Introduction

- The Serial Peripheral Interface bus (SPI) is
  - ✓ synchronous
  - ✓ serial communication interface specification
  - ✓ used for short distance communication
- The interface was developed by Motorola in 1980s
- SPI devices communicate in full duplex mode using a master-slave architecture with a single master.
  - ✓ The master device originates the frame for reading and writing.
  - ✓ Multiple slave devices are supported through selection with individual slave select (SS) lines.

The trouble with asynchronous protocols

- The common serial port, such as UART, is called “asynchronous” because there is no control over when data is sent or any guarantee that both sides are running at precisely the same rate.
- There can be an issue when two systems with slightly different clocks try to communicate with each other.
- To mitigate this issue, asynchronous serial connections add extra start and stop bits to each byte, so as to help the receiver sync up to data as it arrives.
- Both sides must also agree on the same baud rate in advance.

The trouble with asynchronous protocols

- Asynchronous serial works just fine, but

✓ has a lot of overhead in both the extra start and stop bits sent with every byte

✓ the complex hardware required to send and receive data

✓ if both sides aren't set to the same speed, the received data will be garbage. This is because the receiver is sampling the bits at very specific times If the receiver is looking at the wrong times, it will see the wrong bits.

How is ‘synchronous’ beneficial?

- A “synchronous” data bus uses separate lines for data and clock that keeps both sides in perfect sync.
- The clock is an oscillating signal that tells the receiver exactly when to sample the bits on the data line.

✓ This could be the rising (low to high) or falling (high to low) edge of the clock signal; the datasheet will specify which one to use.

- When the receiver detects that edge, it will immediately look at the data line to read the next bit
- Because the clock is sent along with the data, specifying the speed isn’t important, although devices will have a top speed at which they can operate

The SPI Protocol - basics

- The SPI bus can operate with a single master device and with one or more slave devices.
- One unique feature of SPI is that data can be transferred without interruption.
- Any number of bits can be sent or received in a continuous stream.
- ✓ With I2C and UART, data is sent in packets, limited to a specific number of bits.
- ✓ Start and stop conditions define the beginning and end of each packet, so the data is interrupted during transmission.

The SPI Protocol - basics

- Devices communicating via SPI are in a master-slave relationship.
- In SPI, only one side generates the clock signal

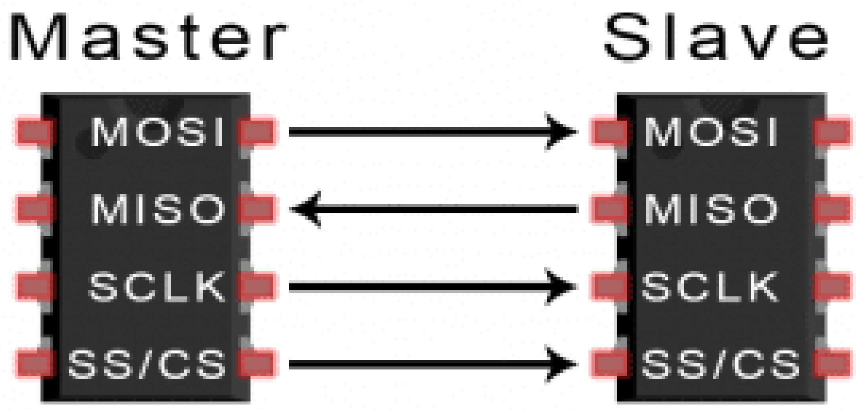
✓ The side that generates the clock is called the "master", and the other side is called the "slave".

- The master is the controlling device (usually a microcontroller), while the slave (usually a sensor, display, or memory chip) takes instruction from the master.
- There is always only one master but there can be multiple slaves.

The SPI Interface

The SPI bus specifies four logic signals:

- ✓ SCLK: Serial Clock (output from master).
- ✓ MOSI: Master Output Slave Input (data output from master).
- ✓ MISO: Master Input Slave Output, or Master In Slave Out (data output from slave).
- ✓ SS: Slave Select (often active low, output from master).



The SPI Interface

Serial Clock (SCLK)

- SPI is a synchronous communication protocol.
- The clock signal synchronizes the output of data bits from the master with the sampling of bits by the slave.
- One bit of data is transferred in each clock cycle, so the speed of data transfer is determined by the frequency of the clock signal.
- SPI communication is always initiated by the master since the master configures and generates the clock signal.

The SPI Interface

Slave Select

- The master can choose which slave it wants to communicate to by setting the slave's CS/SS line to a low voltage level.
- In the idle, non-transmitting state, the slave select line is kept at a high voltage level.

✓ Multiple CS/SS pins may be available on the master, which allows for multiple slaves to be wired in parallel.

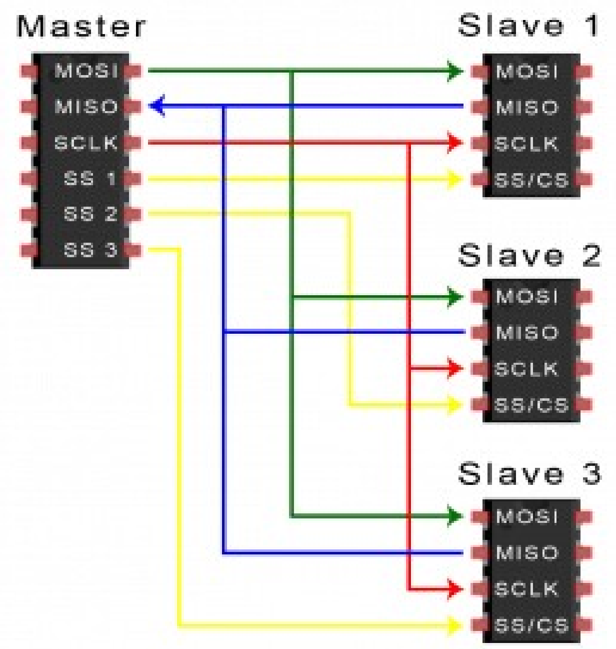
✓ If only one CS/SS pin is present, multiple slaves can be wired to the master by daisy-chaining

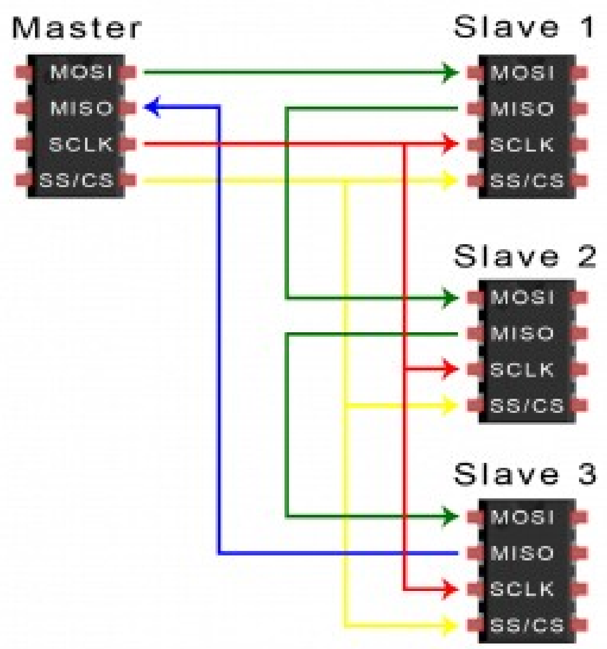
The SPI Interface

Multiple Slaves

- ✓ Multiple CS/SS pins are available on the master

- ✓ Only one CS/SS pin is present, multiple slaves are wired to the master by daisy-chaining





The Protocol

1. To begin communication, the bus master configures the clock, using a frequency supported by the slave device, typically up to a few MHz.
2. The master then selects the slave device with a logic level 0 on the select line.

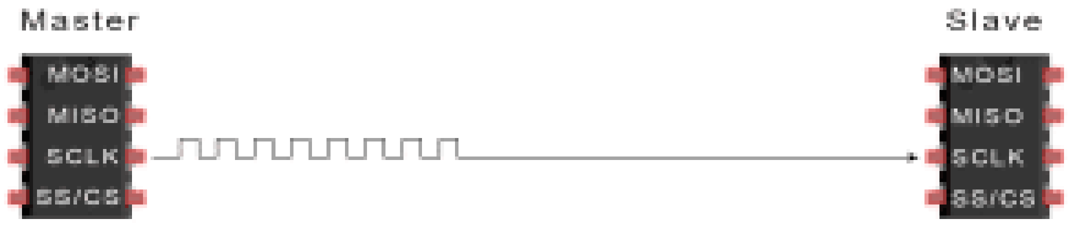
✓ If a waiting period is required, such as for an analog-to-digital conversion, the master must wait for at least that period of time before issuing clock cycles.

3. During each SPI clock cycle, a full duplex data transmission occurs.

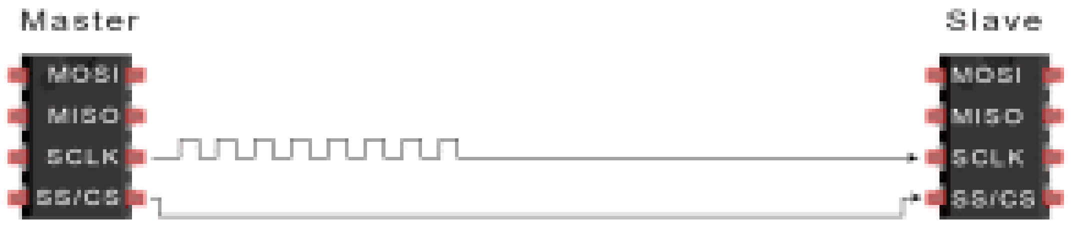
✓ The master sends a bit on the MOSI line and the slave reads it, while the slave sends a bit on the MISO line and the master reads it.

The Protocol (summary)

1. The master outputs the clock signal

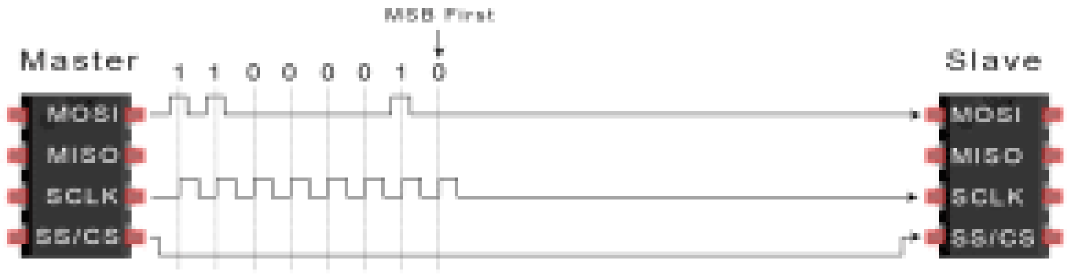


2. The master switches the SS/CS pin to a low voltage state, which activates the slave:

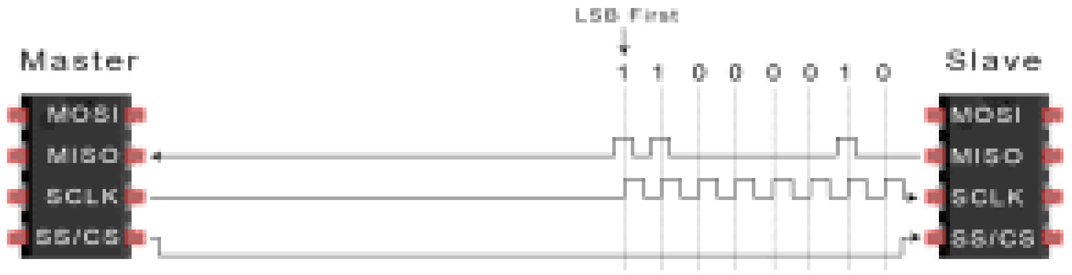


The Protocol (summary)

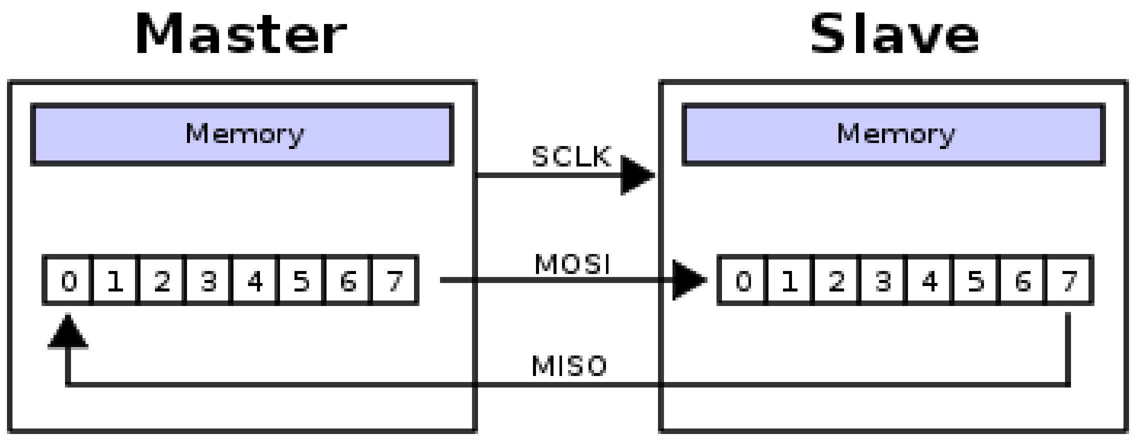
3. The master sends the data one bit at a time to the slave along the MOSI line. The slave reads the bits as they are received



4. If a response is needed, the slave returns data one bit at a time to the master along the MISO line. The master reads the bits as they are received



Data Transmission in SPI



- Transmissions normally involve two shift registers of some given word size (may be 8eight bits), one in the master and one in the slave
- They are connected in a virtual ring topology.
- Data is usually shifted out with the most-significant bit first. On the clock edge, both master and slave shift out a bit and output it on the transmission line to the counterpart.

Data Transmission in SPI

- On the next clock edge, at each receiver the bit is sampled from the transmission line and set as a new least-significant bit of the shift register.
- After the register bits have been shifted out and in, the master and slave have exchanged register values.
- If more data needs to be exchanged, the shift registers are reloaded and the process repeats.
- Transmission may continue for any number of clock cycles.
- When complete, the master stops toggling the clock signal, and typically deselects the slave.

Advantages of SPI

- Full duplex
- Push-pull drivers (as opposed to open drain) provide good signal integrity and high speed
- Higher throughput than I^2C or SMBus.
- Not limited to any maximum clock speed, enabling potentially high speed
- Not limited to 8-bit words
- Slaves use the master's clock and do not need precision oscillators
- Slaves do not need a unique address — unlike I^2C or GPIB or SCSI

Limitations of SPI

- Requires more pins on IC packages than I^2C
- No hardware flow control by the slave (but the master can delay the next clock edge to slow the transfer rate)
- No hardware slave acknowledgment (the master could be transmitting to nowhere and not know it)
- Typically supports only one master device (depends on device's hardware implementation)
- No error-checking protocol is defined

Applications of SPI

SPI is used to talk to a variety of peripherals, such as

- Sensors: temperature, pressure, ADC, touchscreens, video game controllers
- Control devices: audio codecs, digital potentiometers, DAC
- Memory: flash and EEPROM
- Real-time clocks
- LCD, sometimes even for managing image data
- Any MMC or SD card

Review Questions

1. Compare and contrast UART, I2C and SPI, citing the features, advantages and disadvantages of each.
2. What are the demerits of asynchronous communication protocols? How are they overcome in SPI?
3. List out the Interface signals used in SPI protocol and brief out their significance.
4. List the set of operations involved in completing a serial communication using SPI protocol. Draw the hardware interface, in support.
5. List out the advantages and limitations of SPI protocol. Also, list some of the applications.