ABSTRACT

Implementation of the communication protocols SPI using a FPGA by the HDL-Verilog language

Overview:

Communication has a crucial role in human civilization. Nonetheless, new methods and ways were required to enhance the range of communication. numerous protocols into existence to meet the demands like I2C, AMBA, UART, SPI, etc
The objective is to design and implement the SPI communication protocol module.
The Serial Peripheral Interface module permits synchronous, full-duplex serial communication between the microcontroller unit and peripheral devices. The device that generates the clock signal is named the master. Data transmitted between the master and therefore the slave is synchronized to the clock generated by the master.

Relevance:

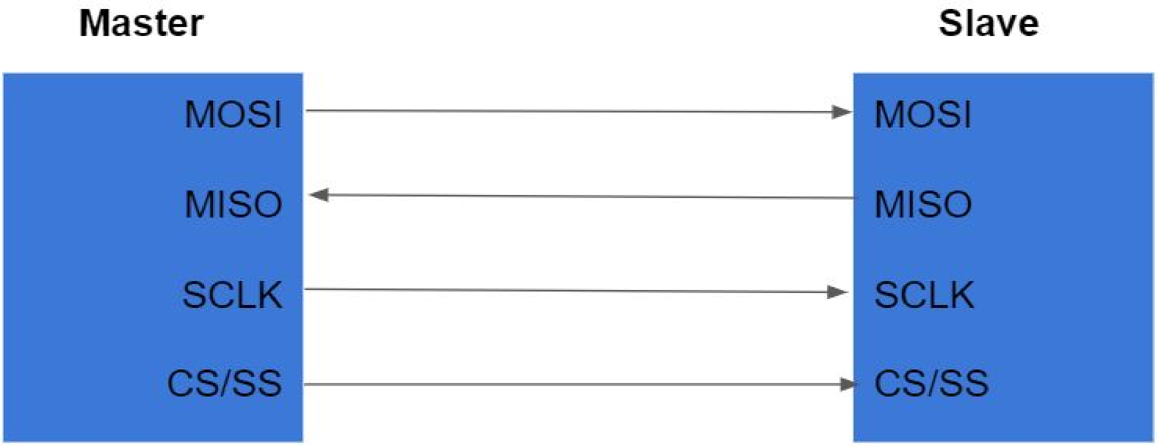
Examples include initiating an ADC conversion, addressing the proper page of non-volatile storage , and processing enough of a command that device firmware can load the primary word of the response. (Many SPI masters do not support that signal directly, and instead rely on fixed delays.)Serial Peripheral Interface is commonly used to send data between microcontrollers and small peripherals like shift registers, sensors, and SD cards

Literature Survey:

S.NO	Year	Author	Description
1	2017	Anusha	On the other hand, it is also a single-master communication protocol in which only one master can exist in the connection at a time to initiate all the communications with slaves.
2	2014	Manish	The focus of the paper used 2 parameters are clock polarity and clock phases using one master and multiple slaves.
3	2009	F.LEE	The focus of this paper is to study the spi protocol used for shift registers.(using verilog and hdl language)fpga.

Architecture and Algorithm and working principle:

SPI is calle a 4-wire bus as it requires four wires for its communication as shown below. In the case of single slave communications, we need only 3 wires, as slave select (SS) is not required. So, SPI requires more communication lines in contrast to UART, I^{2}C, USB etc.



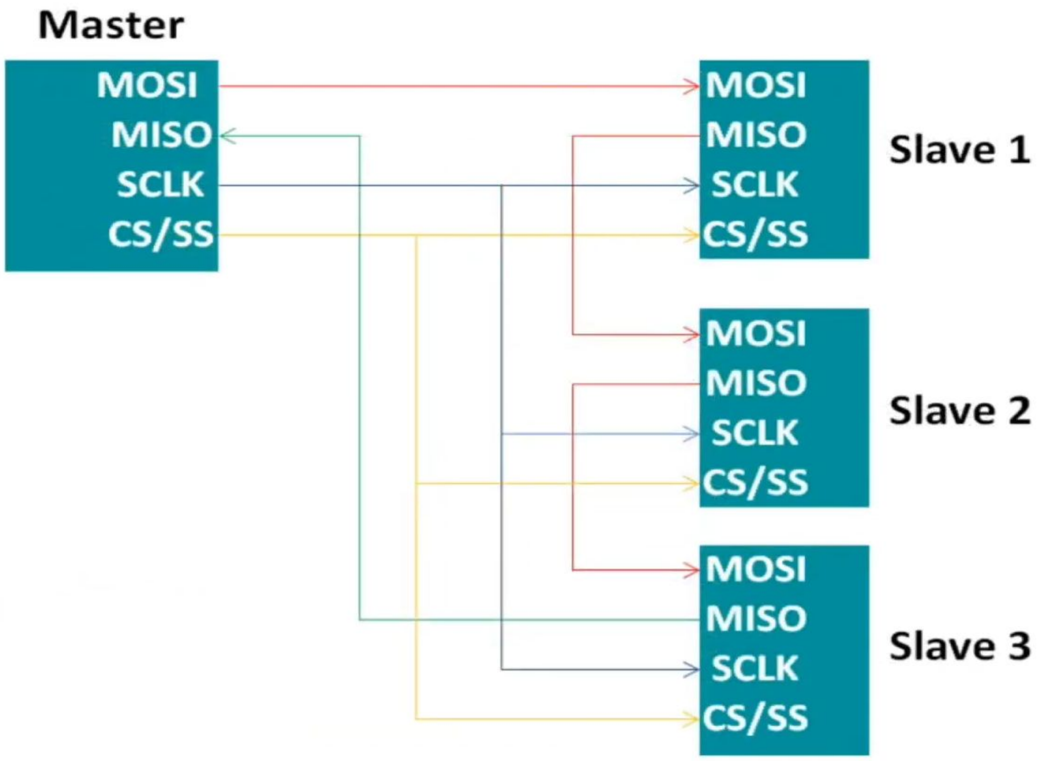
MISO: MISO stands for Master Input Slave Output. It is used to send data from the slave to the master.

MOSI: MOSI stands for Master Output Slave Input. It is used to send data from the master to the slave.

SCK or SCLK (Serial Clock): It is used to generate the clock signal.

SS/CS (Slave Select / Chip Select): It is used by the master to send data by selecting a slave.

SPI Master With multiple slave select:



Master will select only one slave at a time. Mostly slave devices will be equipped with tri-state outputs. So when they are not selected, their output lines appear disconnected.

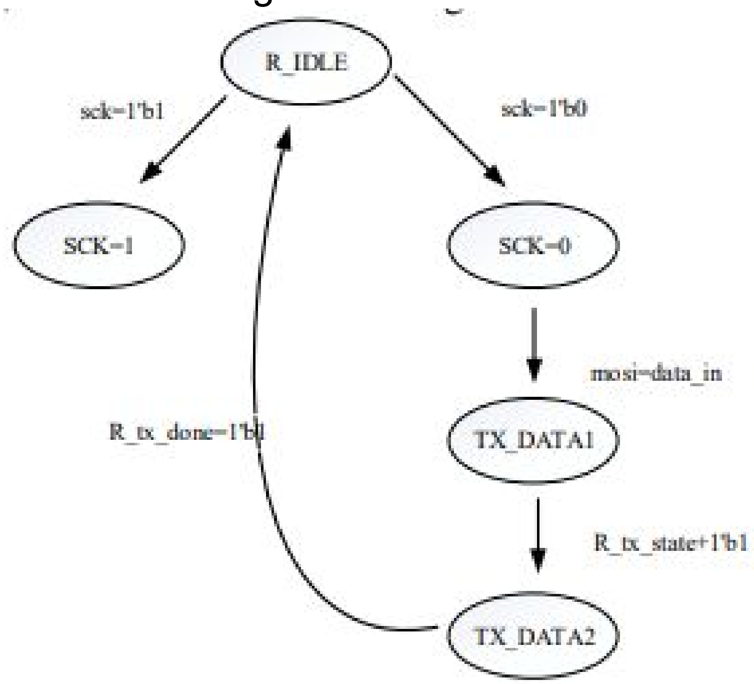
Mechanism (SPI Modes):

Clock Polarity: CPOL or CKP
Clock polarity is the idle / active state of the clock. If idle state is 0, active state will be 1 and vice versa.

Clock Phases: CPHASE, Inverted Clock Phase (Clock Edge) : NCPHA or CKE. Clock phase or clock edge defines when to transfer data. Data can be transferred during LOW (0) to HIGH (1) or HIGH to LOW transitions. If CPHASE is zero, data will be sampled at the rising edge of the clock and if CPHASE is one,data will be sampled at the falling edge of the clock

Methodology:

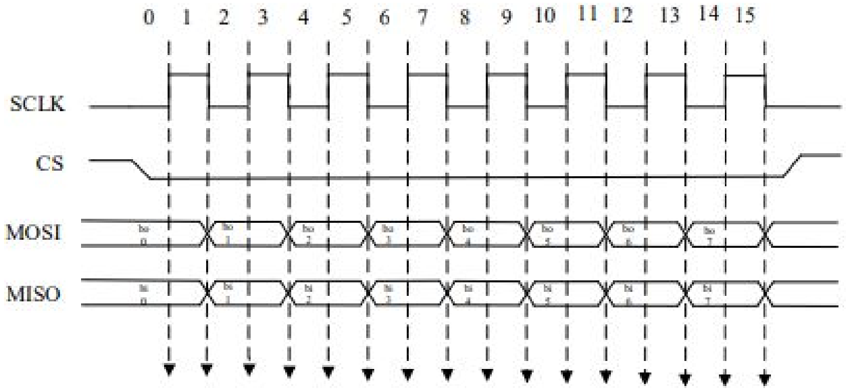
When the FPGA sends a byte (8-bit) data to the slave through the SPI bus, the FPGA first
sets the CS/SS chip select signal to 0, indicating that it is ready to start transmitting data.



This Idea is better than Previous communication protocols such as i2c and UART because, SPI works in :

- There is no start and stop bits, so the data can be streamed continuously without interruption.
- It supports full-duplex.
- No need for precision oscillators in slave devices as it uses a master's clock.
- Higher data transfer rate than I2C (almost twice as fast).No speed limit (practically it will be limited by the clock frequency, rise time, fall time etc.)
- Separate MISO and MOSI lines, so data can be sent and received at the same time.
- Simple software implementation.
- No speed limit (practically it will be limited by the clock frequency, rise time, fall time etc.)
- Not Limited to 8 bit data
- Signals are unidirectional through all lines, makes easy isolation
- No complicated slave addressing system like I2C.No need of a unique address in slaves like in RS485 or I2C.
- No need of precision oscillators in slave devices as it uses master's clock
- No complex transceivers are required

Design and Simulation:



It can be seen from the figure that the idle level of sclk is low level and the time of receiving data is rising edge, so the working mode of the SPI module is CPOL=0, CPHA=0, which is consistent with the previous design. When the transmit enables terminal tx_en=1, CS=1, the bus is in an idle state.

When CS=0, sck starts to output a clock signal, and the bus starts to transmit data. On the rising edge
of the first cycle, the host sends the highest bit of data_in to the slave by mosi, and sends tx_done=1
when transmitting to the 8th bit, indicating that the 8-bit data transfer is complete.

Drawbacks:

- More connections are required. Minimum 3 wires (in single slave) are required.Used for short distances.
- No flow control and no acknowledgment used in SPI
- Master and Slave relationships mapped to the devices can not be altered unlike I2C interface
- In order to add slave device, software needs to be changed and extra CS line is required to be added