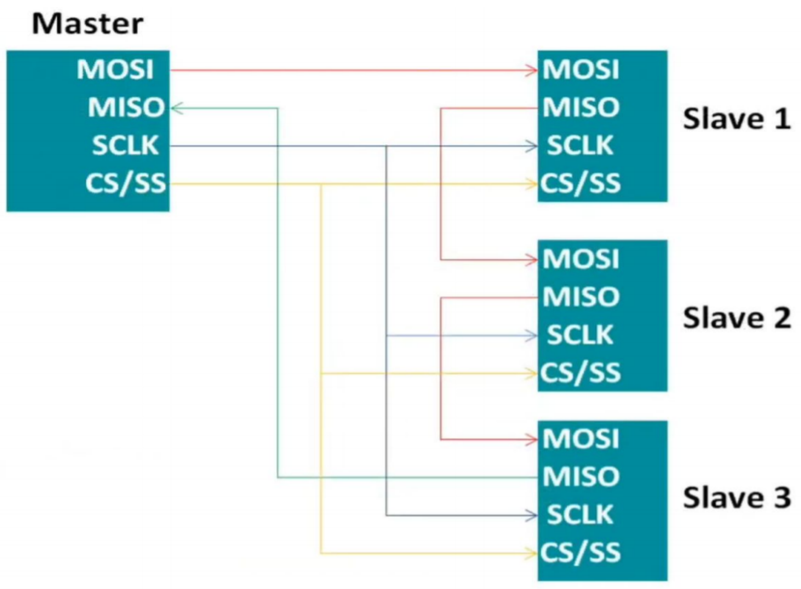
**ABSTRACT :**
Communication has a crucial role in human civilization. Nonetheless, new methods and ways were required to enhance the range of communication. numerous protocols into existence to meet the demands like I2C, AMBA, UART, SPI, etc The objective is to design and implement the SPI communication 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.

**Literature Survey:**
[An introduction to I2c and SPI protocols]
[Motorola, Inc. (Accessed 14.7.2016). Motorola's Spi Block Guide V04.01.]
[Prasad, M. (Accessed 9.6.2016). Serial Peripheral Interface - SPI Basics.]
[Byte Paradigm sprl. (Accessed 8.11.2016). Introduction to I²C and SPI protocols.]

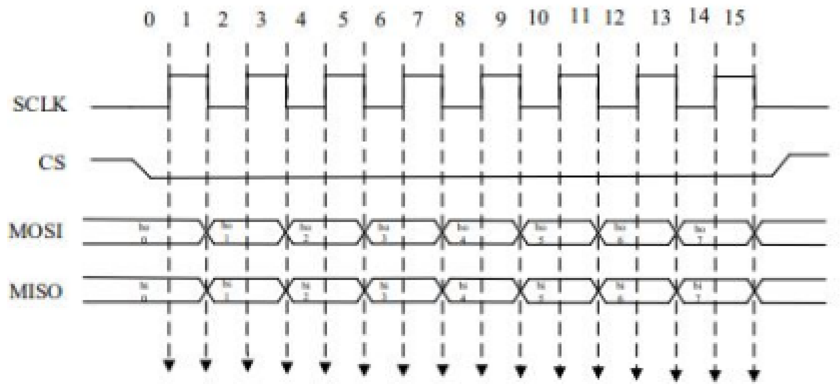
**Scope of Work:**
SPI is called a 4-wire bus as it requires four wires for its communication as shown below. In the case of single slave communications, we need only 3 wires, as slave selection (SS) is not required. So, SPI requires more communication lines in contrast to UART, I2C, USB, etc.



**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.

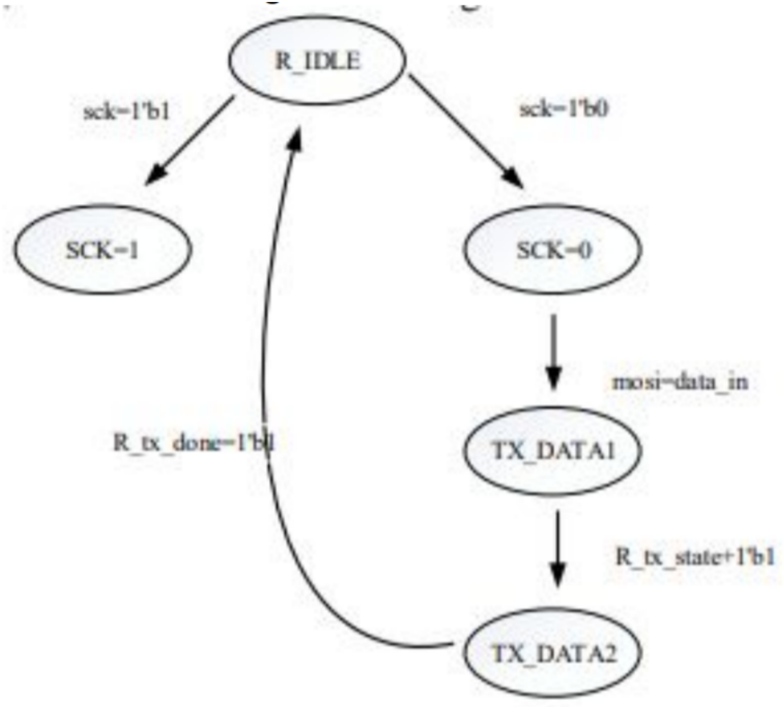
**SS/CS (Slave Select / Chip Select):** It is used by the master to send data by selecting a slave.

**Clock Polarity:** CPOL or CKP
Clock polarity is the idle/active state of the clock. If an idle state is
0, active state will be 1 and vice versa.
**Clock Phases**: PHASE, Inverted Clock Phase (Clock Edge) :
NCPHA or CKE. The clock phase or clock edge defines when to
transfer data. Data can be transferred from LOW (0) to HIGH
(1) or HIGH to LOW transitions. If PHASE is zero, data will be
sampled at the rising edge of the clock and if PHASE is
One, the kick-off of the kick-off the data will be sampled at the falling edge of the clock



| SPI Mode | CPOL | CPHA | Clock Polarity in Idle State | Clock Phase Used to Sample and/or Shift the Data |
|---|---|---|---|---|
| 0 | 0 | 0 | Logic low | Data sampled on rising edge and shifted out on the falling edge |
| 1 | 0 | 1 | Logic low | Data sampled on the falling edge and shifted out on the rising edge |
| 2 | 1 | 1 | Logic high | Data sampled on the falling edge and shifted out on the rising edge |
| 3 | 1 | 0 | Logic high | Data sampled on the rising edge and shifted out on the falling edge |

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



**Assumptions:**

- FPGA Reset is considered as **i_Rst_L**
- **i_Clk** as FPGA Clock
- **i_TX_Byte** as Byte to transmit on MOSI
- **i_TX_DV** as Data Valid Pulse with i_TX_Byte
- **o_TX_Ready** as Transmit Ready for next byte
- **o_RX_DV** as Data Valid pulse
- **o_RX_Byte** as Byte received on MISO
- **CPOL as Clock Polarity**
  CPOL=0 means clock idles at 0, leading edge is rising edge.
  CPOL=1 means clock idles at 1, leading edge is falling edge.
- **CPHA: Clock Phase**
  CPHA=0 means the "out" side changes the data on the trailing edge of the clock
  the "in" side captures data on the leading edge of the clock
  CPHA=1 means the "out" side changes the data on the leading edge of the clock
  the "in" side captures data on the trailing edge of the clock

**Tasks Involved:**

- Theoretical analysis of topic  chosen (SPI)
- Learning about the SPI's benchmarks and characteristics
- Designing SPI state diagram.
- Implementing SPI Master through Verilog
- Implementation of SPI master-slave model through Verilog
- Developing  TESTBENCH and testing the code

**Tasks Completed:**

- Theoretical analysis of topic  chosen (SPI)
- Learning about the SPI's benchmarks and their characteristics
- Designing SPI state diagram.

**The task to be completed:**

- Implementing SPI Master through Verilog
- Implementation of SPI master-slave model through Verilog
- Developing  TESTBENCH and testing the code

**Tools Utilized:**

- EDAPlayground
- Modelsim Altera

- Notepad++

**Conclusion:**
SPI (Serial Peripheral Interface) Master Creates master based on input configuration. Sends a byte one bit at a time on MOSI, Will also receive byte data one bit at a time on MISO. Any data on the input byte will be shipped out on MOSI.To kick off the transaction, the user must pulse i_TX_DV, This module supports multi-byte transmissions by pulsing,i_TX_DV, and loading up i_TX_Byte when o_TX_Ready is high. This module is only responsible for controlling Clk, MOSI, and MISO. If the SPI peripheral requires a chip-select, this must be done at a higher level.