



eSi-SPI

1 Contents

1	Contents	2
2	Overview	3
3	Hardware Interface	4
4	Software Interface	5
4.1	Register Map	5
4.2	Interrupts	6
5	Revision History	8

2 Overview

The eSi-SPI core can be used to implement full-duplex, synchronous, serial communications. It supports the following features:

- Master or slave operation.
- Programmable word size (8 or 16-bits).
- Programmable bit ordering (MSB first / LSB first).
- Programmable clock polarity and phase.
- Programmable bit rate.
- AMBA 3 APB slave interface.
- DMA flow-control interface.

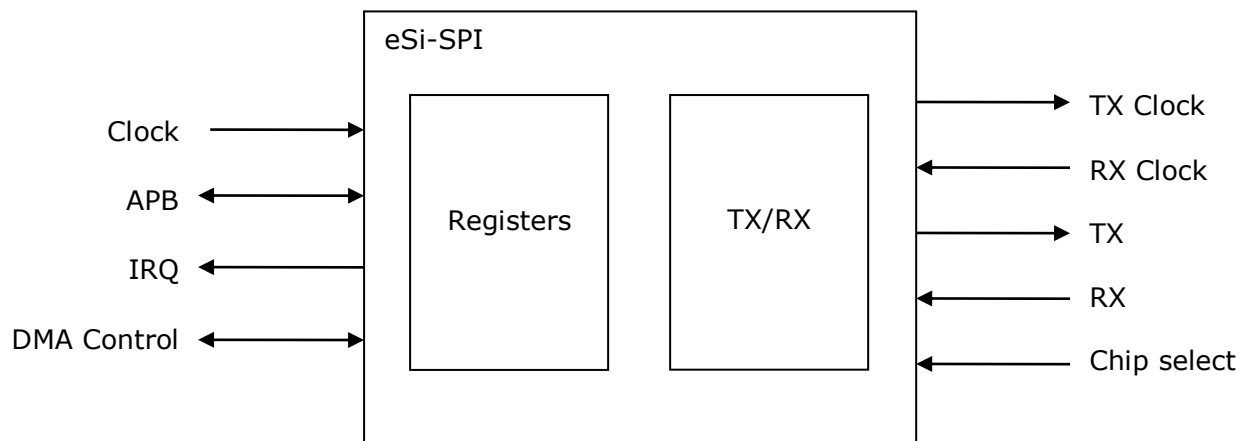


Figure 1: eSi-SPI

3 Hardware Interface

Module Name	cpu_apb_spi
HDL	Verilog
Technology	Generic
Source Files	cpu_apb_spi.v, cpu_peripheral_flow_control.v

Port	Direction	Width	Description
clk	Input	1	Clock used for serial data transmission and reception. This clock must be enabled when <code>cactive</code> is asserted. This clock will be divided by the value in the <code>cycles_per_bit</code> register.
pclk	Input	1	APB clock
presetn	Input	1	APB reset, active-low
paddr	Input	8	APB address
pselect	Input	1	APB slave select
penable	Input	1	APB enable
pwrite	Input	1	APB write
pwrite_data	Input	16	APB write data
rx_clk	Input	1	Receive clock. Only used in slave mode
rx_cs_n	Input	1	Receive chip select, active-low. Only used in slave mode
rx	Input	1	Receive data
tx_ack	Input	1	Acknowledges <code>tx_ready</code> after transfer complete
rx_ack	Input	1	Acknowledges <code>rx_ready</code> after transfer complete
cactive	Output	1	Clock active
pready	Output	1	APB ready
prdata	Output	16	APB read data
pslverr	Output	1	APB slave error
tx_clk	Output	1	Transmit clock. Only valid in master mode
tx	Output	1	Transmit data
interrupt_n	Output	1	Interrupt request, active-low
tx_ready	Output	1	Indicates device can accept new data
rx_ready	Output	1	Indicates device has data to be read

Table 1: I/O Ports

For complete details of the APB signals, please refer to the AMBA 3 APB Protocol v1.0 Specification available at <http://www.arm.com/products/solutions/AMBAHomePage.html>

The SPI does not include internal synchronizing flip-flops. These should be implemented externally for the `rx`, `rx_cs_n` and `rx_clk` ports if the transmitting clock domain is asynchronous to `clk`.

4 Software Interface

4.1 Register Map

Register	Address offset	Access	Description
tx_data	0x00	W	Transmit register
rx_data	0x04	R	Receive register
status	0x08	R/W	Status register
control	0x0c	R/W	Control register
cycles_per_bit	0x10	R/W	Cycles per bit register

Table 2: Register Map

4.1.1 Transmit Data Register

Data to be transmitted over the serial interface should be written to the transmit register. The transmit data register should not be written to while the TXF bit in the status register is set, otherwise data loss may occur. When the WS field in the control register is set to 0, only the lower 8-bits of data written to the register are transmitted.

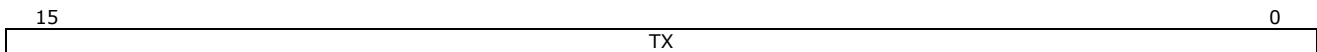


Figure 2: Format of the tx_data register

4.1.2 Receive Data Register

Data that is received over the serial interface can be read in the receive register. When the WS field in the control register is set to 0, only the lower 8-bits of the register contain valid data.

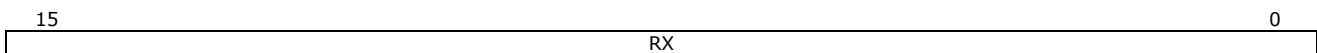


Figure 3: Format of the rx_data register

4.1.3 Status Register

The status register contains a selection of flags that indicate the current status of the SPI. To clear a bit in the status register, write a 1 to it. Writing 0 will leave it unchanged.

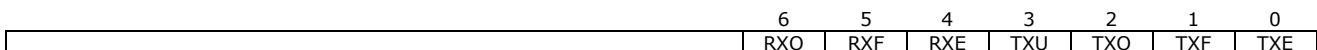


Figure 4: Format of the status register

Register	Values	Description
TXE	0 - Not empty 1 - Empty	Transmit buffer empty
TXF	0 - Not full 1 - Full	Transmit buffer full
TXO	0 - No overflow 1 - Overflow	Transmit buffer overflow
TXU	0 - No underrun 1 - Underrun	Transmit buffer underrun

RXE	0 - Not empty 1 - Empty	Receive buffer empty
RXF	0 - Not full 1 - Full	Receive buffer full
RXO	0 - No overflow 1 - Overflow	Receive buffer overflow

Table 3: Fields of the `status` register

4.1.4 Control Register

The control register contains a selection of flags that control the operation of the SPI.

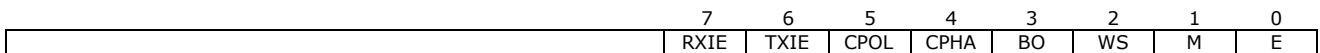


Figure 5: Format of the `control` register

Register	Values	Description
E	0 - Disabled 1 - Enabled	Enables the SPI. When disabled, data will not be received or transmitted
M	0 - Master 1 - Slave	Operating mode
WS	0 - 8-bits 1 - 16-bits	Word size
BO	0 - MSB first 1 - LSB first	Bit ordering
CPHA	0 - Sample on leading edge 1 - Sample of trailing edge	Clock phase
CPOL	0 - Idle low 1 - Idle high	Clock polarity
TXIE	0 - Disabled 1 - Enabled	Transmit interrupt enable
RXIE	0 - Disabled 1 - Enabled	Receive interrupt enable

Table 4: Fields of the `control` register

4.1.5 Cycles Per Bit Register

The cycles per bit register is a 16-bit that specifies how many cycles of the clock, `clk`, each bit is transmitted for in master mode. Use of a 16-bit register provides support for a wide range of clock frequencies and bit rates.

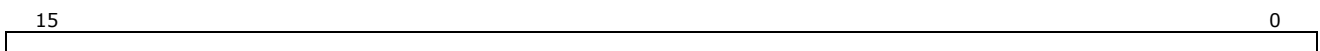


Figure 6: Format of the `cycles_per_bit` register

4.2 Interrupts

The SPI supports the following interrupts.

- Transmit interrupt

- Receive interrupt

The transmit interrupt will be raised when the transmit buffer is empty and the `TXIE` flag in the `control` register is set to 1. This indicates that the transmitter has no data to transmit.

The receive interrupt will be raised when the receiver buffer is not empty and the `RXIE` flag in the control register is set to 1. This indicates that the receiver has received some data.

5 Revision History

Hardware Revision	Software Release	Description
1	1.0.0	Initial release
2	2.3.2	Added <code>tx_ack</code> and <code>rx_ack</code> ports.

Table 5: Revision History