Socle Technology Corporation

SPI Controller with Booting Function IP Specification

Socle Technology reserves the right to change the information contained in this document to improve the function, design, or other aspects of this document. Socle Technology does not assume any liability for the application or use of this information, or for any error or omission in such information. Any warranties, whether express, statutory, implied, or otherwise, including but not limited to the implied warranties of merchantability or fitness for a particular purpose, are excluded. Any license under patent rights or any other intellectual property rights owned by Socle Technology or third parties shall be conveyed by Socle Technology in a separate license agreement between Socle Technology and the licensee.

Trademark

SoC_ImP®, µPlatform,® and the Socle logo are the trademarks of Socle Technology. All other trademarks referred to herein are the property of their respective owners.

REVISION HISTORY

Rev.	Date	By	Description
1.0	2011/11/28	Allen	Initial version

TABLE OF CONTENTS

SPI Controller with booting function

Overview

The Serial Peripheral Interface (SPI) controller with booting function is divieded into two parts:
1) General SPI controller via APB interface, 2) Serial NOR flash controller via AHB interface.

The two parts are introduced as follow:

1) General SPI controoler via APB interface: the master/slave controller controlled via APB interface is a full-duplex, synchronous, serial data link that is standard across many microprocessors, microcontrollers, and peripherals. This controller enables communication between microprocessors and peripherals for inter-processor communication. The SPI master/slave controller system is flexible enough to interface directly with numerous commercially-available peripherals.The SPI master/slave controller is compatible with the above-mentioned protocols as SPI bus master or slave. On the host side, the core acts like an APB-compliant slave device.
2) Serial NOR flash controller via AHB interface: the serial NOR flash controller is used to control serial NOR flash. The controller enables booting function from serial NOR flash via the AHB interface.

Key Features

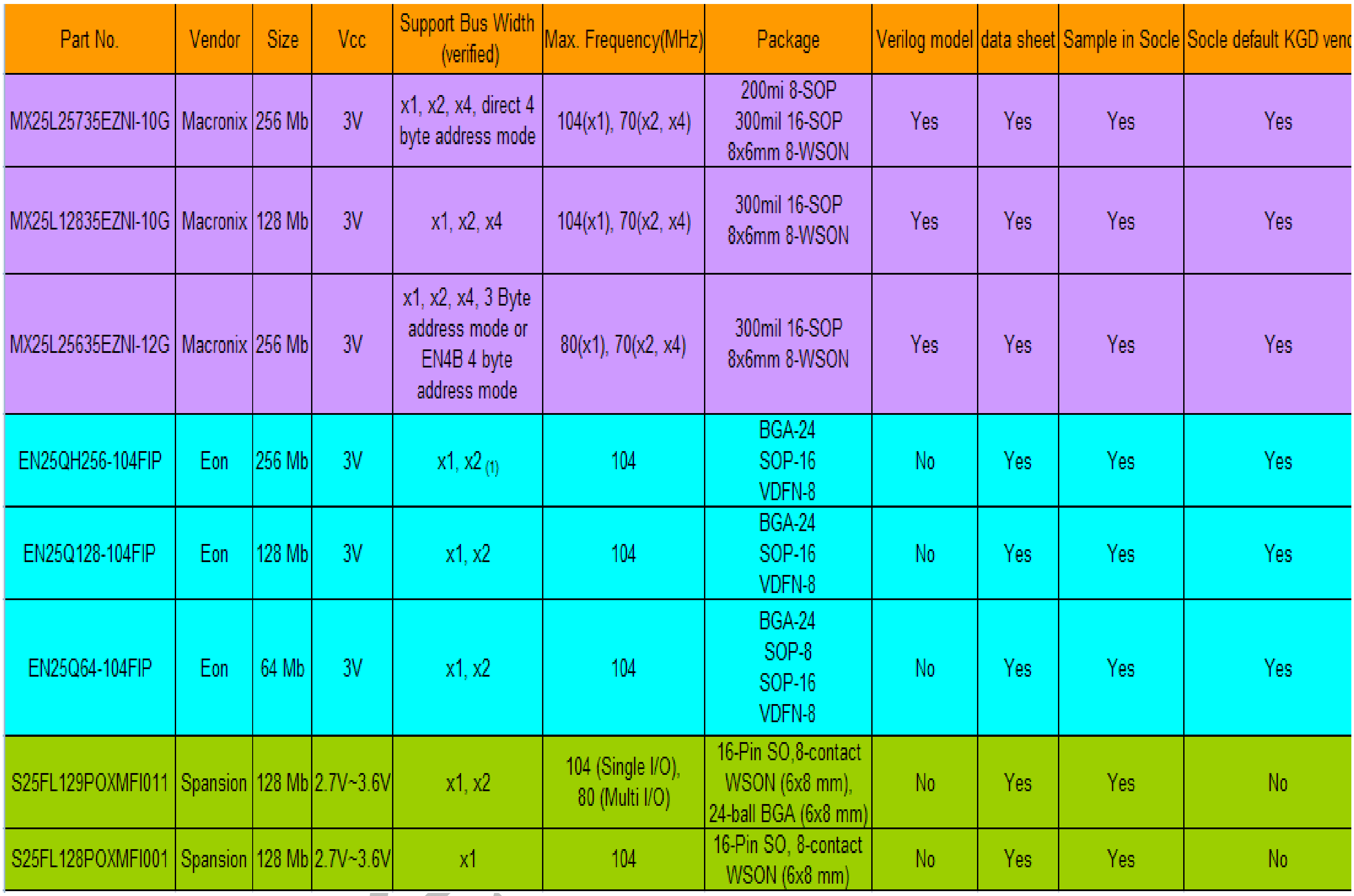
1) General SPI controoler via APB interface:

- AMBA APB slave interface
- Support for master or slave mode
- DMA Interface
- Four transfer protocols available, with selectable clock polarity and clock phases
- Different bit rates available for the SCLK
- Full duplex synchronous serial data transfer
- Bi-direction mode
- Support for 2 slave devices
- MSB or LSB first data transfer

2) Serial NOR flash controoler via AHB interface:

- AMBA AHB slave interface
- Automatic conversion of read transfer to serial NOR flash read cycle on AHB slave interface
- Support 1 I/O, 2 I/O (the list of flash supported is as following table)
- Support 4 I/O mode only for MX25L2573EZNI-10G, MX25L12835EZNI-10G, MX25L25635EZNI-12G (the list of flash supported is as following table)
- Support 3 byte address, direct 4 byte address, EN4B 4 byte address mode(the list of flash supported is as following table)

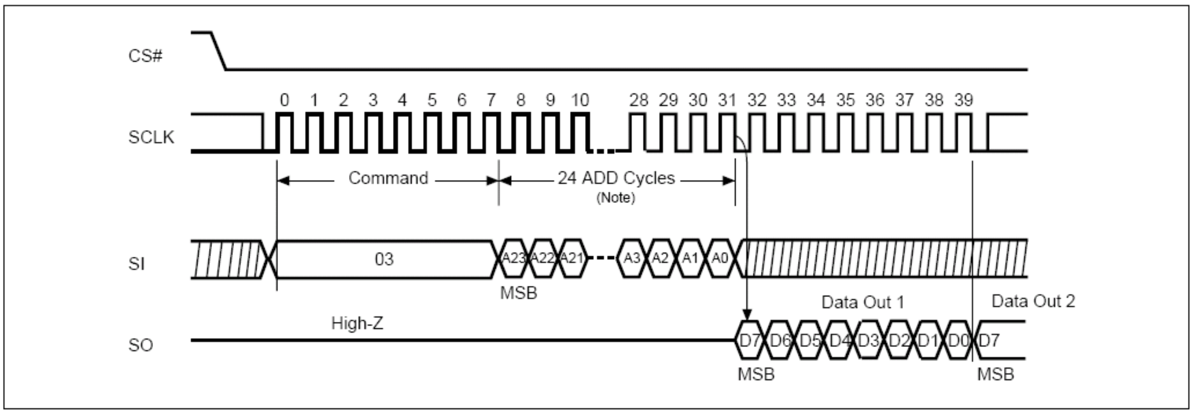
Supported flash and I/O mode



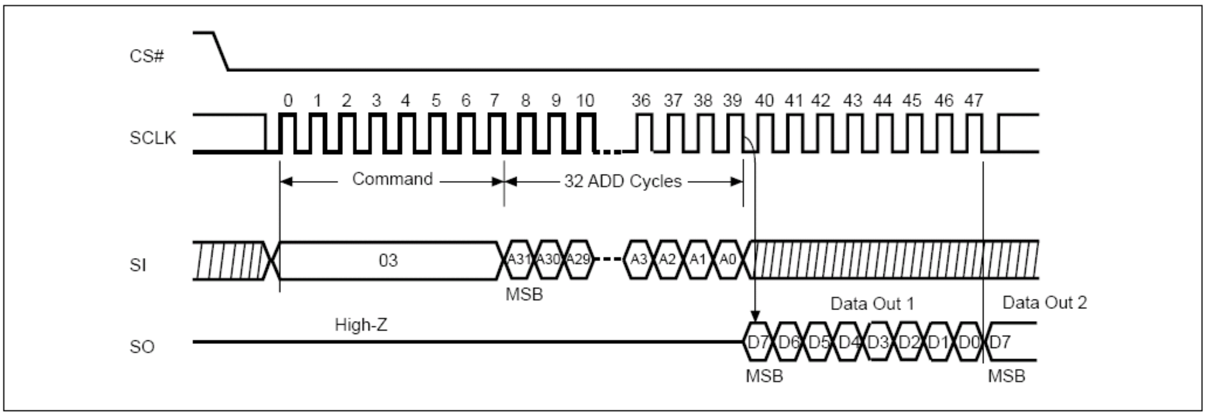
Part No.	Vendor	Size	Vcc	Support Bus Width (verified)	Max. Frequency(MHz)	Package	Verilog model	data sheet	Sample in Socle	Socle default KGD ven
MX25L25735EZNI-10G	Macronix	256 Mb	3V	x1, x2, x4, direct 4 byte address mode	104(x1), 70(x2, x4)	200mi 8-SOP 300mil 16-SOP 8x6mm 8-WSON	Yes	Yes	Yes	Yes
MX25L12835EZNI-10G	Macronix	128 Mb	3V	x1, x2, x4	104(x1), 70(x2, x4)	300mil 16-SOP 8x6mm 8-WSON	Yes	Yes	Yes	Yes
MX25L25635EZNI-12G	Macronix	256 Mb	3V	x1, x2, x4, 3 Byte address mode or EN4B 4 byte address mode	80(x1), 70(x2, x4)	300mil 16-SOP 8x6mm 8-WSON	Yes	Yes	Yes	Yes
EN25QH256-104FIP	Eon	256 Mb	3V	x1, x2 (1)	104	BGA-24 SOP-16 VDFN-8	No	Yes	Yes	Yes
EN25Q128-104FIP	Eon	128 Mb	3V	x1, x2	104	BGA-24 SOP-16 VDFN-8	No	Yes	Yes	Yes
EN25Q64-104FIP	Eon	64 Mb	3V	x1, x2	104	BGA-24 SOP-8 SOP-16 VDFN-8	No	Yes	Yes	Yes
S25FL129POXMFI011	Spansion	128 Mb	2.7V~3.6V	x1, x2	104 (Single I/O), 80 (Multi I/O)	16-Pin SO,8-contact WSON (6x8 mm), 24-ball BGA (6x8 mm)	No	Yes	Yes	No
S25FL128POXMFI001	Spansion	128 Mb	2.7V~3.6V	x1	104	16-Pin SO, 8-contact WSON (6x8 mm)	No	Yes	Yes	No

The following illustrates the waveform on the serial normal flash for different I/O mode and addressing mode (reference to Macronix MX25L25635E, MX25L25735E, MX25L12835E)

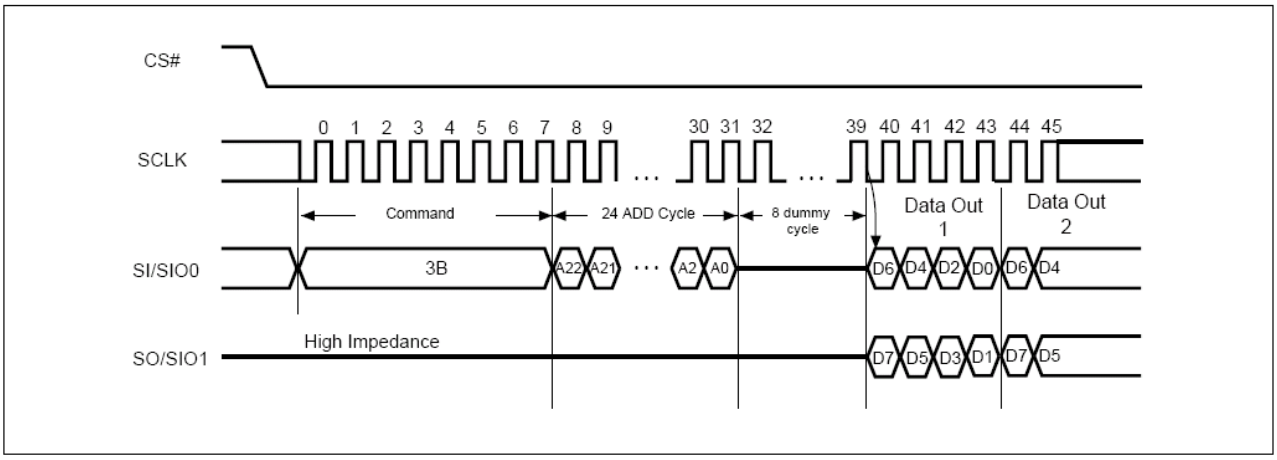
1) read data sequencye (1 I/O, command 03h, 3 byte address mode), for 4 byte address mode, the address cycles will be increased



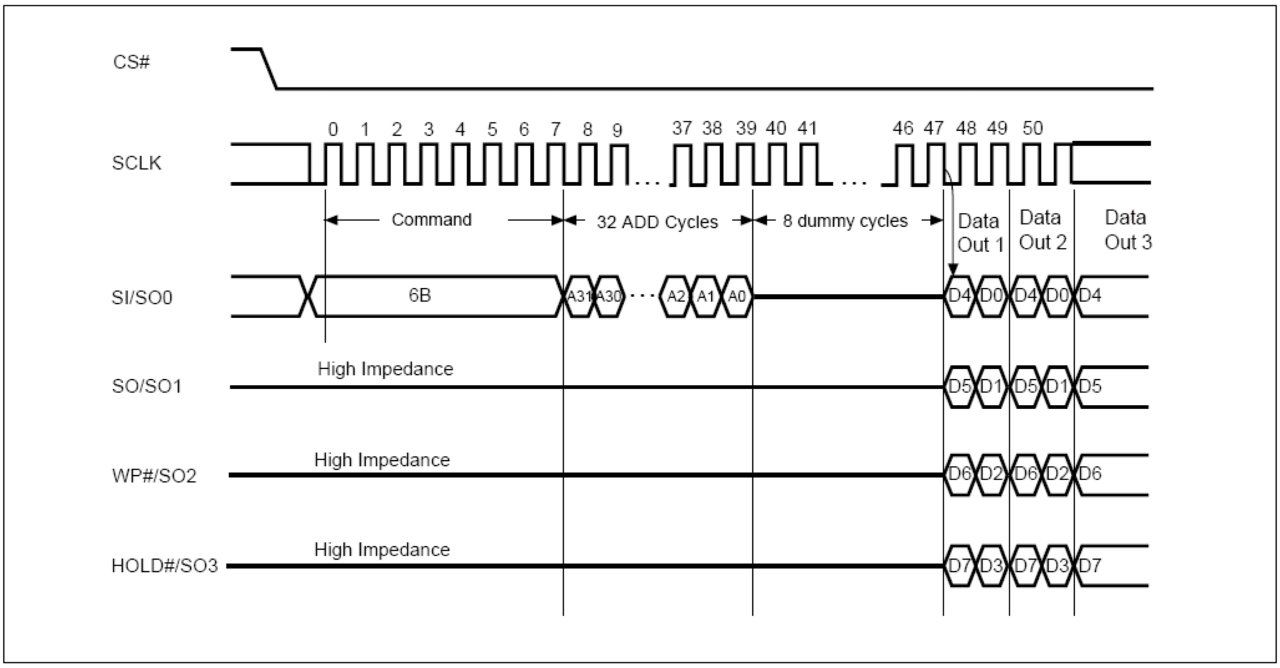
2) read data sequencye (1 I/O, command 03h, 4 byte address mode),



3) read data sequencye (2 I/O, command 3bh, 4 byte address mode),



4) read data sequencye (4 I/O, command 3bh, 4 byte address mode),



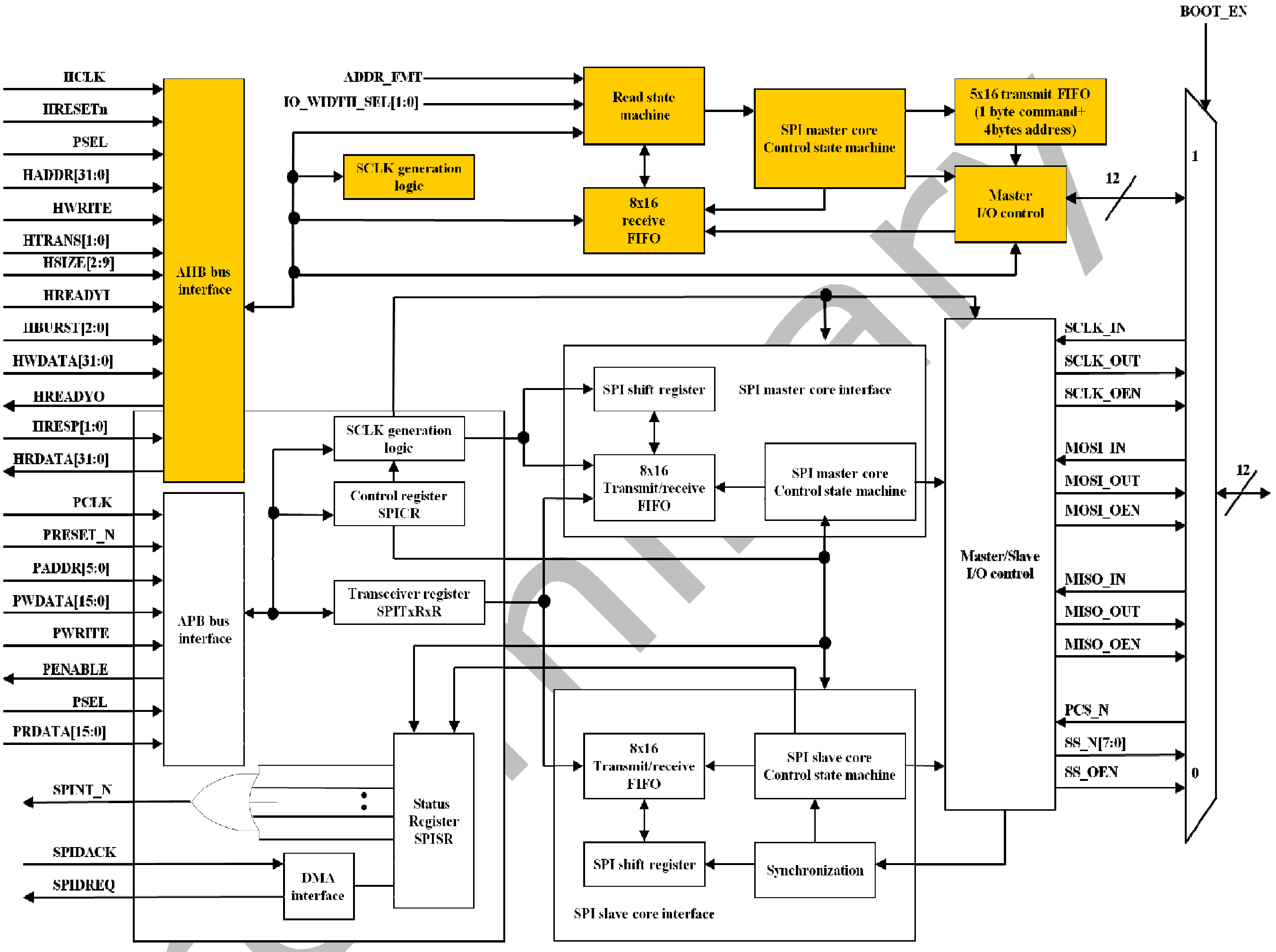
Interfaces

MNEMONIC	Type	Description
Booting control Signals		
io_width_sel[1:0]	In	Select 1/2/4 I/O 2'b00: 1 I/O 2'b01: 2 I/O 2'b10: 4 I/O 2'b11: reserved default: 2'b00
addr_fmt	In	1'b0: 3 bytes address mode 1'b1: direct 4 bytes address mode default: 1'b0
sio2_in	In	data bit 2 input for 4 I/O, this pin is connected to bit 2 bi-directional I/O cell when 4 I/O mode is enabled
sio3_in	In	data bit 3 input for 4 I/O, this pin is connected to bit 3 bi-directional I/O cell when 4 I/O mode is enabled
addr_en4b	In	1'b0: keep in 3 byte address mode 1'b1: enter 4 bytes address mode command(instruction :b7h) , only for Macronix default:1'b0
sio2_sio3_oe	Out	connect to output enable pin of bi-directional I/O cell for data bit 2, 3 when 4 I/O mode is enabled
sclk_divr_ahb[5:0]	In	SPI master controller clock divisor pins, the value of sclk_divr_ahb is used to generate the transmit and receive bit rate of the SPI master controller. The frequency divisor equation for generation sclk is below: sclk divisor = (sclk_divr_ahb[5:3] + 1) x $2^{(\text{sclk\_divr\_ahb}[2:0] + 1)}$ sclk = hclk/sclk divisor
AHB Bus Master Port1 Signals (For boot function, read data from SPI serial NOR flash port)		
hclk	In	AHB clock
hreset_n	In	AHB low active reset
hsel	In	AHB device selection
haddr[31:0]	In	AHB bus address
hwrite	In	AHB transfer direction
htrans[1:0]	In	AHB transfer type
hsize[2:0]	In	AHB transfer size
hreadyi	In	AHB transfer ready input
hburst[2:0]	In	AHB burst type
hwdata[31:0]	In	AHB write data bus
hreadyo	Out	AHB transfer ready output
hresp[1:0]	Out	AHB slave transfer response

hrdata[31:0]	Out	AHB read data bus
APB Bus Master Port2 Signals (Legacy SPI control and read write data port)		
pclk	In	APB clock
preset_n	In	APB low active reset
paddr[5:0]	In	APB bus address
pwdata[15:0]	In	APB write data bus
pwrite	In	APB bus write or read transfer indicator
penable	In	APB bus device enable
psel	In	APB bus device selection
prdata[15:0]	Out	APB bus read data
SPI Bus Interface Signals		
miso_in	In	MISO input
miso_out	Out	MISO output
miso_oen	Out	MISO output enable
mosi_in	In	MOSI input
mosi_out	Out	MOSI output
mosi_oen	Out	MOSI output enable
sclk_in	In	SPI clock input
sclk_out	Out	SPI clock output
sclk_oen	Out	SPI clock output enable
ss_n[7:0]	Out	SPI slave select
pcs_n	In	chip select
ss_oen	Out	slave output enable
DMA Interface Signals		
spidack0	In	DMA channel 0 acknowledge
spireq0	Out	DMA channel 0 request
spidack1	In	DMA channel 1 acknowledge
spireq1	Out	DMA channel 1 request
spi_int	Out	SPI interrupt
DFT Signals		
dftmode	In	DFT enable

Architecture

Block Diagram



Block Descriptions

The Serial Peripheral Interface (SPI) controller with booting function is divieded into two parts:
1) General SPI controller via APB interface, 2) Serial NOR flash controller via AHB interface.
The SPI master/slave controller consists of an APB Slave interface for control and transceiver register setting, a SPI master controller interface that generates SPI signals for peripherals, and a SPI slave controller interface that includes a synchronizer to sync the sclk clock. The transmit/receive FIFO is used to buffer the data between the APB slave interface and the SPI master/slave controller.

The serial NOR flash controller via AHB interface is also depicted in the yellow-shaded part of the above figure. The controller consists of TX FIFO to send the serial NOR flash command and RX FIFO to receive the read data from serial NOR flash.

Registers

Register Summary

Name	Offset	Size	Reset Value	Description
SPI_TxR	0x0000	W	0x00000000	SPI master/slave controller transmit FIFO input
SPI_RxR	0x0000	W	0x00000000	SPI master/slave controller receiver FIFO output
SPI_IER	0x0004	W	0x00000000	Enable/Mask interrupts generated by the SPI master controller
SPI_FCR	0x0008	W	0x00000000	SPI master/slave controller FIFO control register
SPI_FWCR	0x000C	W	0x00000100	SPI master/slave controller transaction flow control register
SPI_DLYCR	0x0010	W	0x00000000	SPI master controller delay control register (master only)
SPI_TxCR	0x0014	W	0x00000000	Transmit counter (master only)
SPI_RxCR	0x0018	W	0x00000000	Receive counter (master only)
SPI_SSCR	0x001C	W	0x00000000	SPI master/slave controller slave select and characteristic register
SPI_ISR	0x0020	W	0x00000000	SPI master/slave controller interrupt status register
SPI_FIFO_STAT	0x0024	W	0x00000011	SPI Transmit FIFO and Receive FIFO status
SPI_TX_REG0	0x0028	W	0x00000000	SPI transmit register 0 for normal usage
SPI_TX_REG1	0x002C	W	0x00000000	SPI transmit register 1 for normal usage
SPI_TX_REG2	0x0030	W	0x00000000	SPI transmit register 2 for normal usage
SPI_TX_REG3	0x0034	W	0x00000000	SPI transmit register 3 for normal usage
SPI_RX_REG0	0x0038	W	0x00000000	SPI receive register 0 for normal

				usage
SPI_RX_REG1	0x003C	W	0x00000000	SPI receive register 1 for normal usage
SPI_RX_REG2	0x0040	W	0x00000000	SPI receive register 2 for normal usage
SPI_RX_REG3	0x0044	W	0x00000000	SPI receive register 3 for normal usage
SPI_BOOT_CR	0x0048	W	0x00000000	Bootable SPI control register

Notes:
Size: **B** – Byte (8 bits) access, **HW** – Half WORD (16 bits) access, **W** –WORD (32 bits) access

Detailed Register Description

SPI_TxR
Address: Operational Base + offset (0x0000)
This register contains data to be transmitted on the SPI master/slave controller bus on the MOSI pin.

Bit	Attr	Reset Value	Description
31:16	-	-	Reserved.
15:0	W	0x0	SPI master controller transmit data FIFO

SPI_RxR
Address: Operational Base + offset (0x0000)
This register contains the data received from the SPI master/slave controller bus on the MISO pin.

Bit	Attr	Reset Value	Description
31:16	-	-	Reserved.
15:0	R	0x0	SPI master controller receive data FIFO

SPI_IER
Address: Operational Base + offset (0x0004)
This register contains bits for controlling the interrupt generation of the SPI master/slave controller

Bit	Attr	Reset Value	Description
31:12	-	-	Reserved.
11	RW	0x0	Character length mismatch interrupt enable bit. (CHARLNIEN) (slave only) The bit enables character length mismatch interrupt. When finishing transfer, the received data that character length mismatch the setting of SPI_SSCR[14:11]. “1” Enabled. “0” Disabled.
10	RW	0x0	Rx register 3 data available interrupt enable bit.

			(RXREG3IEN) (slave only) The bit enables the Rx register 3 data available interrupt when one piece of data is available in Rx register 3. “1” Enabled. “0” Disabled.
9	RW	0x0	Rx register 2 data available interrupt enable bit. (RXREG2IEN) (slave only) The bit enables the Rx register 2 data available interrupt when one piece of data is available in Rx register 2. “1” Enabled. “0” Disabled.
8	RW	0x0	Rx register 1 data available interrupt enable bit. (RXREG1IEN) (slave only) The bit enables the Rx register 1 data available interrupt when one piece of data is available in Rx register 1. “1” Enabled. “0” Disabled.
7	RW	0x0	Rx register 0 data available interrupt enable bit. (RXREG0IEN) (Slave only) The bit enables the Rx register 0 data available interrupt when one piece of data is available in Rx register 0. “1” Enabled. “0” Disabled.
6	RW	0x0	Transmit FIFO empty interrupt enable bit. (TXFEIEN) The bit enables transmit FIFO interrupt when transmit FIFO is empty. “1” Enabled. “0” Disabled.
5	RW	0x0	Slave select signal rising detection enable bit. (SSNRIEN) (Slave only) The bit enables the slave select signal rising detection interrupt when the master set slave select bit changes from “0” to “1”. “1” Enabled. “0” Disabled.
4	RW	0x0	Receive data available enable bit. (RXAVIEN) The bit enables receive FIFO interrupt when at least one data available in receive FIFO. “1” Enabled. “0” Disabled.
3	RW	0x0	Transmit FIFO interrupt enable bit (TxFIEN). This bit enables the transmit FIFO interrupt when the transmit FIFO trigger level is reached. “1” Enabled. “0” Disabled.

2	RW	0x0	Receive FIFO interrupt enable bit (RxFIEN). This bit enables the receive FIFO interrupt when the receive FIFO trigger level is reached. “1” Enabled. “0” Disabled.
1	RW	0x0	Receive FIFO overrun interrupt enable bit (RxFOIEN). This bit enables the receive FIFO overrun interrupt when the receive FIFO overrun condition has occurred. “1” Enabled. “0” Disabled.
0	RW	0x0	*Receive transfer complete interrupt enable bit (RxCIEN). (master only)* *This bit enables the receive transfer complete interrupt each time a receive transaction has ended.* “1” Enabled. “0” Disabled.

SPI_FCR
Address: Operational Base + offset (0x0008)
The FCR allows selection of the FIFO trigger level (the number of entries in the receive FIFO required to enable the receive FIFO interrupt and the number of empty entries in the transmit FIFO required to enable the transmit FIFO interrupt). The FIFOs can also be cleared using this register.

Bit	Attr	Reset Value	Description
31:14	-	-	Reserved.
13:11	RW	0x0	Defines the receive FIFO interrupt trigger level. The receive FIFO interrupt trigger level meaning is described below. For example, trigger level 4 entries indicate that there are at 4 data available entries in the receive FIFO. 0x0 – 2 entries0x1 – 4 entries 0x2 – 6 entries 0x3~0x7 – Reserved.
10:8	RW	0x0	Define the transmit FIFO interrupt trigger level. The transmit FIFO interrupt trigger level meaning is described below. For example, trigger level 4 entries indicate that there are at least 4 empty location entries for pushing data into the transmit FIFO. 0x0 – 2 entries0x1 – 4 entries 0x2 – 6 entries 0x3~0x7 – Reserved.
7:4	-	-	Reserved.
3	W	0x0	SPI master/slave controller receive FIFO reset bit

			(CLRRXF_N). Writing a '1' to this bit resets the receive FIFO.
2	W	0x0	SPI master/slave controller transmit FIFO reset bit (CLRTXF_N). Writing a '1' to this bit resets the transmit FIFO.
1	R	0x0	Transmit FIFO full flag (TxFF). This bit is set whenever the transmit FIFO is full.
0	R	0x0	*Receive data available flag (RxDAF).* This bit is set whenever at least has one data entry is available in the receive FIFO.

Notes:

The transmit and receive FIFO reset signals sustain **3 pclk cycles.** During the reset period, any access to FIFO is ignored.

SPI_FWCR

Address: Operational Base + offset (0x000C)

This register is used to control SPI master/slave controller transaction flow.

Bit	Attr	Reset Value	Description
31:15	-	-	Reserved.
14	RW	0x0	Slave select signal control enable bit (SSC) (master only) 0: Slave select signal is controlled by HW 1: Slave select signal is controlled by SW
13	RW	0x1	Slave select signal active bit (SSA) (master R/W, and slave read only) Writing a '1' to this bit will set the slave select signal not active. The value of slave select signal changes from '0' to '1'. Writing a '0' to this bit will set the slave select signal active. The value of the slave select signal changes from '1' to '0'.
12	RW	0x0	SPI master/slave mode (SPIMD). This bit selects the SPI master/slave mode. "0" SPI is in Slave mode "1" SPI is in Master mode
11	W	0x0	SPI master/slave controller soft reset bit (SRST_N). Writing a '1' to this bit resets the SPI master/slave controller logic.
10	RW	0x0	SPI master/slave enable bit (SPIEN). This bit enables the SPI master controller. "1" SPI master/slave enabled "0" SPI master/slave disabled
9	RW	0x0	SPI run bit (SPIRUN). (Master only) When the CPU sets this bit from "0" to "1", the SPI master/slave controller begins to transfer the data stored in

			the transmit FIFO and/or receive the data into the receive FIFO on the SPI master/slave controller bus. This will automatically be cleared to “0” after the transaction has ended.
8	RW	0x1	SPI master controller clock idle enable bit (CKIDLEN). (Master only) This bit determines whether or not the SPI master controller clock can be asserted in an idle state during a transaction process, assuming that the master core meets the receive FIFO full or transmit FIFO empty conditions. “1” SPI master controller clock can be asserted in an idle state. “0” SPI master controller clock cannot be asserted in an idle state.
7	-	-	Reserved.
6	RW	0x0	Hardware DMA request enable
5	RW	0x0	Transmit and receive simultaneously transfer enable (TxRxsten) (master only) This bit indicates whether or not the transmit and receive transfer can occur concurrently during a transaction. “0” Tx and Rx cannot concurrently happen. ”1” Tx and Rx can concurrently happen.
4	RW	0x0	SPI master/slave controller clock polarity bit (CPOL). This bit determines the polarity of the SCLK. “0” SCLK is low when idle. “1” SCLK is high when idle.
3	RW	0x0	Clock Phase Bit (CPHA). This bit determines the clock phase of the SCLK in relationship to the serial data. "0" data is valid on the first SCLK edge (rising or falling) after slave select has asserted. "1" data is valid on the second SCLK edge (rising or falling) after slave select has asserted.
2	RW	0x0	LSB-First Enable (LSBEN). “1” LSB first transfer. “0” MSB first transfer.
1	-	-	Reserved.
0	RW	0x0	Loop back mode (LBKMD) (master only) This bit is indicates the operation mode of the SPI master controller is in a normal operation mode or in a loop back mode. ‘0’ Normal operation mode. ‘1’ Loop back mode.

Notes:

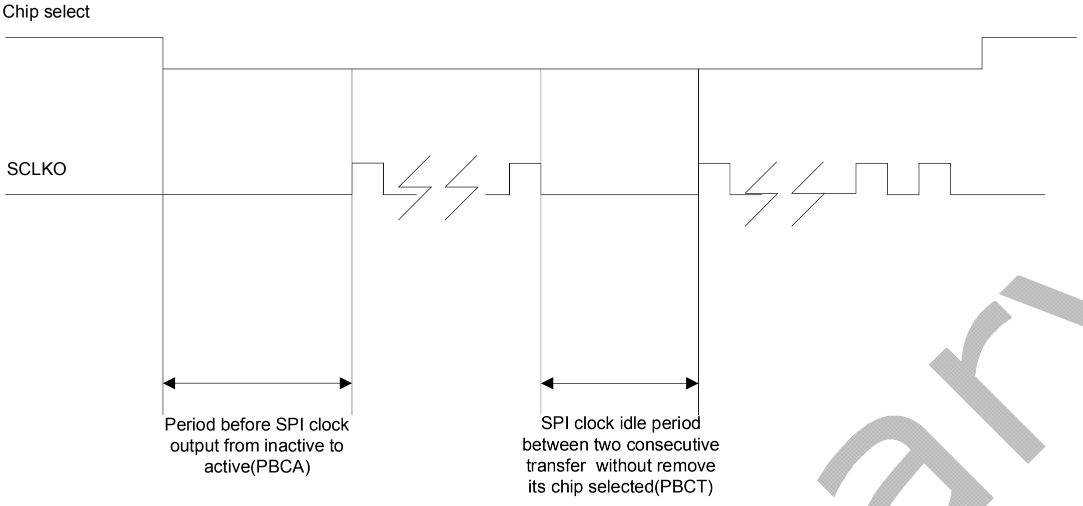
The soft reset signal can sustain **3 pclk cycles**. During the reset period, any access to FIFO is ignored.

SPI_DLYCR
Address: Operational Base + offset (0x0010)
This register sets the necessary clock delay during a transaction according to specific slave device requirements.

Bit	Attr	Reset Value	Description
31:11	-	-	Reserved.
10:8	RW	0x0	Period between Tx and Rx transfer (PBTxRx). This field defines the delay between transmit transfer complete and receive transfer start. 0x0: Non-SPI master controller clock delay. 0x1: 4 SPI master controller clock delay. 0x2: 8 SPI master controller clock delay. 0x3: 16 SPI master controller clock delay. 0x4: 32 SPI master controller clock delay. 0x5: 64SPI master controller clock delay. 0x6: 128 SPI master controller clock delay. 0x7: 256 SPI master controller clock delay.
7:6	-	-	Reserved.
5:3	RW	0x0	Period between two consecutive transfers (PBCT). This field defines the delay between consecutive transfers to the device without removing its chip select. 0x0: Non-SPI master controller clock delay. 0x1: 4 SPI master controller clock delay. 0x2: 8 SPI master controller clock delay. 0x3: 16 SPI master controller clock delay. 0x4: 32 SPI master controller clock delay. 0x5: 64 SPI master controller clock delay. 0x6: 128 SPI master controller clock delay. 0x7: 256 SPI master controller clock delay.
2:0	RW	0x0	Period before SPI master controller clock active (PBCA). This field defines the delay before the SPI master controller clock changes from idle to active after the chip select is asserted. 0x0: 1/2 SPI master controller clock delay. 0x1: 4 SPI master controller clock delay. 0x2: 8 SPI master controller clock delay. 0x3: 16 SPI master controller clock delay. 0x4: 32 SPI master controller clock delay. 0x5: 64SPI master controller clock delay. 0x6: 128 SPI master controller clock delay. 0x7: 256 SPI master controller clock delay.

Notes:

The timing diagram below illustrates the delay meaning.


Chip select
SCLKO
Period before SPI clock output from inactive to active(PBCA)
SPI clock idle period between two consecutive transfer without remove its chip selected(PBCT)

SPI_TxCR
Address: Operational Base + offset (0x0014)
This register controls the total transfer data size in each transmit transaction

Bit	Attr	Reset Value	Description
31:16	-	-	Reserved.
15:0	RW	0x0	0: Stop the transmit transaction. 1-65535: Start a transmit transaction.

SPI_RxCR
Address: Operational Base + offset (0x0018)
This register controls the total transfer data size in each receive transaction.

Bit	Attr	Reset Value	Description
31:16	-	-	Reserved.
15:0	RW	0x0	0: Stop the receive transaction. 1-65535: Start a receive transaction.

SPI_SSCR
Address: Operational Base + offset (0x001C)
SPI master/slave controller slaves select and characteristic control register.

Bit	Attr	Reset Value	Description
31:15	-	-	Reserved.
14:11	RW	0x0	Character length determines bits (SPICHRL). This field specifies how many bits are to be transferred in each transfer. 0x0: Reserved 0x1: Reserved 0x2: Reserved 0x3: 4 bits (master only)

			0x4: 5 bits (master only) 0x5: 6 bits (master only) 0x6: 7 bits (master only) 0x7: 8 bits 0x8: 9 bits 0x9: 10 bits 0xA: 11 bits 0xB: 12 bits 0xC: 13 bits 0xD: 14 bits 0xE: 15 bits 0xF: 16 bits
10:8	RW	0x0	SPI master controller slave select register (SPISSR) (master only) 0x0: Slave0 device (SPIx_SSN[0]) is set to the active state. 0x1: Slave1 device (SPIx_SSN[1]) is set to the active state. 0x2: Slave2 device (SPIx_SSN[2]) is set to the active state. 0x3: Slave3 device (SPIx_SSN[3]) is set to the active state. 0x4: Slave4 device (SPIx_SSN[4]) is set to the active state. 0x5: Slave5 device (SPIx_SSN[5]) is set to the active state. 0x6: Slave6 device (SPIx_SSN[6]) is set to the active state. 0x7: Slave7 device (SPIx_SSN[7]) is set to the active state.
7:6	-	-	Reserved.
5:0	RW	0x0	SPI master controller clock divisor bits (SPIDIVR) (master only) The value of SPIDIVR is used to generate the transmit and receive bit rate of the SPI master controller. The following section describes the bit rate equation in more detail. The frequency divisor equation for generation sclk is below For SPI Master: sclk Divisor = (SPIDIVR[5:3] + 1) x $2^{(SPIDIVR[2:0] + 1)}$ sclk = pclk/ sclk Divisor For SPI Slave: Slave sclk frequency limitation: Sclk = pclk/8. The frequency is the limit of the SPI hardware.

Notes:

CPHA, CPOL, should be set to match the protocol expected by the SPI slave device.

SPI_ISR

Address: Operational Base + offset (0x0020)

SPI master/slave controller interrupt status register.

Bit	Attr	Reset Value	Description

31:12	-	-	Reserved.
11	RW	0x0	Character length mismatch interrupt flag (CHARLNIF) (slave only). While finishing transfer, the received data that character length mismatch the setting of SPI_SSCR[14:11], the bit is set. An interrupt will be asserted to the CPU when this bit is set and the CHARLNIEN bit is enabled. The bit is cleared by writing “1” to the register.
10	RW	0x0	Rx register 3 data available interrupt flag (RXREG3IF) (slave only). This bit is set when data available in Rx register 3. An interrupt will be asserted to the CPU when this bit is set and the RXREG3IEN bit is enabled. The bit is cleared by writing “1” to the register.
9	RW	0x0	Rx register 2 data available interrupt flag (RXREG2IF) (slave only). This bit is set when data is available in Rx register 2. An interrupt will be asserted to the CPU when this bit is set and the RXREG2IEN bit is enabled. The bit is cleared by writing “1” to the register.
8	RW	0x0	Rx register 1 data available interrupt flag (RXREG1IF) (slave only). This bit is set when data is available in Rx register 1. An interrupt will be asserted to the CPU when this bit is set and the RXREG1IEN bit is enabled. The bit is cleared by writing “1” to the register.
7	RW	0x0	Rx register 0 data available interrupt flag (RXREG0IF) (slave only). This bit is set when data is available in Rx register 0. An interrupt will be asserted to the CPU when this bit is set and the RXREG0IEN bit is enabled. The bit is cleared by writing “1” to the register.
6	R	0x0	Transmit FIFO empty interrupt flag (TXFEIF). This bit is set when the transmit FIFO is empty. An interrupt will be asserted to the CPU when this bit is set and the TXFEIEN bit is enabled. This bit will be cleared when the transmit FIFO is not empty.
5	RW	0x0	Slave select signal rising interrupt flag (SSNRIF) (slave only). This bit is set when the slave select signal is asserted. An interrupt will be asserted to the CPU when this bit is set and the SSNRIEN bit is enabled. The bit is cleared by writing “1” to the register.
4	R	0x0	Receive data available interrupt flag (RXAVIF).

			This bit is set when at least one piece of data is available in receive FIFO. An interrupt will be asserted to the CPU when this bit is set and the RXAVIEN bit is enabled. This bit will be cleared when the receive FIFO is empty.
3	R	0x0	Transmit FIFO interrupt flag (TxFIF). This bit is set when the transmit FIFO trigger level is reached, and CPU wishes to keep transmit data to the device. An interrupt will be asserted to the CPU when this bit is set and the TxFIEN bit is enabled. This bit will be cleared when the transmit FIFO pointer drops below the trigger level.
2	R	0x0	Receive FIFO interrupt flag (RxFIF). This bit is set whenever the receive FIFO trigger level is reached. An interrupt will be asserted to the CPU when this bit is set and the RxFIEN bit is enabled. This bit will be cleared when the receive FIFO pointer drops below the trigger level.
1	R	0x0	SPI master/slave controller overrun interrupt flag (SPIORIF). '1' – If the receive FIFO is full and another character has been received in the receiver shift register. If another character is starting to arrive, it will overwrite the data in the shift register, but the FIFO will remain intact. The bit is cleared if the receive FIFO is cleared by software simultaneously. '0' – No overrun state an another SPI master controller transaction
0	R	0x0	Receive complete interrupt flag (RxCIF) (master only). This bit is set whenever the receive transaction is over. An interrupt will be asserted to the CPU when this bit is set and the RxCIEN bit is enabled. The bit is cleared upon reading from the register.

SPI_FIFO_STAT
Address: Operational Base + offset (0x0024)
This register contains the SPI Transmit FIFO and Receive FIFO status.

Bit	Attr	Reset Value	Description
31:7	-	-	Reserved.
6	R	0x0	*Receive FIFO full bit. (RxFF_FULL)* *This bit shows whether or not the Receive FIFO is full.* "1": Receive FIFO is full. "0": Receive FIFO is not full.
5	R	0x0	*Receive FIFO half full bit. (RxFF_HALFFULL)* *This bit shows whether or not the Receive FIFO is half*

			full. "1": There are four pieces of data in the Receive FIFO. "0": There are not four pieces of data in the Receive FIFO.
4	R	0x1	*Receive FIFO empty bit. (RxFF_EMPTY)* *This bit shows whether or not the Receive FIFO is empty.* "1": Receive FIFO is empty. "0": Receive FIFO is not empty.
3	-	-	Reserved.
2	R	0x0	*Transmit FIFO full bit. (TxFF_FULL)* *This bit shows whether or not the transmit FIFO is full.* "1": Transmit FIFO is full. "0": Transmit FIFO is not full.
1	R	0x0	*Transmit FIFO half full bit. (TxFF_HALFFULL)* *This bit shows whether or not the transmit FIFO is half full.* "1": There are four pieces of data in the Transmit FIFO. "0": There are not four pieces of data in the Transmit FIFO.
0	R	0x1	*Transmit FIFO empty bit. (TxFF_EMPTY)* *This bit shows whether or not the transmit FIFO is empty.* "1": Transmit FIFO is empty. "0": Transmit FIFO is not empty.

SPI_TX_REGx (x = 0, 1, 2, 3)
Address: Operational Base + offset (0x0028/0x002C/0x0030/0x0034)
This register contains data to be transmitted on the SPI controller bus on the MISO pin.

Bit	Attr	Reset Value	Description
31:16	-	-	Reserved.
15:0	RW	0x0	SPI master/slave controller transmit data register (slave only)

SPI_RX_REGx (x = 0, 1, 2, 3)
Address: Operational Base + offset (0x0038/0x003C/0x0040/0x0044)
This register contains the data received from the SPI controller bus on the MOSI pin.

Bit	Attr	Reset Value	Description
31:16	-	-	Reserved.
15:0	R	0x0	SPI master/slave controller receive data register (slave only)

Notes: Attr: **RW** – Read/writable, **R** – Read only, **W** – Write only

SPI_BOOT_CR	0x0048	W	0x00000000	Bootable SPI control register

SPI_BOOT_CR
Address: Operational Base + offset (0x0048)
This register contains the control bit to control SCLK (divided from HCLK) frequency and boot enable.

Bit	Attr	Reset Value	Description
31:7	-	-	Reserved.
6	RW	0x1	Boot enable 0x0:APB interface can access SPI, for legacy SPI 0x1: AHB interface can access SPI, support bootable function When we need to set register in legacy APB-interfaced SPI, this bit should be set to 0x0
5:0	RW	0x0	SPI master controller clock divisor bits (SPIDIVR) The value of SPIDIVR is used to generate the transmit and receive bit rate of the SPI master controller. The following section describes the bit rate equation in more detail. The frequency divisor equation for generation sclk is below: sclk Divisor = $(\mathrm{SPIDIVR}[5:3] + 1) \times 2^{(\mathrm{SPIDIVR}[2:0] + 1)}$ sclk = hclk/ sclk Divisor

Notes: Attr: **RW** – Read/writable, **R** – Read only, **W** – Write only

Functional Description

Operation

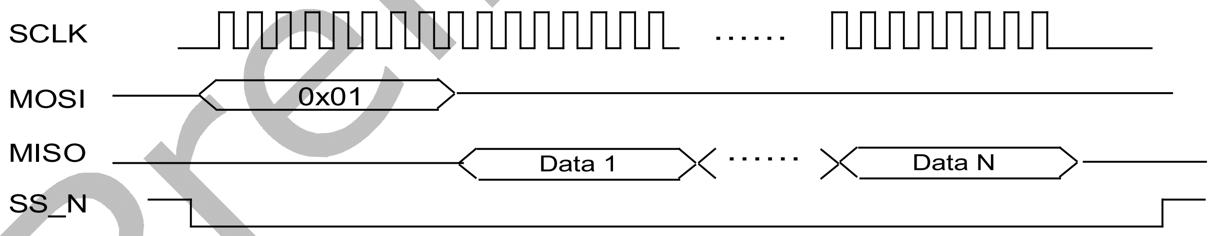
The SPI Slave receives HW commands from the Master before transferring data.
The following table contains a list of these instructions and their operation codes. All instructions start at a high-to-low pcs_n transition.

Command	Description
0x01	MISO data from Slave Tx FIFO
0x02	MISO data from Slave Tx/Rx FIFO status
0x03	MISO data from Slave Tx register 0
0x04	MISO data from Slave Tx register 1
0x05	MISO data from Slave Tx register 2
0x06	MISO data from Slave Tx register 3
0x08	MOSI data to Slave Rx FIFO
0x09	MOSI data to Slave Rx register 0
0x0A	MOSI data to Slave Rx register 1
0x0B	MOSI data to Slave Rx register 2
0x0C	MOSI data to Slave Rx register 3

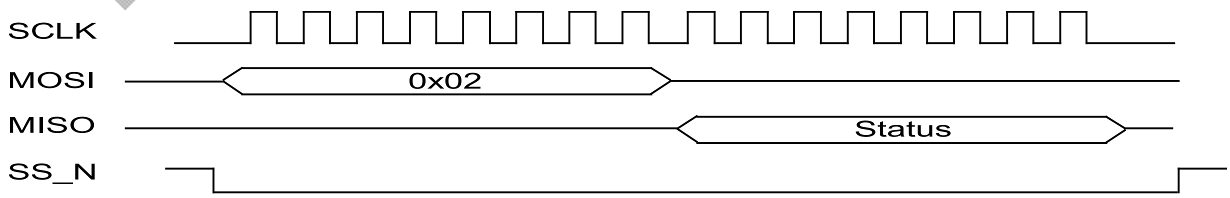
Note: The command length depends on SPI_SSCR[14:11]. The minimum size is 8 bits. For example, if the bit length is 16 bits, the master should transmit 0x0008 to the slave to write data to the slave Rx FIFO.

HW command timing

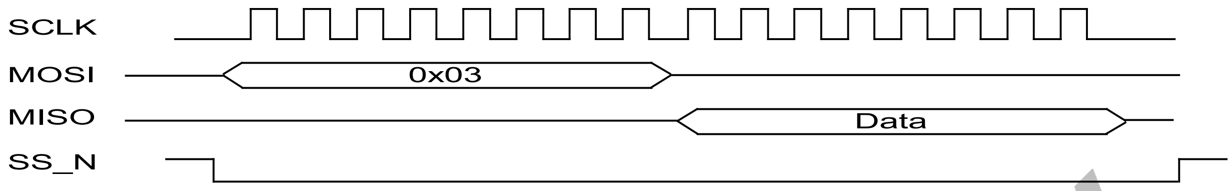
Command 0x01



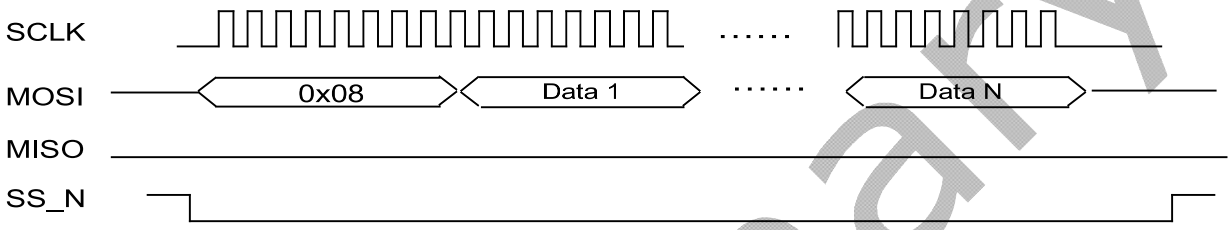
Command 0x02



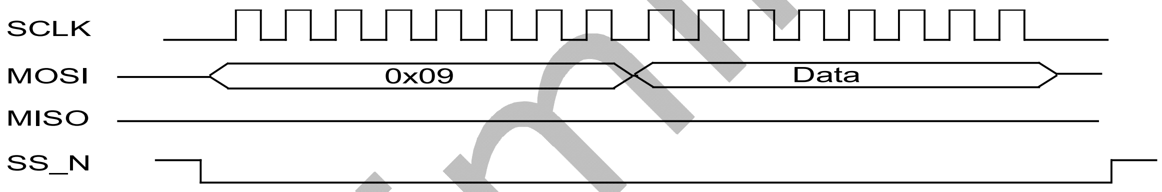
Command 0x03 ~ Command 0x06



Command 0x08

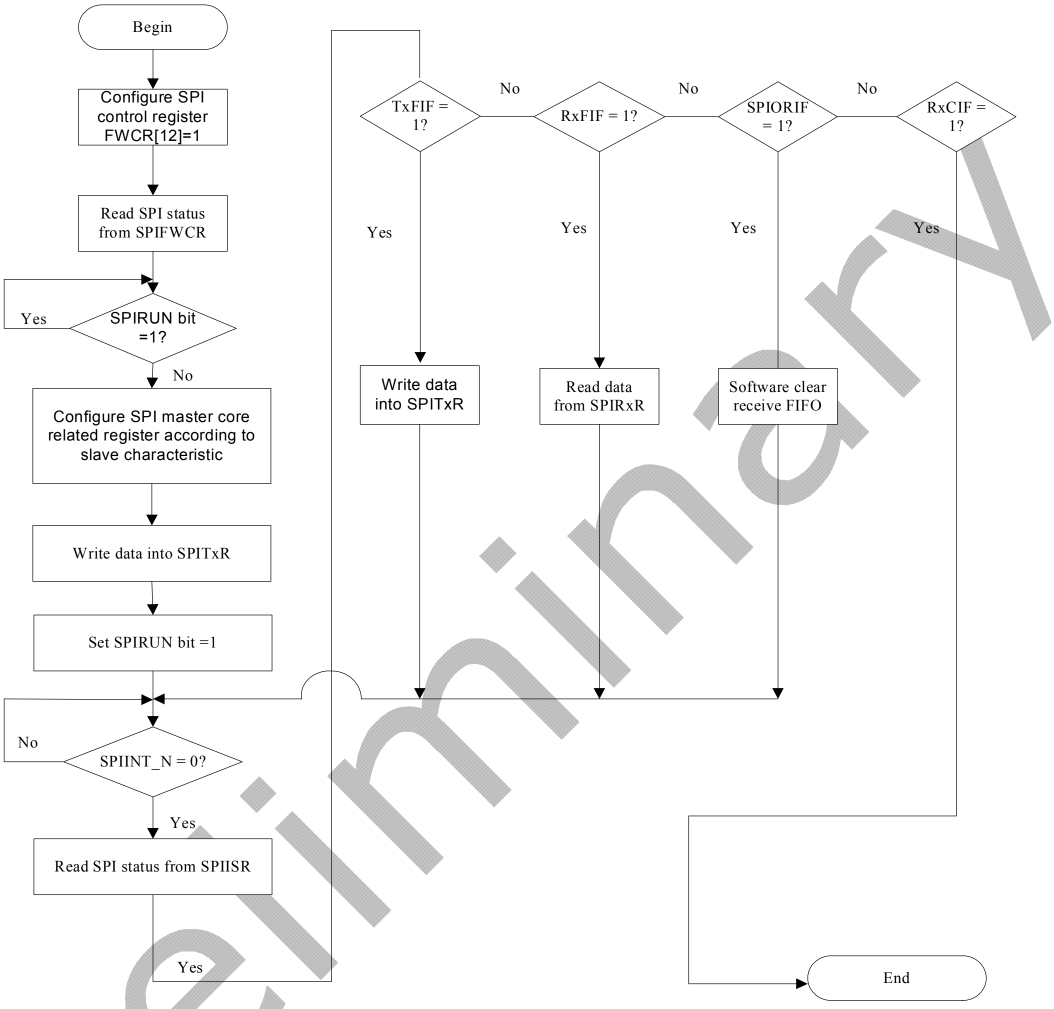


Command 0x09 ~ Command 0x0C

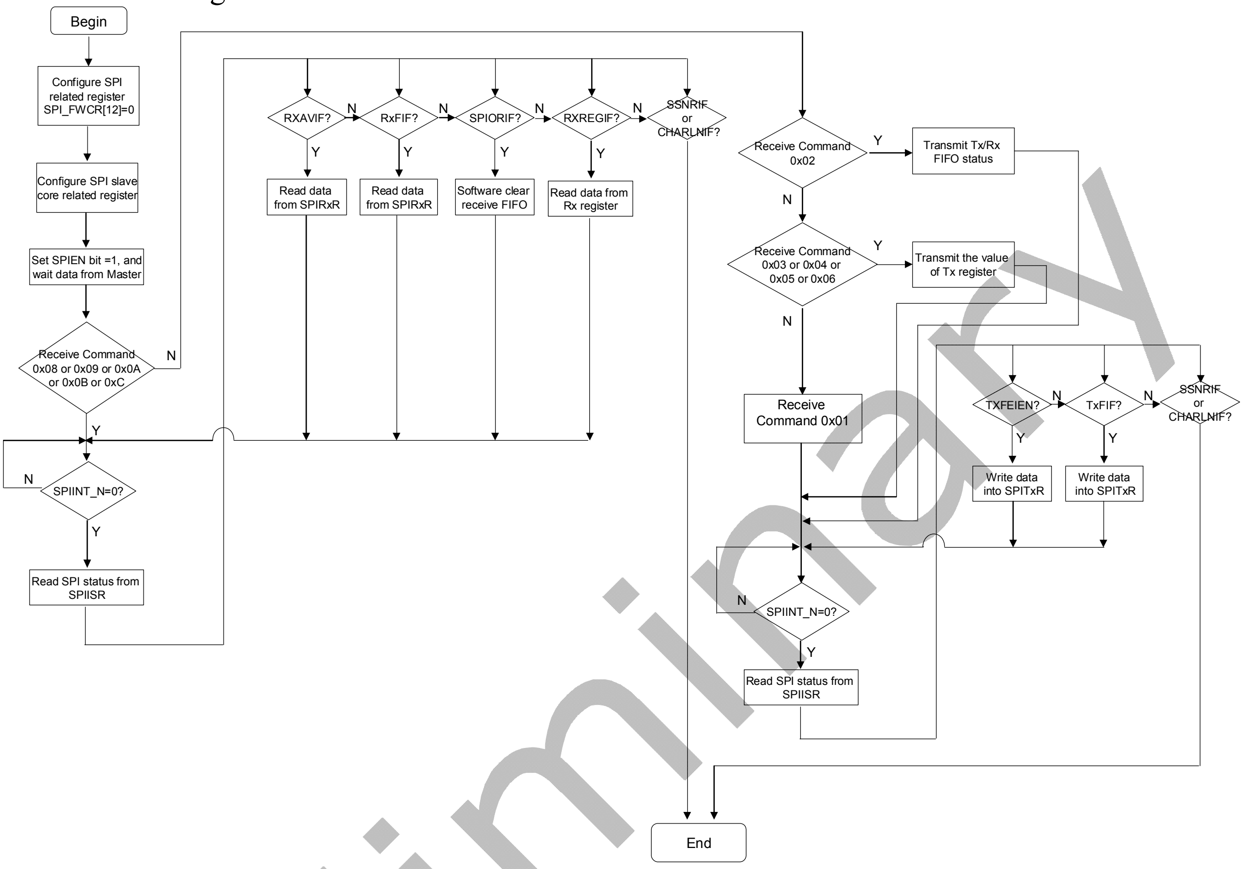


SPI master/slave controller operation flow chart

The flow chart below describes how the software configure and performs a SPI master controller transaction through the SPI master/slave controller.

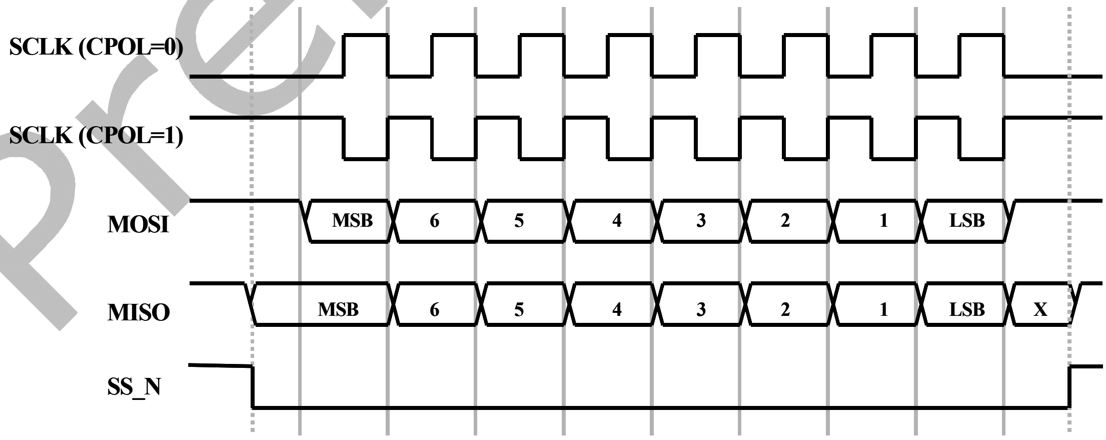

Begin
Configure SPI control register FWCR[12]=1
Read SPI status from SPIFWCR
Yes
SPIRUN bit =1?
No
Configure SPI master core related register according to slave characteristic
Write data into SPITxR
Set SPIRUN bit =1
No
SPIINT_N = 0?
Yes
Read SPI status from SPIISR
Yes
TxFIF = 1?
No
RxFIF = 1?
No
SPIORIF = 1?
No
RxCIF = 1?
Yes
Yes
Yes
Yes
Write data into SPITxR
Read data from SPIRxR
Software clear receive FIFO
End

The flow chart below describes how the software configures and performs a SPI slave controller transaction through the SPI master/slave controller.

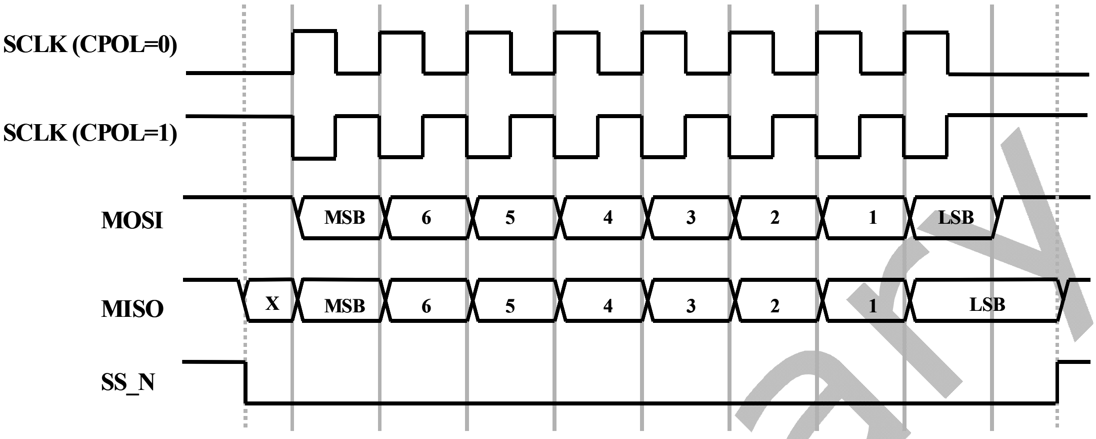

Begin
Configure SPI related register SPI_FWCR[12]=0
Configure SPI slave core related register
Set SPIEN bit =1, and wait data from Master
Receive Command 0x08 or 0x09 or 0x0A or 0x0B or 0xC
N
Y
SPIINT_N=0?
N
Y
Read SPI status from SPIISR
RXAVIF?
N
Y
Read data from SPIRxR
RxFIF?
N
Y
Read data from SPIRxR
SPIORIF?
N
Y
Software clear receive FIFO
RXREGIF?
N
Y
Read data from Rx register
SSNRIF or CHARLNIF?
Receive Command 0x02
Y
Transmit Tx/Rx FIFO status
N
Receive Command 0x03 or 0x04 or 0x05 or 0x06
Y
Transmit the value of Tx register
N
Receive Command 0x01
TXFEIEN?
N
Y
Write data into SPITxR
TxFIF?
N
Y
Write data into SPITxR
SSNRIF or CHARLNIF?
SPIINT_N=0?
N
Y
Read SPI status from SPIISR
End

SPI master/slave controller data transfer waveform

CPHA=0 transfer waveform



CPHA=1 transfer waveform


SCLK (CPOL=0)
SCLK (CPOL=1)
MOSI
MSB
6
5
4
3
2
1
LSB
MISO
X
MSB
6
5
4
3
2
1
LSB
SS_N