INTRODUCTION

Serial to Peripheral Interface (SPI) is a hardware/firmware communications protocol developed by Motorola and later adopted by others in the industry. Sometimes SPI is also called a "four wire" serial bus.The Serial Peripheral Interface or SPI-bus is a simple 4-wire serial communications interface used by many microprocessor/microcontroller peripheral chips that enables the controllers and peripheral devices to communicate with each other. Even though it is developed primarily for the communication between host processor and peripherals, a connection of two processors via SPI is just as well possible. The SPI bus, which operates at full, is a synchronous type data link setup with a Master / Slave interface . Both single-master and multi-master protocols are possible in SPI. The SPI Bus is usually used only on the PCB. The SPI Bus was designed to transfer data between various IC chips, at very high speeds

SERIAL PERIPHERAL INTERFACE DESIGN

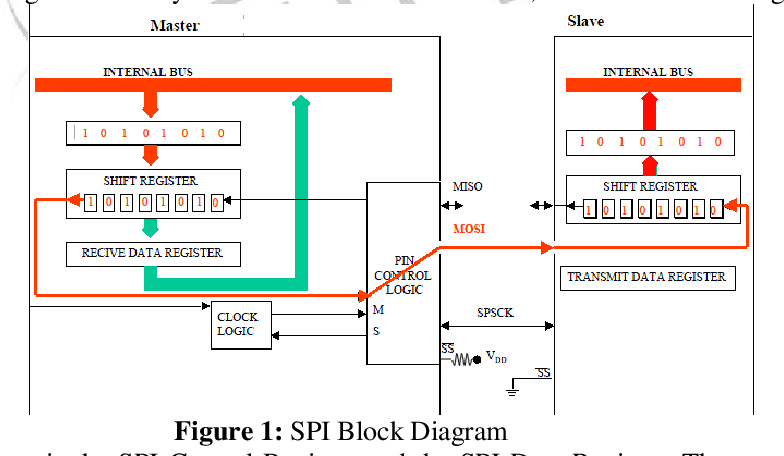
The internal architecture of SPI mainly consists of two modules, master module and slave module as shown in figure 1. The SPI module allows a full duplex, synchronous, serial communication between the MCU and peripheral devices [1]. The main parts of the SPI are status, control and data registers, shifter logic, baud rate generator, master/slave control logic and port control logic. Software can poll the SPI status flags or the SPI operation can be interrupt driven. The SPI module has a total of 4 external pins:

• Master out Slave in (MOSI) - Output data from the master to the inputs of the slaves.

• Master in Slave out (MISO) - Output data from a slave to the input of the master.

• Serial Clock (SCLK) - Clock driven by the master to slaves, used to synchronize the data

• Slave Select (SS) - Select signal driven by the master to individual slaves, used to select the target slave



Data Transmission

To begin SPI communication, the main must send the clock signal and select the subnode by enabling the CS signal. Usually chip select is an active low signal; hence, the main must send a logic 0 on this signal to select the subnode. SPI is a full-duplex interface; both main and subnode can send data at the same time via the MOSI and MISO lines respectively. During SPI communication, the data is simultaneously transmitted (shifted out serially onto the MOSI/SDO bus) and received (the data on the bus (MISO/SDI) is sampled or read in). The serial clock edge synchronizes the shifting and sampling of the data. The SPI interface provides the user with flexibility to select the rising or falling edge of the clock to sample and/or shift the data. Please refer to the device data sheet to determine the number of data bits transmitted using the SPI interface.

### Clock Polarity and Clock Phase

In SPI, the main can select the clock polarity and clock phase. The CPOL bit sets the polarity of the clock signal during the idle state. The idle state is defined as the period when CS is high and transitioning to low at the start of the transmission and when CS is low and transitioning to high at the end of the transmission. The CPHA bit selects the clock phase. Depending on the CPHA bit, the rising or falling clock edge is used to sample and/or shift the data. The main must select the clock polarity and clock phase, as per the requirement of the subnode. Depending on the CPOL and CPHA bit selection, four SPI modes are available. Table 1 shows the four SPI modes.

|  |  |  |
| --- | --- | --- |
| **SPI Mode** | **CPOL** | **Clock Polarity in Idle State** |
| 0 | 0 | Logic low |
| 1 | 0 | Logic low |
| 2 | 1 | Logic high |
| 3 | 1 | Logic high |