# SPI MASTER

Design Document

# Contents

1. Introduction…………………………………………………………...3

2. SPI block diagram and pin description ……………………………………5

3. Transaction Structure……………………………………………………7

4. Test Cases…………………………………………………………………8

5. Simulation Results………………………………………………………9

# 1. Introduction

Serial Peripheral Interface (SPI) is a serial communication protocol that is used to connect low-speed devices. It is a full-duplex synchronous serial communication, which means that data can be simultaneously transmitted from both directions. The main advantage of the SPI is to transfer the data without any interruption. Many bits can be sent or received at a time in this protocol. In this protocol, devices are communicated in the master-slave relationship. The master device controls the slave device, and the slave device takes the instruction from the master device. The simplest configuration of the Serial Peripheral Interface (SPI) is a combination of a single slave and a single master. But, one master device can control multiple slave devices.

The SPI protocol uses the four wires for the communication. There are shown in the figure 1.

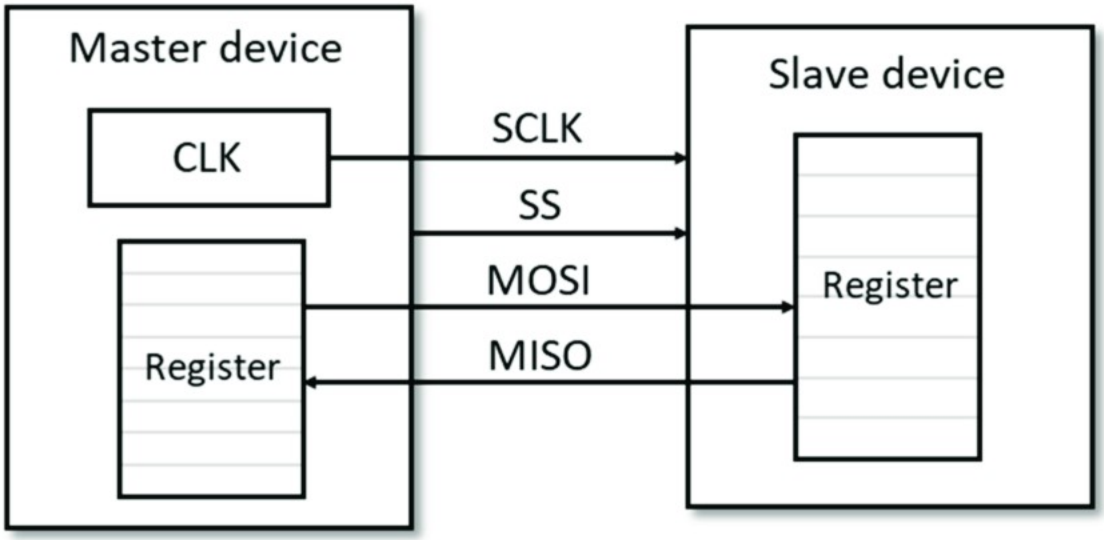


Figure 1: SPI block diagram

1. **MOSI:** MOSI stands for Master Output Slave Input. It is used to send data from the master to the slave.
2. **MISO:** MISO stands for Master Input Slave Output. It is used to send data from the slave to the master.
3. **SCK or SCLK (Serial Clock):** It is used to the clock signal.
4. **SS/CS (Slave Select / Chip Select):** It is used by the master to send data by selecting a slave.

Modes of operation: There are four modes of SPI operation based on clock polarity (CPOL) and clock phase (CPHA).

| Mode | CPOL | CPHA | Clock idle state | |
|---|---|---|---|---|
| 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 | 0 | Logic High | Data sampled on the rising edge and shifted out on the falling edge |
| 3 | 1 | 1 | Logic High | Data sampled on the falling edge and shifted out on the rising edge |

SPI Master here is designed for the following specifications:

1. Single master and single slave
2. 4- wire bidirectional
3. 8-bit fixed data width
4. Single byte transmission

Applications of SPI

Memory: SD Card, MMC, EEPROM, and Flash.
Sensors: Temperature and Pressure.
Control Devices: ADC, DAC, digital POTS, and Audio Codec.
Others: Camera Lens Mount, Touchscreen, LCD, RTC, video game controller, etc.

## 2. SPI Master Block Diagram and Pin Description

SPI master block diagram is shown in figure 2.

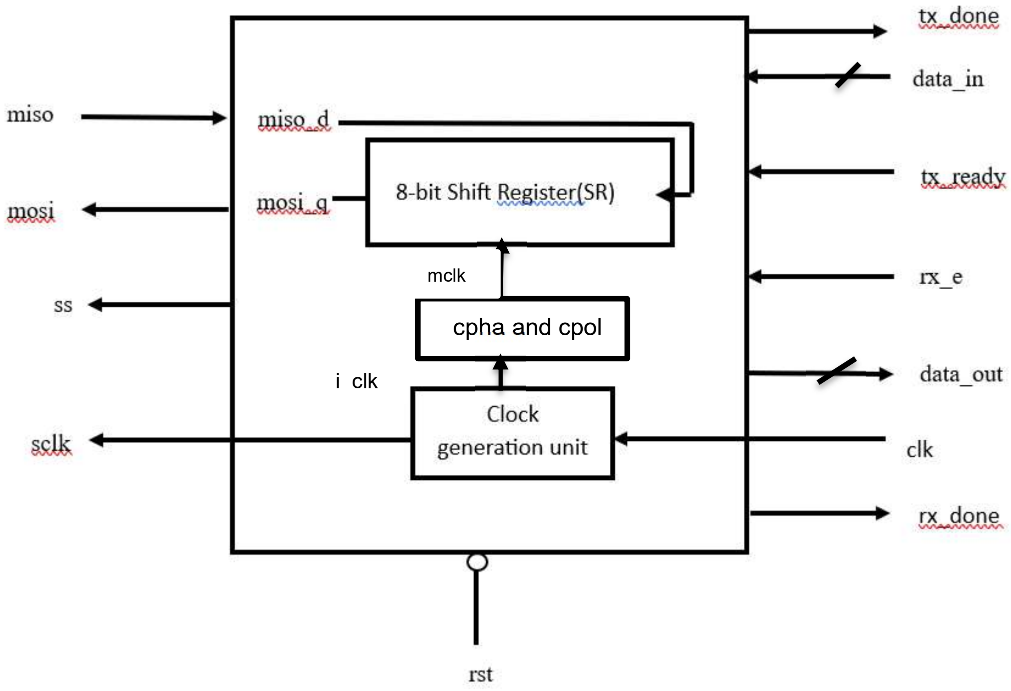


Figure 2: SPI Master Block Diagram

| Pin Name | Default value | Direction | Width | Description |
|---|---|---|---|---|
| rst | 1'b1 | Input | 1 | If rst=0, registers |
| sclk | Decide by the mode of SPI | Output | 1 | Whenever ss=0, clock signal will be sent to the slave over sclk. |
| ss | 1'd1 | Output | 1 | When ss='0', slave is selected by the master for communication. |

| | | | | |
|---|---|---|---|---|
| miso | 1’d0 | Input | 1 | Data bit is received over miso transmitted by slave. |
| mosi | 1’d0 | Output | 1 | Data bit is transmitted over mosi to slave. |
| data_in | 8’d0 | Input | 8 | This to be transmitted by the master is stored in data_in when tx_ready is ‘1’. |
| tx_ready | 1’d0 | Input | 1 | tx_ready is made ‘1’ whenever master wants to initiate the communication. |
| rx_e | 1’d0 | Input | 1 | Receive enable |
| data_out | 8’d0 | Output | 8 | Received byte is loaded on to data_out when rx_e=0 |
| tx_done | 1’d0 | Output | 1 | Whenever transmission of a byte is over tx_done will be ‘1’ |
| rx_done | 1’d0 | Output | 1 | Whenever reception of a byte is over rx_done will be ‘1’ |

| REGISTERS | | | |
|---|---|---|---|
| Name | Default Value | Width | Description |
| cpha | 1’d0 | 1 | Indicates clock phase |
| cpol | 1’d0 | 1 | Indicates clock polarity |
| sr | 8’d0 | 8 | Miso is shifted into the LSB of sr at rising edge of mclk |

| | | | |
|---|---|---|---|
| | | | and MSB of sr is shifted onto mosi at the falling edge of mclk. |
| rx_count | 4’d0 | 4 | Increments by one after reception of every bit (Counts from 0 to 8). |
| tx_count | 3’d0 | 3 | Increments by one after transmission of every bit (Counts from 0 to 7). |
| tx_ready | 1’d0 | 0 | Set to ‘1’ whenever master wants to start communication with slave. |
| empty | 1’d1 | 1 | Indicates whether the shift register is empty or not. If empty then only data can be loaded into shift register. |
| done | 1’d0 | 0 | If ‘1’ indicated received byte is transferred to shift register. |
| i_clk | Depends on cpol | 1 | SPI clock generated from the source clock. Mode needs to be checked with respect to i_clk |
| mclk | Depends on cpol | 1 | Based on the mode(cpha and cpol) mclk is generated for sampling and transmission. |

## 3. Transaction Structure

1. Whenever master wants to start the transaction with slave tx_ready should be made high.
2. Whenever tx_ready is high,data_in is loaded into sr register, slave select line pulled down and clock signal will be sent to the slave over sclk line.

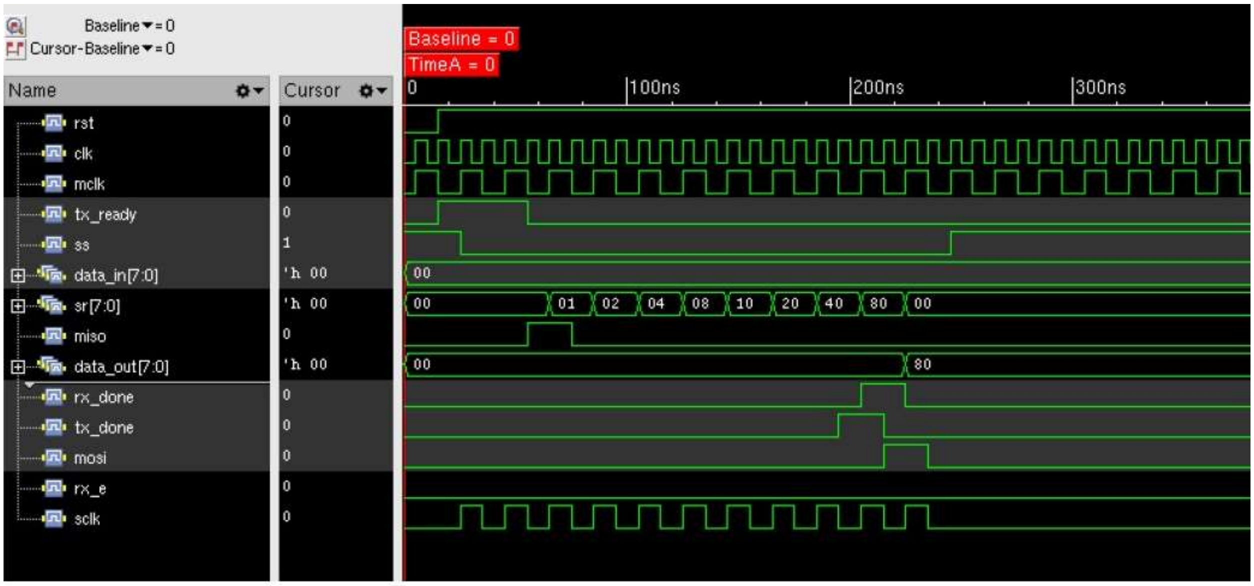
3. Data from slave is received over **miso** and shifted into LSB of sr.
4. Mode should be checked with respect to i_clk.
5. Upon receiving one bit of data, master transmits one bit of data to the slave i.e., MSB of sr is shifted onto **mosi.**
6. tx_count increments by one after the transmitting every bit.
7. rx_count increments by one after the receiving every bit.
8. If (rx_e==0 and rx_count==4'd8) then the received byte from sr is loaded into data_out bus.
9. Once one byte of data is transmitted/received slave select line is once again made 'high' and transaction is terminated.
10. To start the next transaction once again tx_ready should be made 'high'.

## 4. Test Cases

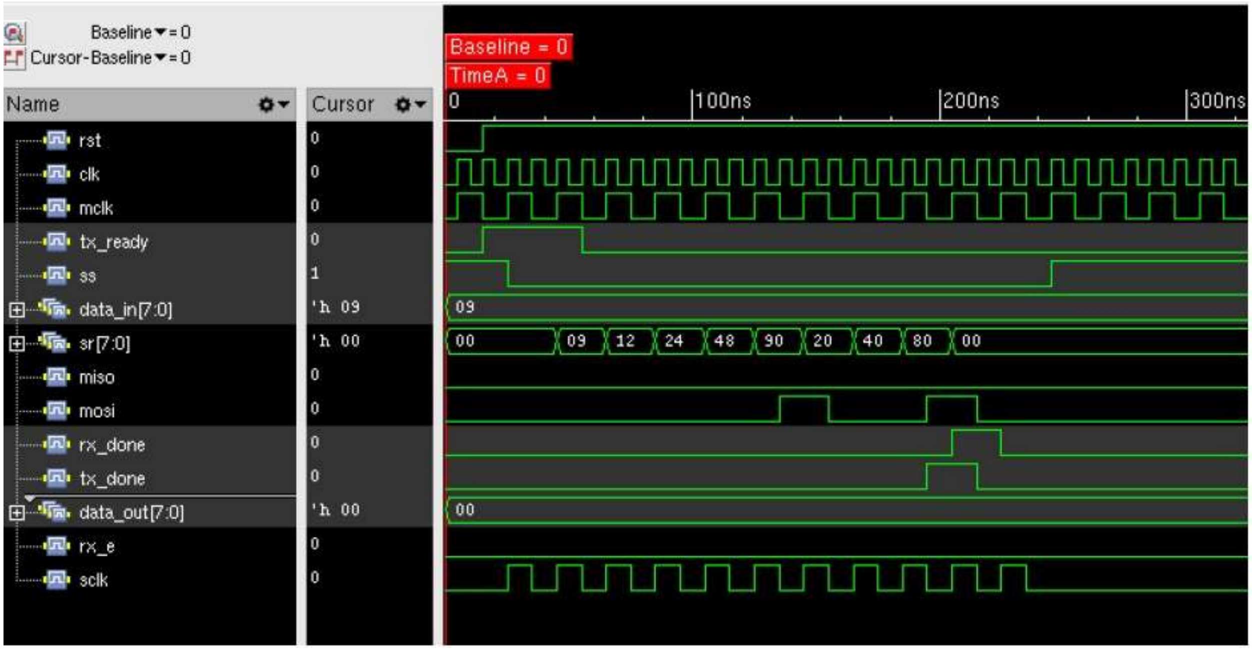
| Serial No. | Test Case | Results |
|---|---|---|
| 1. | Whenever tx_ready='1', ss='0', sclk should be sent to the slave and data_in should be loaded into sr register. | The desired results are obtained. |
| 2. | Data reception | Data is received bit by bit through miso and shifted into LSB of sr during rising edge of mclk. After receiving 8 bits data is loaded into data_out. |
| 3. | Data transmission | Data is loaded into sr when tx_ready=1 and transmitted over mosi bit by bit during falling edge of mclk |
| 4. | Full duplex verification | Data transmission and reception are done together. |

## 5. Simulation Results

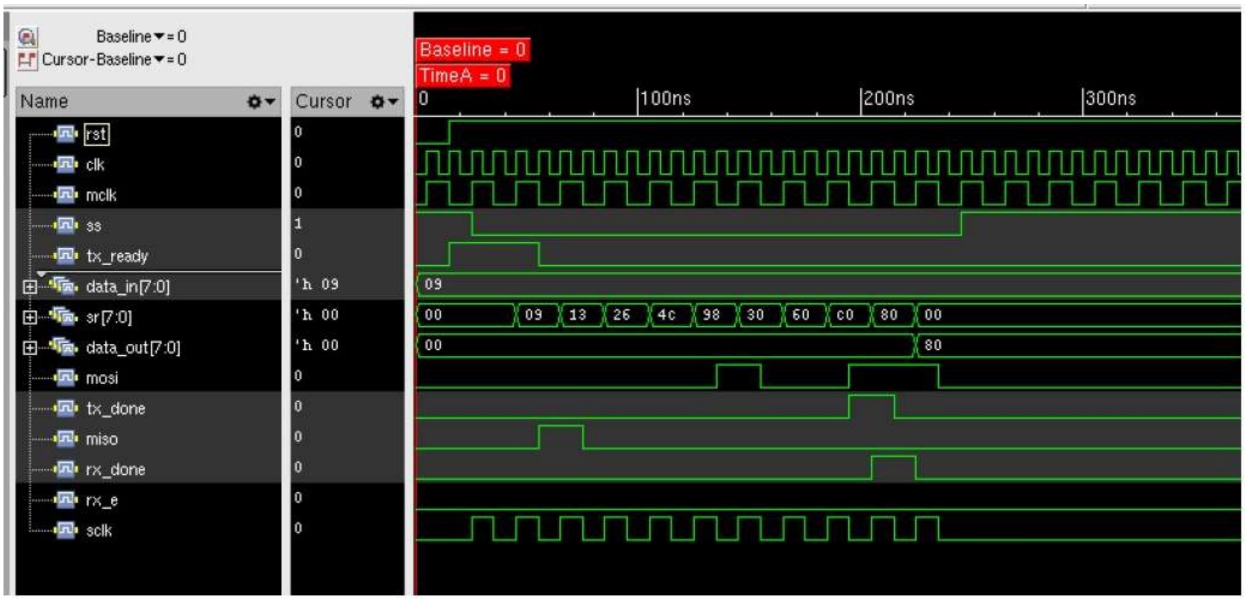
a. 10000000 is received from the slave and loaded into data_out after the completion of reception



b. 09 is transmitted to the slave



c. Both transmission and reception


Baseline ▾ = 0
Cursor-Baseline ▾ = 0
Baseline = 0
TimeA = 0
Name
Cursor
0
100ns
200ns
300ns
rst 0
clk 0
mclk 0
ss 1
tx_ready 0
data_in[7:0] 'h 09
sr[7:0] 'h 00
data_out[7:0] 'h 00
mosi 0
tx_done 0
miso 0
rx_done 0
rx_e 0
sclk 0
09
00
09
13
26
4c
98
30
60
c0
80
00
00
80