the time when the SPI master is fully resumed from power saving mode, becomes ready for the data transfer and starts SPI_CLK.

7.2.2.7 T7 = Maximum delay time

T7 is the maximum duration requested by the slave for the SPI master to start the data transfer phase after the end of T1. T7 is the maximum Ts value requested by the SPI slave.

The SPI master shall initiate a data transfer before the T1+T7, if the slave sends a T7 value different from 'FFFFFF'.

The slave shall request a time period T7 at least equal to the length of T5, or reply with the value 'FFFFFF' for the T7 parameter in MCT_DATA. The value 'FFFFFF' for T7 indicates that the slave does not request a maximum delay time. If the SPI master does not receive any T7 value from the slave (i.e. slave with a lower SPI interface version) or if the slave sent T7 = 'FFFFFF', the SPI master should start the SPI access not later than T1+T5.

The slave shall not request a time T7 if either the slave did not receive a T5 time value from the SPI master, i.e. the SPI master has sent 'FFFFFF' for T5 or if the SPI master did not sent any T5 value (e.g. SPI master with a lower SPI interface version).

NOTE: If T6 > T1+T7, a slave that requests a time T1+T7 lower than a valid T6 or that requests a time T1+T7 when T6 is not supported by the SPI master (i.e. the SPI master sent T6 with 'FFFFFF') will prevent the SPI master to enter power saving mode. In such cases, it is recommended for the slave to inform the SPI master as soon as it does not require or expect any further activities as described in clause 7.8.2.1, to allow the SPI master to enter power saving mode.

7.2.2.8 T8 = Master MAC access request ready time

T8 is the maximum time required by the master from the point it de-asserts the SPI_NSS after completion of a link layer frame transfer to the time it accepts a slave MAC access request.

T8 is reported by the master within the MCT_DATA, as shown in table 7.5. The master should use the minimum possible value for T8 for optimal performance.

Before initiating the MAC access request, the slave shall wait for a time ≥ T8 from the completion of the prior SPI access i.e. de-assertion of the SPI_NSS by the master or by the slave in case of slave-driven flow control.

If the master needs this time before a MAC access request (e.g. to reconfigure the SPI_NSS as input for 4 signals interface or to enable the SPI_INT interrupt for 5 signals interface), the master provides T8. If T8 is provided (i.e. T8 > 0) the slave shall wait at least for that T8 time before issuing the MAC access request. For the 5 signals interface, T8 is illustrated in figure 7.2 in clause 7.2.3.2, for the 4 signals interface, T8 is illustrated in figure 7.5 in clause 7.2.4.3.

7.2.3 5 signals MAC layer

7.2.3.1 Initiation of the data transfer from the master

In this case at the start of a MAC phase master asserts the SPI_NSS. Slave may also assert the SPI_INT for making a MAC access request as described in clause 7.2.3.3.

Figure 7.1 illustrates the initiation of the data transfer by the master.

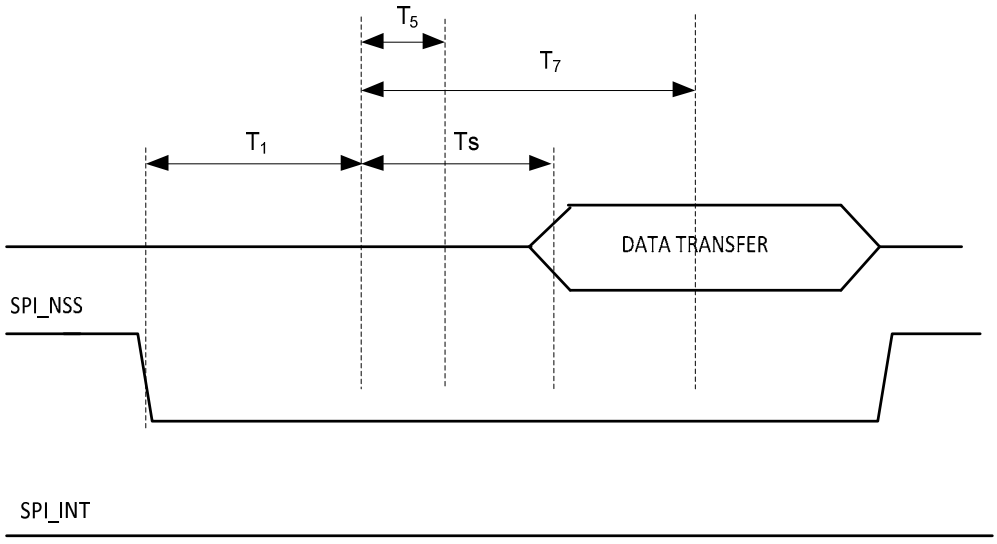


Figure 7.1: Initiation of the data transfer from the Master

The master shall run the following procedure:

1) Master asserts SPI_NSS then goes to the step 2.
2) Master waits at least for T1 seconds , but lower than T1+T7 if T7 is provided, then goes to the step 3.
3) Master starts the bidirectional data transfer by toggling the SPI_CLK signal.
4) Master de-asserts SPI_NSS after data transfer completion i.e. SPI_CLK stopped.

7.2.3.2 Initiation of the data transfer from the slave

Figure 7.2 illustrates the initiation by the slave of a MAC access request for a data transfer to be performed by the master.

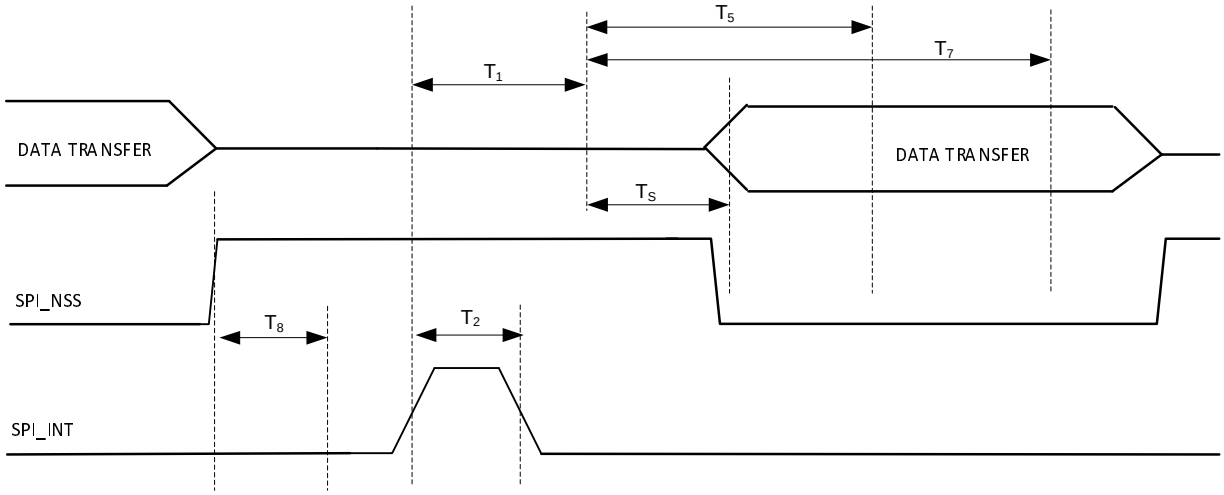


Figure 7.2: Initiation of the data transfer from the slave

The slave runs the following procedure:

1) In case the master reported T8 > 0, the slave shall wait for a time ≥ T8 from the de-assertion of the SPI_NSS at completion of the previous SPI access and then check the SPI_NSS status. If the slave determines that the SPI_NSS signal is de-asserted (i.e. at the high level state) by reading SPI_NSS_SI then the slave goes to the step 2.

2) The slave asserts SPI_INT by generating an SPI_INT pulse with a minimum width of T2 seconds then goes to the step 3.

3) Depending on its implementation, the slave configures its SPI module. The slave waits for data transfer from the master.

4) As a consequence of SPI_INT assertion by the slave at step 2, the master starts data transfer at a time greater than T1, but lower than T1+T7 if T7 is provided, following the leading edge of SPI_INT, by asserting SPI_NSS and then starts SPI_CLK. At data transfer completion, the master enters step 5.

5) After data transfer completion and SPI_CLK stop, the master de-asserts SPI_NSS.

7.2.3.3 Simultaneous initiation of a data transfer from both master and slave

Figure 7.3 illustrates a simultaneous initiation from the master and the slave.

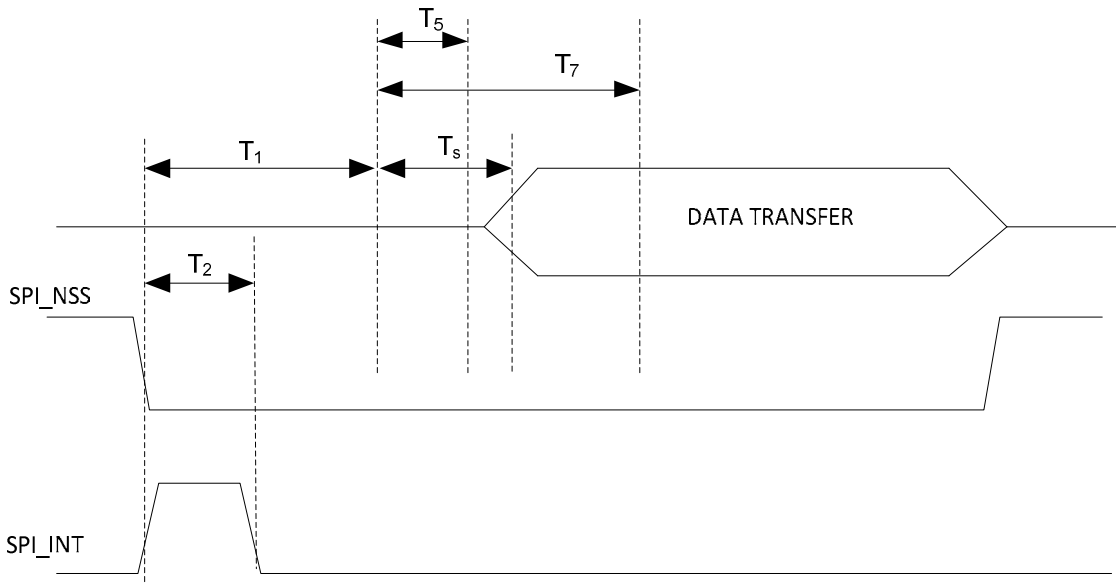


Figure 7.3: Simultaneous initiation of the data transfer from both master and slave

Both endpoints request a data transfer and run simultaneously their respective procedures. From the master perspective the resulting procedure is equivalent to the initiation from the master.

The slave makes a MAC access request by asserting SPI_INT according to the procedure described in clause 7.2.3.2 and waits for the master to generate the access for data transfer.

The time T1 - T2 shall be long enough for the slave to prepare the SPI module for the data transfer.

NOTE: It is recommended for the slave to report a T1 time assuming that the SPI master may not add the delay time Ts.

7.2.3.4 MAC activation

The MAC activation procedure at power on shall be the following:

- Master shall set the SPI_NSS output to the de-asserted state.
- The master shall drive the VDD power line on.

When the power supply of the slave needs to be toggled independently from the power of other devices sharing the SPI bus and while VDD is off or not valid, the slave shall keep SPI_NSS, SPI_CLK, SPI_MOSI and SPI_MISO lines as inputs or in high impedance.

7.2.3.5 MAC deactivation

The MAC deactivation procedure for power off shall be the following:

- Master shall set the SPI_NSS output to the de-asserted state.
- The master shall drive the VDD power line off.

7.2.4 4 signals MAC layer

7.2.4.1 Introduction

As a slave SPI module is normally always ready for an access from the master when selected by SPI_NSS assertion, the slave configures its SPI module in disabled state before asserting SPI_NSS for a MAC access request. This is necessary in order to avoid self-selection e.g. driving MISO in contention with MISO signals of other slaves potentially selected at the same time on the same bus.

7.2.4.2 Initiation of the data transfer from the master

Figure 7.4 illustrates the initiation of the data transfer by the master.

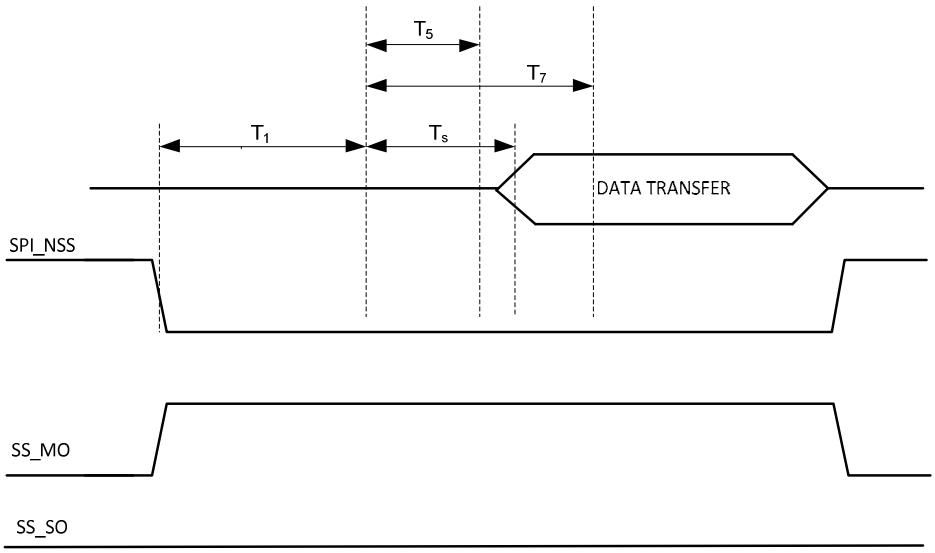


Figure 7.4: Initiation of the data transfer from the master

The Master shall perform the following procedure:

1) The master checks the state of the SPI_NSS signal (by reading SS_MI) and if it is de-asserted (i.e. at the high state) the master goes to the step 2 otherwise loops on the step 1.

2) The master asserts SS_MO (to drive SPI_NSS signal to the asserted state) then goes to the step 3.

3) The master waits for at least for T1 seconds, but lower than T1+T7 if T7 is provided, then goes to the step 4.

4) The master starts the bidirectional data transfer by toggling the SPI_CLK signal.

5) After data transfer completion i.e. SPI_CLK stop, the master de-asserts SPI_NSS.

7.2.4.3 Initiation of the data transfer from the slave

Figure 7.5 illustrates the initiation of the data transfer from the slave.

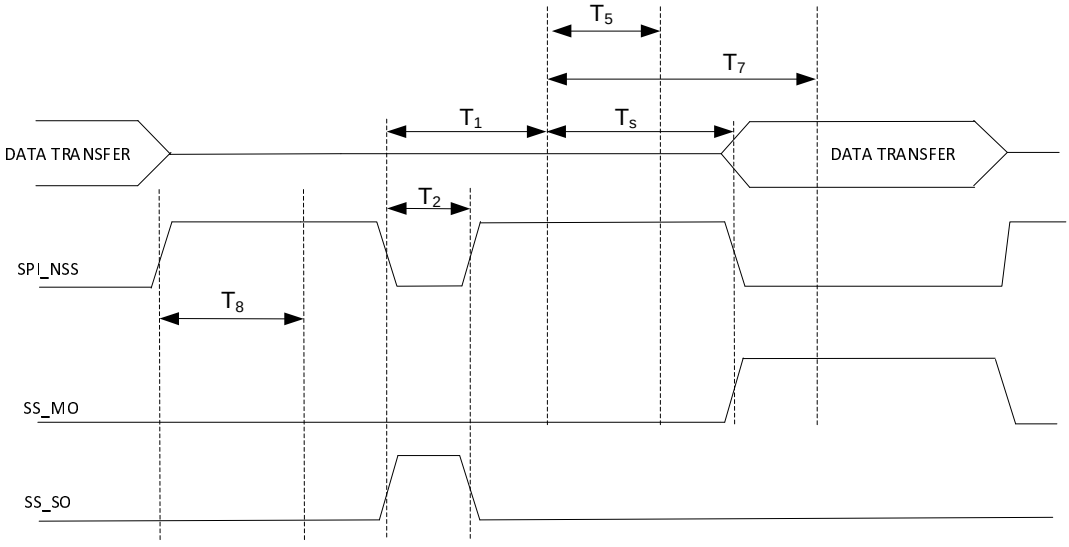


Figure 7.5: Slave initiates a MAC access request

The slave runs the following procedure:

1) In case the master reported T8 > 0, the slave shall wait for a time ≥ T8 from the de-assertion of the SPI_NSS at completion of the previous SPI access and then check the SPI_NSS status. If the SPI_NSS signal is found de-asserted i.e. at the high level state by reading SS_SI then the slave goes to the step 2.

2) The slave disables its SPI module then goes to the step 3.

3) The slave asserts SS_SO to drive SPI_NSS to the asserted state (i.e. low level) for at least T2 seconds then goes to step 4.

4) The slave enables its SPI module and waits for the master to initiate the data transfer.

5) The SS_MI signal generates an internal interrupt for the master, triggered on SPI_NSS falling edge. This interrupt initiates a data transfer procedure: the master asserts SS_MO after at least T1, but lower than T1+T7 if T7 is provided, following the SPI_NSS assertion by the slave for the MAC access request.

6) SPI_CLK starts after the SPI_NSS assertion by the master in the previous step, data transfer is performed.

7) After data transfer completion i.e. SPI_CLK stop, the master de-asserts SPI_NSS.

7.2.4.4 Simultaneous initiation of the data transfer from both master and slave

Figure 7.6 illustrates a simultaneous initiation from the master and the slave.

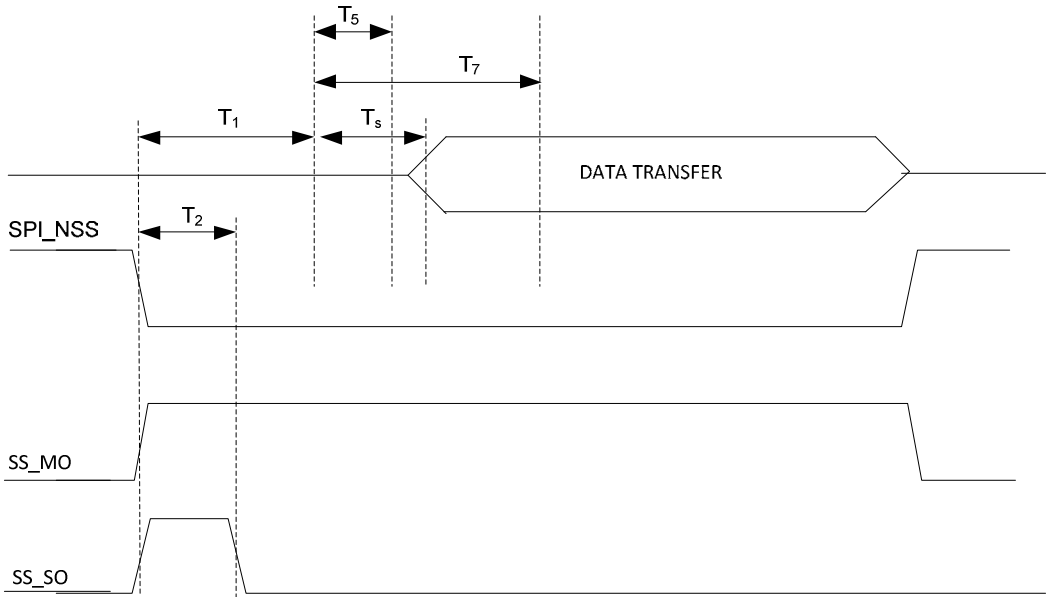


Figure 7.6: Simultaneous initiation of the data transfer from both master and slave

Both endpoints initiate a MAC phase for data transfers and run simultaneously their respective procedures. From a master perspective the resulting procedure is equivalent to the initiation from the master. The slave initiates a MAC access request and waits for the access start from the master as per the timings defined.

The time T1 - T2 shall be long enough for the slave to enable the SPI module.

NOTE: It is recommended for the slave to report a T1 time assuming that the SPI master may not add the delay time Ts.

7.2.4.5 Slave-driven Flow Control

In certain use-cases and depending on the SPI module design, a slave may assert SPI_NSS (via its SS_SO) during a data transfer in progress for flow control e.g. when the slave needs to delay a subsequent master access as the slave estimates it may not be ready to receive or transmit data. It is assumed that the slave has already prepared data to be sent (if available) for the current data transfer before the data transfer started. Such an assertion of SS_SO while the master access is in progress is not a slave MAC access request and a slave shall not start to send a new frame in the middle of the current access, as a new frame shall be always aligned with the start of an SPI access.

As long as the SPI_NSS signal is asserted and even if the data transfer is completed, the master shall not run the MAC initiation by the master procedure as defined in clause 7.2.4.2.

Slave may assert SS_SO for flow control at any time between the start of the data transfer and SPI_NSS de-assertion by master at end of the data transfer.

NOTE: Detection by the slave of a data transfer for initiating flow control could be performed either by sensing the assertion of the SPI_NSS by master or by detection of the first bytes transferred.

If the SPI bus is shared among multiple slaves and the slave performing flow control has its SPI module enabled during this time, the master shall not initiate a data transfer to any other slaves on the shared SPI bus as long as the slave keeps the SPI_NSS asserted. The slave informs the master about the applicability of this requirement in bit 4 of the slave capabilities in MCT_DATA, as shown in table 7.9.

The slave should not use that mechanism longer than 500 µs; after that duration, the master may use its own recovery mechanisms.

Figure 7.7 illustrates the waveforms for this procedure.

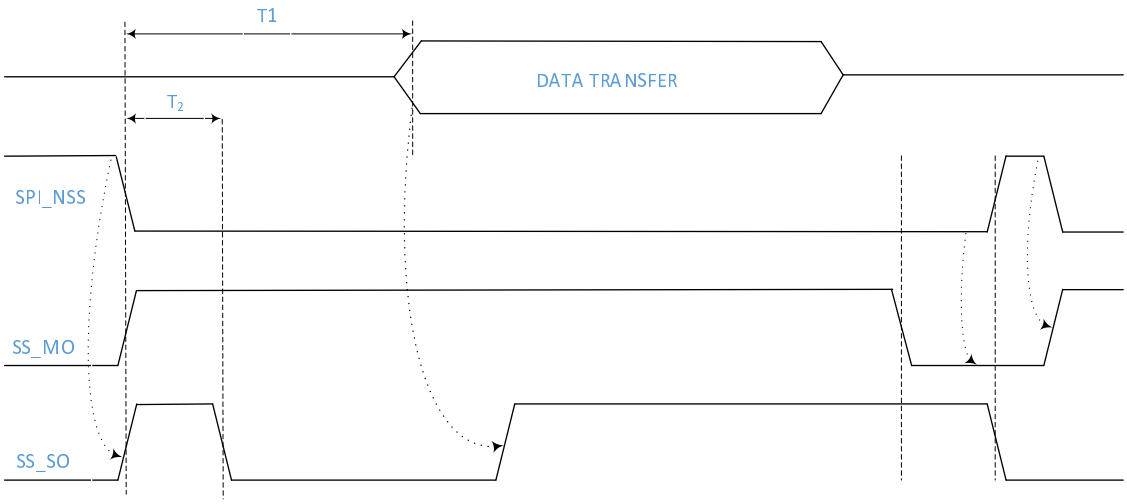


Figure 7.7: Slave activates hardware flow control by assertion of SS_SO

7.2.4.6 MAC activation

The MAC activation procedure at power on shall be the following:

- The SS_MO shall be de-asserted (i.e. at low level) setting the SPI_NSS output to high impedance.
- The master shall drive the VDD power line on.

When the power supply of the slave needs to be toggled independently from the power of the other devices sharing the same SPI bus and while VDD is off or not valid, the slave shall keep SS_SO de-asserted and SPI_CLK, SPI_MOSI and SPI_MISO as inputs or in high impedance.

7.2.4.7 MAC deactivation

The MAC deactivation procedure for power off shall be the following:

- The SS_MO shall be de-asserted (i.e. at low level) setting the SPI_NSS output to high impedance.
- The master shall drive the VDD power line off.

7.3 Link Layer Frame

7.3.1 Overview

Master and slave exchange frames. The format of the frames generated by master and slave is determined by the link layer and it is shown in figure 7.8.

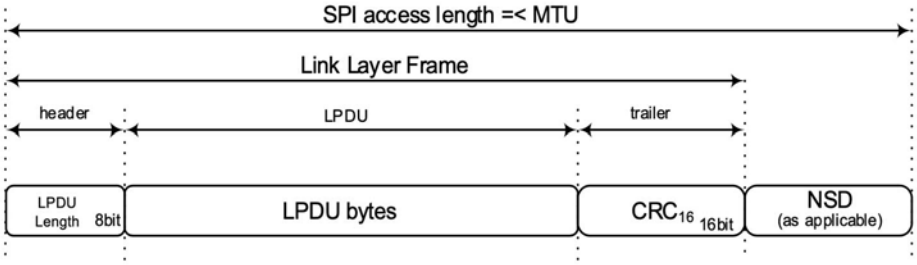


Figure 7.8: Link Layer Frame structure

All bytes shall be transmitted with Most Significant Bit first (MSB first). All frames are transferred with the bytes in the order shown in figure 7.8, starting with the LPDU Length. The LPDU field is transferred starting with the LLC Control byte followed by the data generated by the upper OSI layer.

The link layer frame shall contain the following fields:

- LPDU length: length of the LPDU, 1 byte.
- LPDU (Link Protocol Data Unit): LPDU includes the LLC control byte as defined in clause 7.4.
- CRC informs about the integrity of the whole frame i.e. Length and LPDU. Detection of errors in a frame shall be based on the 16-bit frame checking sequence as given in ISO/IEC 13239 [3]. The CRC polynomial is: $X^{16} + X^{12} + X^5 + 1$. Its initial value is 'FFFF'.

Link Layer Frames are exchanged between SPI master and slave during SPI accesses. The SPI master determines the number of bytes exchanged in an SPI access. The maximum length of an SPI access is MTU.

Link Layer frames (including header, LPDU and trailer) shall always be prepared with length less than or equal to MTU.

MTU values are negotiated at SPI interface initialization as described in clause 7.6. The resultant MTU shall be the smallest MTU value between the MTU of the master and the MTU of the slave. The MTU shall be the same irrespective of the transfer direction.

Non-Significant Data (NSD) may be appended at the end of a master or slave link layer frame until the end of the SPI access according to the rules described below, considering LPDU Length + 3 + NSD length ≤ MTU.

NSD bytes may have any values; in some particular implementations, the master and/or a slave may send a constant value of '00' or 'FF' instead of random values.

Master frames, slave frames or remaining bytes of a slave frame (i.e. in a second SPI access for retrieving a slave frame) shall always start aligned on the first bytes transmitted on SPI_MOSI and SPI_MISO at SPI_CLK start.

The LPDU Length value shall be compliant with the values indicated in table 7.2.

Table 7.2

LPDU Length	LPDU
'00'	Indicates that no frame is transferred (there is no LPDU)
'01' to '1D'	With MTU = 32
'01' to '3D'	With MTU = 64
'01' to '7D'	With MTU = 128
'01' to 'FD'	With MTU = 256
'FE'	RFU
'FF'	Indicates that no frame is transferred (not a valid LPDU length)

7.3.2 Frame generation and transfer rules

7.3.2.1 Overview

The SPI master initiates an SPI access either to send a frame, retrieve a frame from the slave after a MAC access request or both. Depending on the master and the slave specific implementations (as described in the following clauses), the SPI master retrieves the slave frames over one or two SPI accesses.

7.3.2.2 Generally applicable rules

If the SPI master has a frame to send, the SPI master shall send that frame in a single SPI access. The SPI master may initiate a SPI access with a length higher than the length of the frame to send.

In case the SPI access is longer than the length of the frame being sent, SPI master and/or slave shall add Non-Significant Data (NSD) bytes following the CRC until the end of the SPI access.

The SPI master may apply one of the two methods below for retrieving a slave frame, according to the information provided by the slave in bit 5 of the MCT_DATA field byte 0 (see table 7.9):

- If the slave sets this bit to 0, the SPI master shall retrieve the slave frames within a single SPI access.
- If the slave sets this bit to 1, the slave also allows the SPI master to retrieve its frames over two SPI accesses.

The length byte of any frame shall always be the first byte sent in an SPI access, i.e. a new frame shall not be started in the same SPI access. According to table 7.2, when there is no frame to transfer, the transmitter (i.e. the SPI master or the slave) shall set the first byte of the SPI access to '00' or 'FF' to indicate that it does not send any frame; this applies as well for the first MOSI byte of the second SPI access when a slave frame is retrieved over two SPI accesses.

In both cases above, regardless if it has a frame to transfer, the SPI master may generate the first SPI access of maximum length, i.e. MTU, in order to transfer the complete slave frame.

7.3.2.3 Slave frame retrieval in one SPI access

If the slave sets the bit 5 to 0 in the MCT_DATA field byte 0 (see table 7.9), the SPI master shall retrieve the slave frames within a single SPI access, i.e. without de-asserting SPI_NSS for the entire slave frame transfer, possibly with a gap when SPI_CLK is stopped (i.e. SPI_CLK is idle).

If the SPI master has a frame to transfer, it should transfer it prior to stopping the SPI_CLK.

The SPI master maintains the SPI_NSS asserted after retrieving one or more bytes of the slave frame, may stop the SPI_CLK while processing the slave frame length information acquired and then re-starts the SPI_CLK for retrieving any remaining bytes. The slave frame is retrieved in a single SPI access with a pause in SPI_CLK. The SPI master may pause the SPI_CLK anywhere during the SPI access while maintaining the SPI_NSS asserted. After processing the slave frame length information, if the SPI master finds out that the complete slave frame was transferred or there is no slave frame, then the SPI master de-asserts SPI_NSS. The maximum number of bytes transferred while retrieving the slave frame is limited to the maximum frame length, i.e. MTU. The SPI master should insert only one SPI_CLK pause, if any, during one SPI access.

This approach for retrieving the slave frame in one SPI access is also allowed for slaves which set bit 5 to 1 in the MCT_DATA field byte 0 (see table 7.9).

7.3.2.4 Slave frame retrieval in two SPI accesses

When the slave sets bit 5 to 1 in the MCT_DATA field byte 0 (see table 7.9), the SPI master may use two SPI accesses to retrieve a slave frame. The SPI master de-asserts the SPI_NSS when the first SPI access is completed and asserts again the SPI_NSS for the second SPI access for retrieving any remaining bytes of the slave frame. In this case, the SPI master should not insert gaps in the SPI_CLK during the first or the second access. A slave frame shall be retrieved in at most two SPI accesses if the number of bytes of the first SPI access is shorter than the slave frame. If the SPI master did not receive the entire slave frame in one SPI access, the master shall initiate a second SPI access with a length equal or greater than the number of remaining bytes of the frame to be retrieved from the slave. The total number of bytes transferred on MISO over both SPI accesses shall be less than or equal to the MTU.

In the second SPI access, the slave shall continue to send the same frame from the point where the previous SPI access stopped. The remaining part of a slave frame retrieved in a second SPI access shall start on the first byte of the second access with the byte following the last byte retrieved in the prior SPI access.

During the second SPI access for retrieving the remaining bytes of a slave frame, the SPI master shall set the first byte of the SPI access to '00' or 'FF' to indicate that it does not send any frame.

To retrieve a slave frame the SPI master may proceed with the following steps:

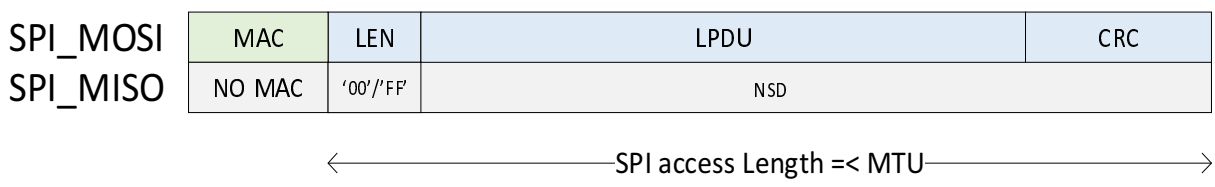
- When the SPI master does not have a frame to send and slave initiates a MAC access request asking for data transfer:
  - Step 1: the SPI master may generate an SPI access of minimum 1 byte to retrieve the slave frame length information.
  - Step 2: if the slave frame has not been entirely received in the first SPI access, the SPI master shall generate a second SPI access for retrieving the remaining bytes of the slave frame based on the information on the slave frame length received in the slave frame header during the first SPI access.
- When master has a frame to send:
  - Step 1: the SPI master generates an SPI access with the length equal or higher than its frame length. At the same time, the SPI master may receive on SPI_MISO part of or a full slave frame. When the SPI access length is greater than the slave frame length, the slave sends NSD bytes at the end of its link layer frame, following the CRC bytes.
  - Step 2: if the SPI access length in step 1 is less than the slave frame length, the SPI master shall generate a second SPI access for retrieving the remaining bytes of the slave frame based on the information on the slave frame length received in the slave frame header during the first SPI access.

7.3.3 Data transfer cases

Some of the most representative data transfer cases based on the frames generation and transfer rules in clause 7.3.2 are described below. Any frame sent by the SPI master or the slave shall be preceded by a MAC phase issued respectively by the SPI master or the slave. When two SPI accesses are required for transferring a slave frame, the SPI master shall generate the second SPI access at any time greater than or equal to the tCS value in clause 6.4.2.

Case 1: the SPI master initiates the MAC phase and then sends a frame. The SPI access length is determined by the master frame length. No frame is received from the slave.

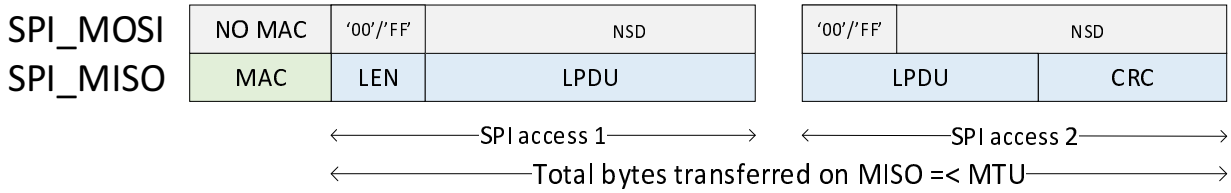
The slave sends on MISO the first byte set to '00' or 'FF' and the following bytes are NSD.



Case 2: the slave initiates a slave MAC access request to transfer a frame, while the SPI master has no frame to send.

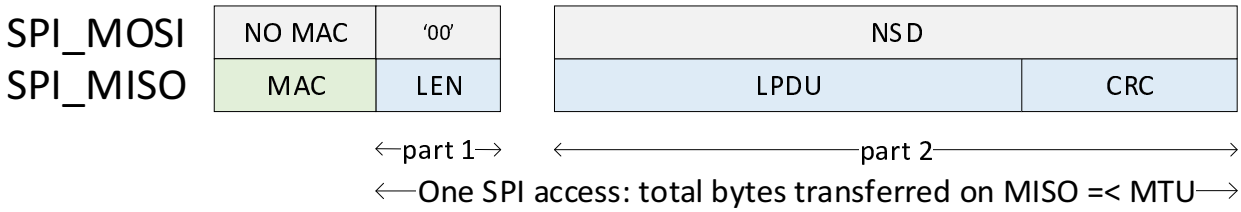
Case 2.1: the slave sets the bit 5 to 1 in the MCT_DATA field byte 0 (see table 7.9), the SPI master performs two SPI accesses for retrieving the slave frame. The SPI master may perform a first SPI access with the length e.g. equal to 4, i.e. the minimum frame length, or based on a best estimate, to retrieve the slave frame length. The SPI master then performs a second SPI access to retrieve the remaining bytes of the slave frame considering the length information from the first SPI access. The total number of bytes transferred on MISO over both SPI accesses is less than or equal to the MTU. For each SPI access, the SPI master sets the first byte it sends on MOSI to '00' or 'FF' followed by NSD bytes up to the end of the access to indicate it sends no frame.

In the following figure, the slave frame is retrieved over two SPI accesses.



Case 2.2: the SPI master retrieves the slave frame in a single SPI access as requested by the slave with the bit 5 set to 0 in the MCT_DATA field byte 0 (see table 7.9). In this case, the SPI master may start with transferring at least one byte or up to MTU bytes to retrieve at least the frame length information or the complete slave frame. The SPI master sets the first MOSI byte to '00' or 'FF', indicating that it does not send any frame. Following a gap with SPI_CLK stopped and SPI_NSS maintained asserted, the SPI master re-starts SPI_CLK for retrieving any remaining bytes of the slave frame. The SPI master sends NSD bytes for the remaining bytes of the SPI access.

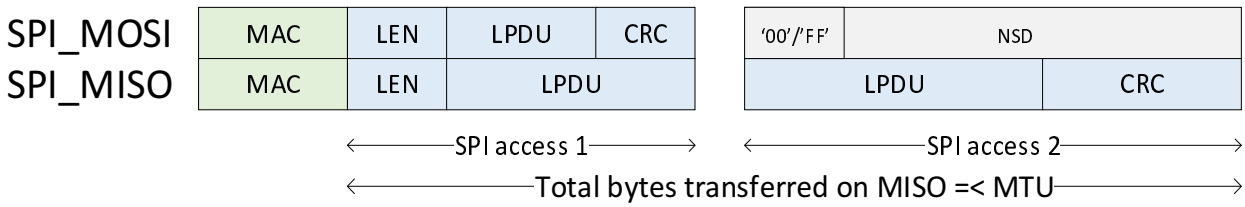
In the following figure, the slave frame is retrieved in a single access. SPI_NSS is maintained asserted from the first byte of the slave frame and until the last byte of the slave frame is transferred.



Case 3: both the SPI master and the slave have frames to transfer and MAC phase is initiated by both simultaneously.

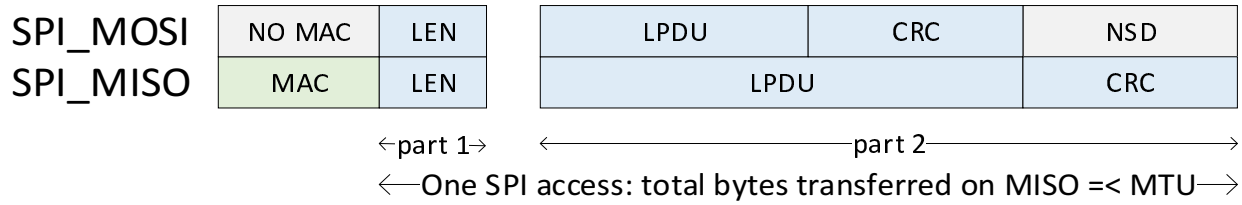
Case 3.1: the slave sets the bit 5 to 1 in the MCT_DATA field byte 0 (see table 7.9). In this example, the SPI master generates an SPI access with the length determined by its frame length and the length of the slave frame is longer than the length of the master frame. The SPI master finds out that only a part of the slave frame was received and generates a second SPI access to retrieve the remaining bytes of the slave frame. The SPI master sets the first byte of the second access to '00' or 'FF', followed by NSD bytes up to the end of the access.

In the following figure, the slave frame is retrieved over 2 SPI accesses.

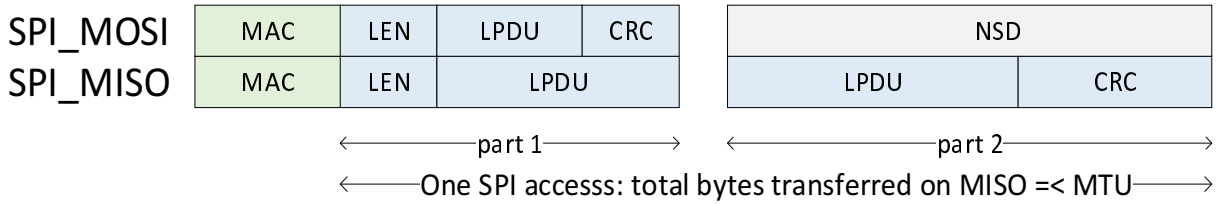


Case 3.2: the slave sets bit 5 to 0 in the MCT_DATA field byte 0 (see table 7.9). The SPI master may start the access transferring at least one byte, for retrieving the slave frame length. This is followed by a gap with SPI_CLK stopped for evaluating the slave frame length information. The SPI_CLK is then re-started for retrieving the remaining bytes of the slave frame. In this example, the slave frame is longer than the master frame and the SPI master adds NSD bytes after the end of its frame up to the end of the SPI access.

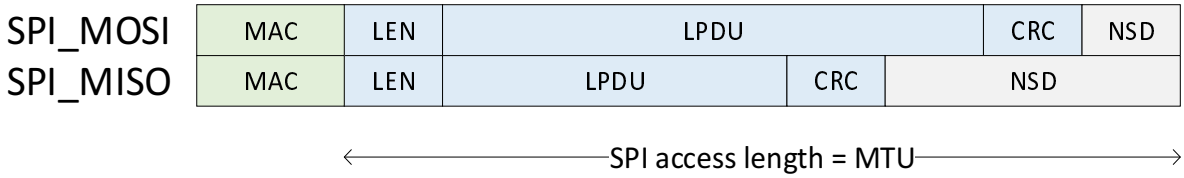
In the following figure, the slave frame is retrieved in a single access. The SPI_NSS is maintained asserted from the first byte of the slave frame and until the last byte of the slave frame is transferred.



In the following figure, the slave frame is retrieved in a single access. The SPI_NSS is maintained asserted from the first byte of the slave frame and until the last byte of the slave frame is transferred.



Case 3.3: the SPI master may generate an SPI access with the length equal to its frame length up to MTU to receive any slave frame occurring at the same time within a single access. In this case, the slave frame is shorter than the master frame and the slave adds NSD bytes after the end of its frame, up to the SPI access end. If the length of the SPI access is longer than the length of the master frame, e.g. equal to MTU, both the SPI master and the slave append NSD bytes after the end of their frames, up to access completion.



The green boxes in the figures above indicate the initiator(s) of the MAC phase.

7.4 LLC layers

Three Logical Link Control (LLC) layers are defined in the present document:

- SHDLC: this is the generic LLC. SHDLC is defined in ETSI TS 102 613 [2], clause 10. Support of this LLC is mandatory for the master and the slave.
- CLT: this LLC is used for some proprietary protocol handling. CLT mode is defined in ETSI TS 102 613 [2], clause 11. Support of this LLC is optional for the master and the slave.
- MCT: this LLC consists of frames used during interface activation. Support of this LLC is mandatory for the master and the slave.

The control field is the first byte of the LPDU. Definition for the different LLC layers can be found in table 7.3.

Table 7.3: LLC control field coding

Frame types	Bit field							
	8	7	6	5	4	3	2	1
RFU	0	0	0	All settings				
MCT	0	0	1	MCT type				
ACT (not used)	0	1	1					
CLT	0	1	0	CLT CMD				
SHDLC	1	All settings						

The LPDUs shall be structured according to figures 7.9, 7.10 or 7.11, depending on the frame type.

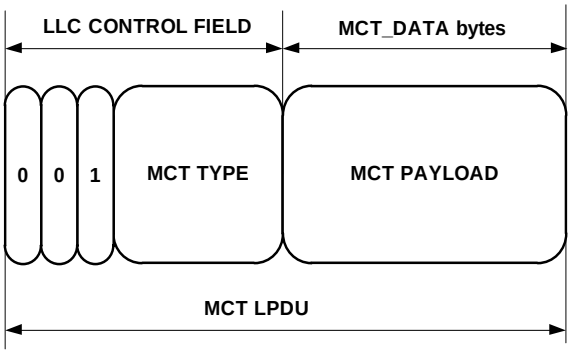


Figure 7.9: LPDU structure of the LLC layer of type MCT

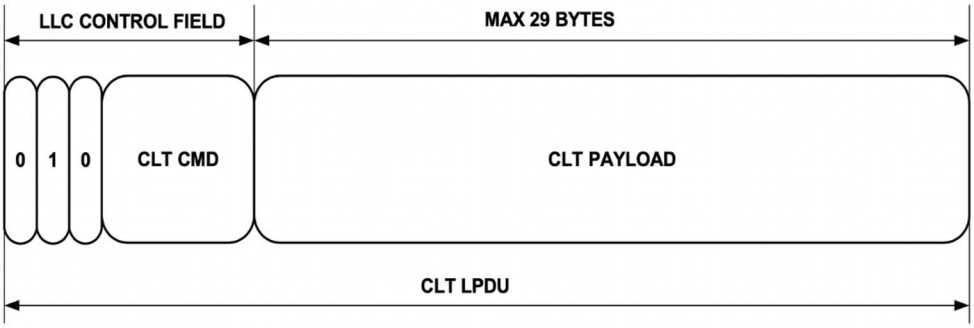


Figure 7.10: LPDU structure of the LLC layer of type CLT

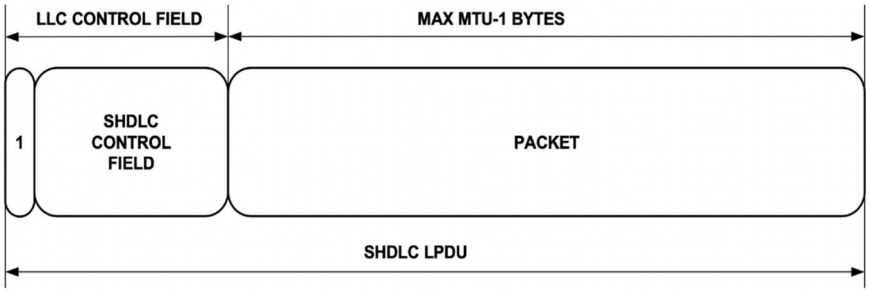


Figure 7.11: LPDU structure of the LLC layer of type SHDLC

7.5 Interworking of the LLC layers

After MAC activation, the SHDLC link shall not be established and no CLT session shall be open. Only the MCT LLC shall be used by the master and by the slave for the SPI interface initialization.

The master shall take the following action after a successful MCT LLC phase:

- If the master has data to be sent to the slave (e.g. due to a contactless transaction) that requires the use of the CLT LLC, it shall initiate a CLT LLC session.
- Otherwise it shall start the establishment of an SHDLC link as soon as possible.

After the slave and the master have established the SHDLC link or opened the CLT session, the slave and the master shall not send MCT LLC frames; received MCT LLC frames shall be ignored.

To enter the SHDLC LLC for the first time after MCT LLC, the link establishment procedure as described in clause 7.7.1 shall apply.

Once the SHDLC link is established, a CLT session shall not invalidate the SHDLC context and the endpoint capabilities negotiated during the SHDLC link establishment.

To enter the CLT LLC from MCT LLC or SHDLC LLC, the CLT session shall be opened as described in clause 11.6 of ETSI TS 102 613 [2]. The master shall open a CLT session only when all SHDLC I-Frames are acknowledged. SHDLC LLC frames received by the slave or by the master during a CLT session close the CLT session.

In case the slave or the master receives a corrupted frame, then the receiving entity shall use the error recovery procedure defined for the LLC of the last correctly received frame. Immediately after MAC activation, the error handling of the MCT LLC shall apply.

LPDU may be the SCL packet as defined in ETSI TS 103 666-1 [1], clause 8.3.2.

7.6 MCT LLC definition

7.6.1 MCT LPDU structure

The MCT LPDU shall be structured according to figure 7.12.

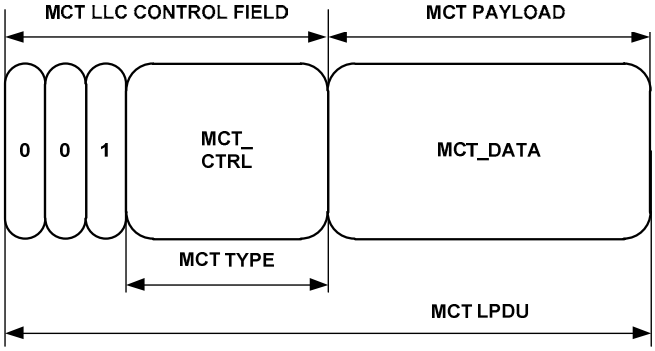


Figure 7.12: MCT LPDU structure

The meaning of MCT_CTRL and MCT_DATA is given in table 7.4.

Table 7.4: Meaning of MCT_CTRL and MCT_DATA

MCT_CTRL	Meaning	MCT_DATA
00000	MCT_READY Sent from Slave to Master	See table 7.5
00010	MCT_MASTER_REQ Sent from Master to Slave	See table 7.7
All other values (see note)	RFU	
NOTE: All other values are reserved for future use. These values shall not be set by the transmitting entity and shall be ignored by the receiving entity.		

The MCT LPDU length shall be lower than or equal to 29 bytes.

NOTE: 29 bytes is equal to the smallest MTU size (32 bytes) minus the overhead of the link layer frame as described in clause 7.3.1.

7.6.2 MCT_DATA from master

Tables 7.5, 7.6 and 7.7 define the capabilities of the master.

Table 7.5: Master-specific MCT_DATA field

Byte	Info/parameter	Meaning
0	Spec_Ver	Specification version to which master is compliant. Defined in table 7.6.
1	Capabilities	Defined in table 7.7.
2, 3	T4	Inactivity period for power saving mode (ms) according to clause 7.8.2.1.
4, 5, 6	T5	Master Ready Time (µs) according to clause 7.2.2.5.
7, 8, 9	T6	Master Resume Time from power saving mode (µs) according to clause 7.2.2.6.
10, 11	T8	Time (µs) required by the master, from the end of the SPI access to the start of a slave MAC access request, according to clause 7.2.2.8.
NOTE: All additional bytes (12 to 28) are reserved for future extensions of the protocol. They should not be sent by the master and shall be ignored by the slave.		

Table 7.6: Specification version

Bit field	Value	Meaning
8 to 4	00001	Major version of the SPI interface
3 to 1	001	Minor version of the SPI interface

Table 7.7: Master capabilities indication in MCT_DATA field

Bit field	Value	Meaning
8 to 6	000	RFU (see note)
5, 4	11	Full Power Mode 3 supported
	10	Full Power Mode 2 supported
	01	Full Power Mode 1 e.g. 10 mA supported
	00	Low Power Mode e.g. 5 mA supported
3, 2	11	MTU 256 bytes
	10	MTU 128 bytes
	01	MTU 64 bytes
	00	MTU 32 bytes
1	1	RFU
	0	Flow control SHDLC-based (default)
NOTE: These bits shall not be set by the master and shall be ignored by the slave.		

After the SPI activation as defined in clause 7.2.3.4 or in clause 7.2.4.6, the master shall send the MCT_MASTER_REQ frame and the slave shall respond with the MCT_READY frame.

The MCT phase shall be performed with default SPI_CLK = 1 MHz and T1 $\geq$ 255 µs.

The MTU negotiation between the master and slave shall be between the MTU sent by the master in the MCT_MASTER_REQ frame and the MTU sent by the slave in the MCT_READY. The lower of the MTU values will be used by both master and slave for all frames.

Master shall indicate in MCT_MASTER_REQ its highest power source capability (i.e. Low Power, Full Power Mode 1, Full Power Mode 2 or Full Power Mode 3) available for the respective slave. The power levels are defined as increasing respectively for Low Power Mode, Full Power Mode 1, Full Power Mode 2, Full Power Mode 3.

A slave able to limit its maximum current according to the received master power capabilities shall:

- support Low Power Mode;
- support Full Power Mode 1 if the slave requires a maximum current corresponding to Full Power Mode 1 or a higher current;
- start in Low Power Mode following VDD ON and may switch to a Full Power Mode, depending on the power mode capabilities received from the master in MCT_MASTER_REQ.

Low Power, Full Power Mode 1, Full Power Mode 2 and Full Power Mode 3 current values are out of scope of the present document. These power modes are described for general reference for the master and the slave when applicable and are not mandatory. The master and the slave may agree at the system design time on the implementation specific power requirements, on the power modes to be supported and specific current levels may be defined for the power modes. In a particular case the master includes in its power budget the maximum current required by the slave and in this case the slave may ignore the power modes information sent by master in MCT_DATA.

7.6.3 MCT_DATA from slave

Tables 7.8.and 7.9 define the capabilities of the slave.

Table 7.8: Slave-specific MCT_DATA field

Byte	Info/parameter	Meaning
0	Spec_Ver	Specification version to which slave is compliant. Defined in table 7.6.
1	Capabilities	Defined in table 7.9.
2	SPI_CLK	Max SPI_CLK value supported by SPI slave (MHz).
3	T1	Slave MAC Ready Time (µs).
4	T3	Slave resume time from power saving mode (µs) (see note 1).
5, 6	T4	Slave supported Inactivity period for entering power saving mode T4 (ms) according to clause 7.8.2.1.
7	POT	Power-ON Time: time after slave VDD valid when master can send MCT_MASTER_REQ (ms) (see note 2).
8, 9, 10	T7	Maximum delay time, after Slave MAC Ready Time for the data transfer start (µs) according to clause 7.2.2.7.
NOTE 1: In case a slave may "self-resume" (e.g. due to activities on another interface) and it has T3 < T1 slave shall report T1 value for T3. NOTE 2: This value should be used by master at next power on. The initial value used by the master should be 1 s. NOTE 3: All additional bytes (11 to 28) are reserved for future extensions of the protocol. They should not be sent by the slave and shall be ignored by the master.		

Table 7.9: Slave capabilities indication in MCT_DATA field byte 0

Bit field	Value	Meaning
8 to 6	0000	RFU (see note 1)
5	1	The master may perform two SPI accesses to retrieve a slave frame (see clause 7.3.2.4)
	0	The master shall perform a single SPI access for retrieving a slave frame (see clause 7.3.2.3)
4	1	Slave-driven flow control with SPI module enabled (see note 2)
	0	No slave-driven flow control master restrictions
3, 2	11	MTU 256 bytes
	10	MTU 128 bytes
	01	MTU 64 bytes
	00	MTU 32 bytes
1	1	RFU
	0	SHDLC-based flow control (default)
NOTE 1: These bits shall not be set by the slave and shall be ignored by the master. NOTE 2: The master shall postpone any transfers to other slaves on the shared bus as long as the slave is performing slave-driven flow control.		

7.6.4 MCT activation procedure

Slave start-up time following power-on is defined as POT and has an initial value of 1 s for the first power on, when the slave reported MCT_READY parameters are not yet available. After POT time (from the time VDD is valid after power-on) slave shall be ready to receive the MCT_MASTER_REQ from master. Shorter POT values may be reported by slave in MCT_READY. Master shall use the POT value reported by slave or a higher value in subsequent power-up sequences.

Master shall wait for MCT_READY from slave after sending MCT_MASTER_REQ. In case slave did not send MCT_READY response (no MAC access request) within MCT_SLAVE_TIMEOUT or if MCT_READY is corrupted, master shall retry the MCT activation by sending another MCT_MASTER_REQ frame. Master shall retry at least two times i.e. shall re-send MCT_MASTER_REQ at least twice without power toggle. Specific recovery procedure further steps after these retries are implementation specific and out of scope of the present document.

After power-up, if slave gets a corrupted frame or any other frame instead of the MCT_MASTER_REQ, slave shall discard the data and remain in receive state. If slave receives three corrupted or invalid frames instead of MCT_MASTER_REQ, slave should enter power saving mode. If the MCT activation has not been successfully performed or master did not initiate an SPI access within MCT_MASTER_TIMEOUT following power-on or for the subsequent retries in case of errors, slave should enter power saving mode.

MCT_MASTER_TIMEOUT is the maximum time within which the master shall send the MCT_MASTER_REQ after power-on or for the retries in case of errors. The value defined is 1 s.

MCT_SLAVE_TIMEOUT is the maximum time within which the slave shall send the MCT_READY response to MCT_MASTER_REQ. The default value defined is 200 ms. A slave may send MCT_READY faster according to certain applications requirements.

7.7 SHDLC LLC definition

7.7.1 SHDLC overview

The provisions of ETSI TS 102 613 [2], clause 10.1 shall apply. The SWP SHDLC layer is replaced by the SPI SHDLC layer defined in the present document.

The SHDLC layer shall ensure that data passed up to the next layer has been received exactly as transmitted i.e. error free, without loss and in the correct order. Also, the SHDLC layer manages the flow control, which ensures that data is transmitted only as fast as the receiver may receive it.

The provisions of ETSI TS 102 613 [2] clauses from 10.3 to 10.8 shall apply. Additional SHDLC rules are defined below.

7.7.2 Endpoints

SHDLC communication occurs between two endpoints. Those endpoints may be either the master endpoint or the slave endpoint. There is no priority of traffic.

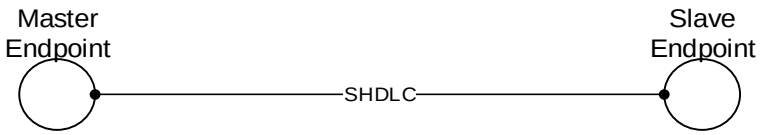


Figure 7.13: Endpoints

In ETSI TS 102 613 [2], clause 10, the term CLF refers to the master endpoint and the term UICC to the slave endpoint.

7.7.3 Flow control

7.7.3.1 Overview

Flow control is performed by a transmitter in order to avoid corruption or loss of data. It consists of methods applied by the transmitter and receiver in order to send a maximum number of SHDLC frames that can be accepted by the receiver, after which it shall stop sending data until the receiver sends at least an acknowledgement (e.g. SHDLC I-frame or SHDLC S-frame) for one of the received SHDLC frames.

7.7.3.2 Flow control based on SHDLC

The method defined in this clause is based on SHDLC flow control, as defined in ETSI TS 102 613 [2].

In addition to the provisions of clause 7.3.2, the total number of bytes transferred on SPI_MOSI while retrieving a slave frame over two SPI accesses shall be less than or equal to the maximum slave frame length i.e. MTU.

NOTE: The number of free window size slots is not to be decremented for the second SPI access for retrieving the remaining bytes of a slave frame.

The maximum number of SHDLC frames that can be sent by a transmitter is determined by the negotiated window size.

7.8 Power management

7.8.1 Power saving mode

In order to optimize the power consumption, both the master and the slave may enter into power saving mode. This clause defines the conditions for the slave to enter into power saving mode and the procedures for the master for resuming the slave from power saving mode.

The master and the slave shall both resume in the same LLC context following the master or the slave resumption. The master and the slave may then initiate the switch to a different LLC context according to clause 7.5.

7.8.2 Conditions for entering power saving mode

7.8.2.1 Slave entering power saving mode

The slave shall not enter into power saving mode, if the slave has issued a MAC access request and is waiting for data transfer from the master.

The slave may enter into power saving mode in one of the cases below, assuming there is no pending activity:

1) All frames have been transmitted by the slave, acknowledged successfully by the master and the slave detects an inactivity time T4 with no assertion of SPI_NSS by the master. Entering into power saving mode based on the inactivity period may be disabled by the master thru the inactivity period value negotiated at interface initialization as described below.

2) The slave informs that it does not require or expect any further activities e.g. through EVT_LINK_END_OF_OPERATION sent from its Link Application Gate to the Link Service Gate of the Network Controller Host, as defined in ETSI TS 103 666-1 [1]. The slave may enter into power saving mode after it receives the SHDLC link layer acknowledgement for EVT_LINK_END_OF_OPERATION.

3) Following the power-up or during MCT activation procedure as described in clause 7.6.4, when the slave does not detect any activity for more than the default inactivity period MCT_MASTER_TIMEOUT.

The inactivity period T4 is negotiated by the master and the slave at interface initialization as described in clause 7.6. The master shall provide an inactivity period in MCT_MASTER_REQ and the slave shall send back an MCT_READY with the same T4 value for acceptance, or a different value if it cannot support the value received from the master. If the master sends T4='FFFF' in MCT_MASTER_REQ, the slave shall disable the entering power saving mode on detection of an inactivity period and the slave shall send an MCT_READY with the same T4 value. If the slave sends T4='FFFF' in MCT_READY it indicates to the master that the slave will not enter into power saving mode based on the detection of inactivity time, regardless the value sent by the master in the MCT_MASTER_REQ and the master should not resume the slave on this condition.

A slave which supports a resume time T3 lower than T1 and is either not able to systematically enter into power saving mode (i.e. it may remain active) or may resume independently of the activity on the SPI interface, shall report in MCT_READY a T3 value equal to T1 (or higher).

In power saving mode, the slave shall maintain its SPI interface as for the case when SPI_NSS is de-asserted with the slave not in power saving mode.

7.8.2.2 Master entering power saving mode

The master may enter into an implementation-specific power saving mode at any time, without informing the slave. The slave power source status and capabilities indicated by the master at interface initialization shall not change when the master enters into power saving mode, is in power saving mode, is resuming from power saving mode or after the master has resumed. SPI_NSS shall be maintained de-asserted by the master when the master is in power saving mode and when the master is resuming.

If the slave has requested a valid T7 different from 'FFFFFF' and if one of the following conditions is met:

- the slave has requested T1+T7 lower than the SPI master resume time T6 when T6 is different from 'FFFFFF'; or
- the SPI master indicates no support for a maximum resume time i.e. T6 is 'FFFFFF';

then the SPI master shall only enter into power saving mode after it sends the SHDLC link layer acknowledgement for EVT_LINK_END_OF_OPERATION sent by the slave.

7.8.3 Resuming from power saving mode

7.8.3.1 Resuming the slave from power saving mode

Resuming the slave from power saving mode shall be performed by the master when any of the conditions above for the slave to enter into power saving mode has been previously met. The master shall ensure that all signals it drives are in the idle state corresponding to SPI mode 0 before initiating the resuming of the slave. The leading edge of the SPI_NSS assertion shall trigger the slave to resume.

The master shall perform the following procedure to resume the slave:

1) In the case of a 4 signals SPI interface, the master checks the state of the SPI_NSS signal and if it is de-asserted, it goes to the step 2, otherwise loops on step 1.
   In the case of a 5 signals SPI interface, the master starts with step 2.
2) The master asserts SPI_NSS and waits for at least T3, then goes to step 3.
3) The master considers the slave as resumed from the power saving mode and starts the data transfer. The slave shall support the resumption up to the data transfer phase with SPI_NSS continuously asserted by the master during T3.

7.8.3.2 Resuming the master from power saving mode

If the master has entered into power saving mode, it shall resume when the slave initiates a MAC access request. The slave initiates the MAC access request with the procedures as described in clause 7, regardless the power management status of the master. Following a resume by a MAC access request during T2 as described in clause 7.2.2.3, the master shall start an SPI access after T1 or later. The SPI access shall start not later than T1+T7 if T7 was provided by the slave. If T6 was provided by the SPI master and T7 not provided by the slave or if the slave sent 'FFFFFF' for T7, the SPI master should start the SPI access not later than T6. This duration starts from the leading edge of the slave MAC access request pulse, i.e. clauses 7.2.3.2 and 7.2.4.3 apply.

The leading edge of the MAC access request (SPI_NSS or SPI_INT assertion) should trigger the resuming of the master.

Annex A (informative): Slave SPI interface states electrical description

A.1 Slave SPI interface for 5 wire interface

A.1.1 Slave SPI 5 wire interface diagram

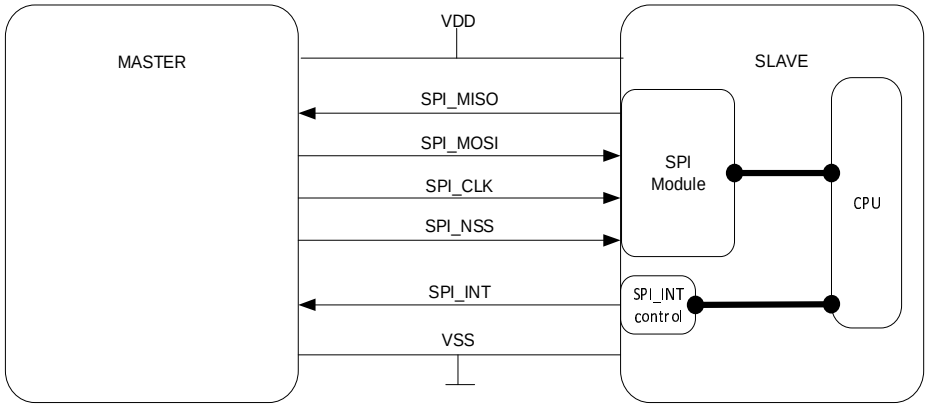


Figure A.1: SPI electrical interface with 5 signals

SPI_INT is generated separately from the SPI module which does not include the SPI_INT by default. SPI_INT is an output of the slave and an input to the master.

A.1.2 Slave SPI 5 wire interface states electrical description

The signals SPI_MISO, SPI_MOSI and SPI_CLK may be shared between multiple slaves. In that case each slave has a dedicated SPI_NSS. Shared SPI_INT is out of the scope of the present document.

The slave allows SPI shared bus and states according to table A.1.

Table A.1: 5 wire SPI interface states

slave state	SPI_INT		SPI_NSS		SPI_CLK		SPI_MISO		SPI_MOSI	
	slave	master	slave	master	slave	master	slave	master	slave	master
initial state (see note 1)	HiZ or DA	II	II or HiZ	DA	II or HiZ	Idle or active (see note 2)	HiZ	II or IL (see note 2)	II or HiZ	Idle or active (see note 2)
configured/ de-selected	HiZ or DA	IL	IL	DA	II	Idle or active (see note 2)	HiZ	II or IL (see note 2)	II	Idle or active (see note 2)
configured/selected	HiZ or DA	IL or II	IL	A	IL	D	D	IL	IL	D
pro-active (see note 3)	A	IL	IL	DA	II	Idle or active (see note 2)	HiZ	II or IL (see note 2)	II	Idle or active (see note 2)
power saving mode	HiZ or DA	IL	IL	DA	II	Idle or active (see note 2)	HiZ	II or IL (see note 2)	II	Idle or active (see note 2)

NOTE 1: State after VDD being valid until the slave internal configuration is completed.
NOTE 2: Master may be on an ongoing communication with another slave.
NOTE 3: Applies only during T2.

A.2 Slave SPI interface states for 4 wire interface

A.2.1 Slave SPI 4 wire interface diagram

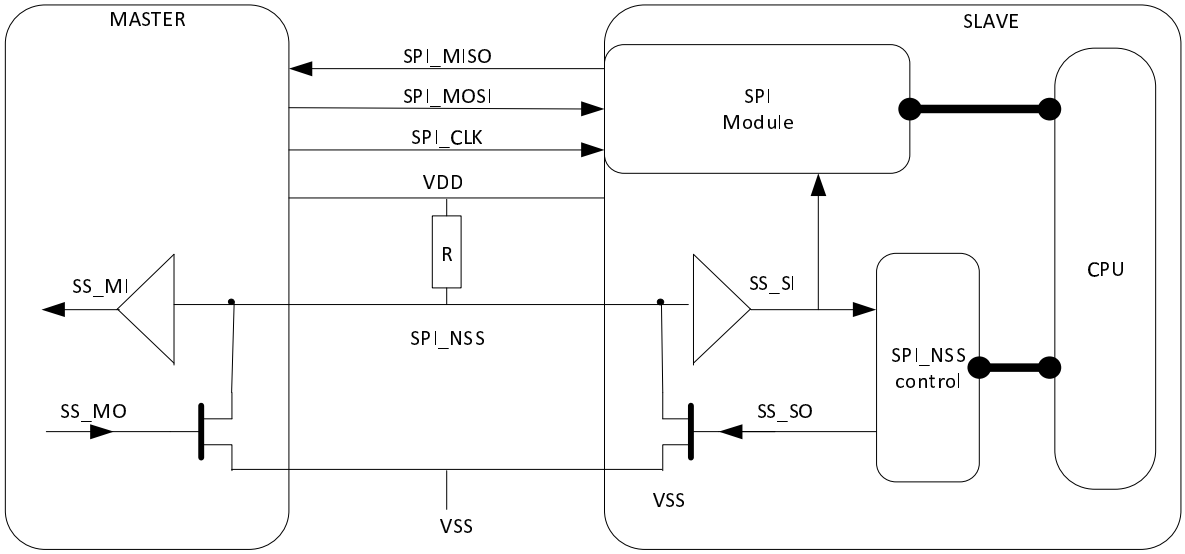


Figure A.2: SPI interface with 4 wire

Slave asserts the SPI_NSS signal independently from the SPI module in which the slave select signal is an input.

A.2.2 Slave SPI 4 wire interface states electrical description

The signals SPI_MISO, SPI_MOSI and SPI_CLK may be shared between multiple slaves. In that case each slave has a dedicated SPI_NSS.

Slave allows SPI shared bus and states according to table A.2.

Table A.2: 4 wire SPI interface states

slave state	SPI_NSS		SPI_CLK		SPI_MISO		SPI_MOSI	
	slave	master	slave	master	slave	master	slave	master
initial state (see note 1)	II or HiZ	DA	II or HiZ	Idle or active (see note 2)	HiZ	II or IL (see note 2)	II or HiZ	Idle or active (see note 2)
configured/ de-selected	DA, IL (see note 3)	DA, IL (see note 3)	II	Idle or active (see note 2)	HiZ	II or IL (see note 2)	II	Idle or active (see note 2)
configured/ selected	DA, A or IL (see notes 3 and 8)	A	IL	D	D	IL	IL	D
busy (see note 4)	A	DA, IL (see note 3)	IL or II (see note 7)	Idle or active (see notes 2 and 7)	D or HiZ (see note 7)	II or IL	IL or II (see note 7)	Idle or active (see notes 2 and 7)
pro-active (see note 5)	A	IL or A (see note 3 or 6)	II	Idle or active (see note 2)	HiZ	II or IL (see note 2)	II	Idle or active (see note 2)
power saving mode	DA, IL (see note 3)	DA, IL (see note 3)	II	Idle or active (see note 2)	HiZ	II or IL (see note 2)	II	Idle or active (see note 2)

NOTE 1: State after VDD being valid until the slave internal configuration is completed.
NOTE 2: Master may be on an ongoing communication with another slave.
NOTE 3: The state is not enforced (SS_MO and SS_SO not asserted), the line is sampled (SS_MI and SS_SI).
NOTE 4: State SPI_NSS is asserted only by slave after data transfer according to clause 7.2.4.5.
NOTE 5: Applies only during the time T2.
NOTE 6: Applies also during simultaneous assertion of SPI_NSS by master and slave for MAC phase.
NOTE 7: According to Busy state description in clause 6.5.1 and slave capabilities information in table 7.9 bit 4.
NOTE 8: Slave may assert its SPI_NSS at the same time with master SPI_NSS to enter into the busy state.

Annex B (informative): Change history

The table below indicates all changes that have been incorporated into the present document since it was placed under change control.

Change history								
Date	**Meeting**	**Plenary Doc**	**CR**	**Rev**	**Cat**	**Subject/Comment**	**Old**	**New**
03/10/2019	SCP#89	SCP(19)000211	-	-	-	Version 15.0.0 first publication	-	15.0.0
07/01/2020	SCP#90	SCP(19)000247	001	-	F	CR 103 713 Rel-15 MCT LLC RFU additions	15.0.0	15.1.0
07/01/2020	SCP#90	SCP(19)000248	002	-	D	CR 103 713 Rel-15 Clarification of the naming of the signal 'master internal interrupt'	15.0.0	15.1.0
07/01/2020	SCP#90	SCP(19)000249	003	-	F	CR 103 713 Rel-15 Data Transfer definition	15.0.0	15.1.0
07/01/2020	SCP#90	SCP(19)000250	004	-	C	CR 103 713 Rel-15 AC characteristics clarifications	15.0.0	15.1.0
07/01/2020	SCP#90	SCP(19)000251	005	-	C	CR 103 713 Rel-15 DC characteristics clarifications	15.0.0	15.1.0
07/01/2020	SCP#90	SCP(19)000252	006	-	B	CR 103 713 Rel-15 Power Management	15.0.0	15.1.0
01/04/2020	SCP #91	SCP(20)091023	007	-	F	CR 103 713 Rel-15 SPI Resume Time	15.1.0	15.2.0
01/04/2020	SCP #91	SCP(20)091024	008	-	F	CR 103 713 Rel-15 SPI_INT description	15.1.0	15.2.0
01/04/2020	SCP #91	SCP(20)091025r1	009r1	-	F	CR 103 713 Rel-15 MAC SPI interface states definition	15.1.0	15.2.0
01/04/2020	SCP #91	SCP(20)091026	010		D	CR 103 713 Rel-15 Editorial Changes	15.1.0	15.2.0
01/04/2020	SCP #91	SCP(20)091027	011	-	F	CR 103 713 Rel-15 SPI slave flow control clarification	15.1.0	15.2.0
28/07/2020	SCP#94	SCP(20)000046r1	12r1	-	F	CR 103 713 Rel-15 Slave interface states	15.2.0	15.3.0
28/07/2020	SCP#94	SCP(20)000047r1	13r1	-	F	CR 103 713 Rel-15 MAC activation	15.2.0	15.3.0
23/09/2020	SCP#95	SCP(20)000106	14	-	F	CR 103 713 Rel-15 Power Modes updates	15.2.0	15.3.0
16/12/2020	SCP#97	SCP(20)000161	015	-	F	CR_103_713_Rel-15_AC_DC_Characteristics_Updates	15.3.0	15.4.0
19/03/2021	SCP#99	SCP(21)000042	016	-	B	CR 103 713 Rel-15 Improvement of the NSD description and introduction of new transfer methods	15.4.0	15.5.0
19/05/2021						Editorial correction of the Annex A structure	15.3.1	15.3.1
19/05/2021						Editorial correction of the Annex A structure	15.4.1	15.4.1
19/05/2021						Editorial correction of the Annex A structure	15.4.0	15.5.0
29/09/2021	SCP#101	SCP(21)000138	017	-	F	CR 103 713 Rel-15 Data transfer starting time clarification	15.5.0	15.6.0
06/01/2022						Editorial corrections: removal of text duplications added during the pre-processing in several clauses	15.6.0	15.6.1
24/03/2022						Automatic upgrade	15.6.1	16.0.0
24/03/2022	SET#104	SET(22)000056r1	018r1	-	B	CR 103 713 Rel-16 Master maximum time for MAC access request	15.6.1	17.0.0
07/12/2022	SET#108	SET(22)000224r1	019r1	-	F	CR 103 713 Rel-17 MAC Access Request Delay	17.0.0	17.1.0
07/12/2022	SET#108	SET(22)000225	020		F	CR 103 713 Rel-17 Electrical Characteristics Update	17.0.0	17.1.0
09/03/2023	SET#109	SET(23)000034	021		B	CR 103 713 Rel-18 Class D Definition	17.1.0	18.0.0

History

Document history		
V18.0.0	July 2023	Publication