Registers

1. Core Registers list

Name	Address	Width	Access	Description
Rx0	0x00	32	R	Data receive register 0
Rx1	0x04	32	R	Data receive register 1
Rx2	0x08	32	R	Data receive register 2
Rx3	0x0c	32	R	Data receive register 3
Tx0	0x00	32	R/W	Data transmit register 0
Tx1	0x04	32	R/W	Data transmit register 1
Tx2	0x08	32	R/W	Data transmit register 2
Tx3	0x0c	32	R/W	Data transmit register 3
CTRL	0x10	32	R/W	Control and status register
DIVIDER	0x14	32	R/W	Clock divider register
SS	0x18	32	R/W	Slave select register

Table 1: List of core registers

All registers are 32-bit wide and accessible only with 32 bits (all wb_sel_i signals must be active).

2. Data receive registers[RxX]

Bit #	31:0
Access	R
Name	Rx

Table 2: Data Receive register

Reset Value: 0x00000000

RxX

The Data Receive registers hold the value of received data of the last executed transfer. Valid bits depend on the character length field in the CTRL register (i.e. if CTRL[9:3] is set to 0x08, bit RxL[7:0] holds the received data). If character length is less or equal to 32 bits, Rx1,Rx2 and Rx3 are not used, if character length is less than 64 bits, Rx2 and Rx3 are not used and so on.

NOTE: The Data Received registers are read-only registers. A Write to these registers will actually modify the Transmit registers because those registers share the same FFs.

3. Data transmit register [TxX]

Bit #	31:0
Access	R/W
Name	Tx

Table 3: Data Transmit register

Reset Value: 0x00000000

TxX

The Data Receive registers hold the data to be transmitted in the next transfer. Valid bits depend on the character length field in the CTRL register (i.e. if CTRL[9:3] is set to 0x08, the bit Tx0[7:0] will be transmitted in next transfer). If character length is less or equal to 32 bits, Tx1, Tx2 and Tx3 are not used, if character len is less than 64 bits, Tx2 and Tx3 are not used and so on.

4. Control and status register [CTRL]

Bit #	31:14	13	12	11	10	9	8	7	6:0
Access	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W
Name	Reserved	ASS	IE	LSB	Tx_NEG	Rx_NEG	GO_BSY	Reserved	CHAR_LEN

Table 4: Control and Status register

Reset Value: 0x00000000

ASS

If this bit is set, ss_pad_o signals are generated automatically. This means that slave select signal, which is selected in SS register is asserted by the SPI controller, when transfer is started by setting CTRL[GO_BSY] and is de-asserted after transfer is finished. If this bit is cleared, slave select signals are asserted and de-aserted by writing and clearing bits in SS register.

IE

If this bit is set, the interrupt output is set active after a transfer is finished. The Interrupt signal is deasserted after a Read or Write to any register.

LSB

If this bit is set, the LSB is sent first on the line (bit TxL[0]), and the first bit received from the line will be put in the LSB position in the Rx register (bit RxL[0]). If this bit is cleared, the MSB is transmitted/received first (which bit in TxX/RxX register that is depends on the CHAR_LEN field in the CTRL register).

Tx_NEG

If this bit is set, the mosi_pad_o signal is changed on the falling edge of a sclk_pad_o clock signal, or otherwise the mosi_pad_o signal is changed on the rising edge of sclk_pad_o.

Rx_NEG

If this bit is set, the miso_pad_i signal is latched on the falling edge of a sclk_pad_o clock signal, or otherwise the miso_pad_i signal is latched on the rising edge of sclk_pad_o.

GO_BSY
Writing 1 to this bit starts the transfer. This bit remains set during the transfer and is automatically cleared after the transfer finished. Writing 0 to this bit has no effect.

NOTE: All registers, including the CTRL register, should be set before writing 1 to the GO_BSY bit in the CTRL register. The configuration in the CTRL register must be changed with the GO_BSY bit cleared, i.e. two Writes to the CTRL register must be executed when changing the configuration and performing the next transfer, firstly with the GO_BSY bit cleared and secondly with GO_BSY bit set to start the transfer.
When a transfer is in progress, writing to any register of the SPI Master core has no effect.

CHAR_LEN
This field specifies how many bits are transmitted in one transfer. Up to 64 bits can be transmitted.
CHAR_LEN = 0x01 … 1 bit
CHAR_LEN = 0x02 … 2 bits
…
CHAR_LEN = 0x7f … 127 bits
CHAR_LEN = 0x00 … 128 bits

5. Divider register [DIVIDER]

Bit #	31:16	15:0
Access	R	R/W
Name	Reserved	DIVIDER

Table 5: Divider register

Reset Value: 0x0000ffff

DIVIDER
The value in this field is the frequency divider of the system clock wb_clk_i to generate the serial clock on the output sclk_pad_o. The desired frequency is obtained according to the following equation:

$$f_{sclk} = \frac{f_{wb\_clk}}{(DIVIDER + 1) * 2}$$

6. Slave select register [SS]

Bit #	31:8	7:0
Access	R	R/W
Name	Reserved	SS

Table 6: Slave Select register

Reset Value: 0x00000000

SS

If CTRL[ASS] bit is cleared, writing 1 to any bit location of this field sets the proper ss_pad_o line to an active state and writing 0 sets the line back to inactive state. If CTRL[ASS] bit is set, writing 1 to any bit location of this field will select appropriate ss_pad_o line to be automatically driven to active state for the duration of the transfer, and will be driven to inactive state for the rest of the time.