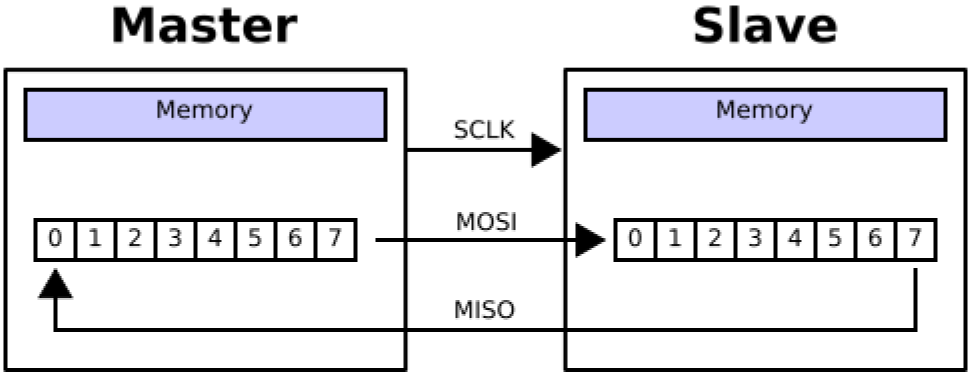
Serial Peripheral Interface

- Was developed by motorola
- It is commonly used for short distance communications (send data between microcontrollers and small peripherals)
- It uses separate clock and data lines along with select line (to chose the slave device)
- One master and multiple slave

Architecture:
SPI is a four wire interface and has synchronous serial interface .the architecture is master slave architecture.
The device that generates the clock signals is called masterand the device that uses the clock is called slave. Both master and slave consists of 8 bit serial in parallel out shift registers .the two shift registers are connected as per picture shown below.



A SPI device has four wires :
SS :
It stands for slave select , the slave select connections can be understood as a gpio controlling the chip enable pin of any peripheral. Ie master controller's gpio is connected to chip select pin of slave IC.

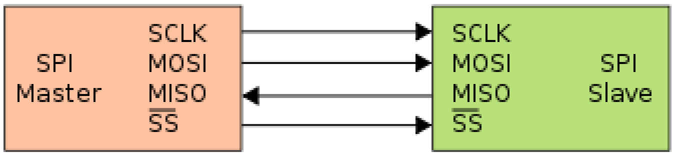
SCLK: It stands for serial clock. Master toggles this pin to generate a serial clock by which is used by the slave device.it is an oscillating signal which tells the devices when to sample the bits on the data line.

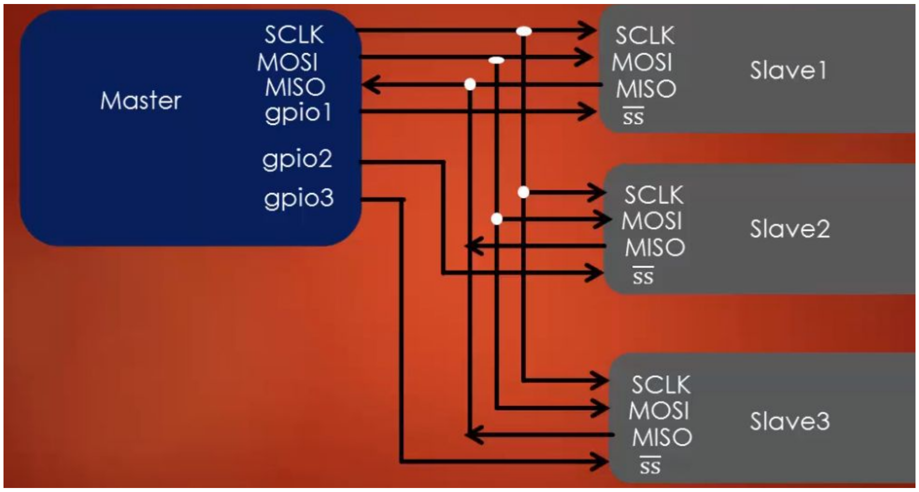
MOSI: It stands for master output slave input and is a data line from master to slave.
SIMO, MTSR ,SDO, DO, DOUT, SO

MISO: It stands for master input slave output
SOMI, MRST,SDI, DI, DIN, SI

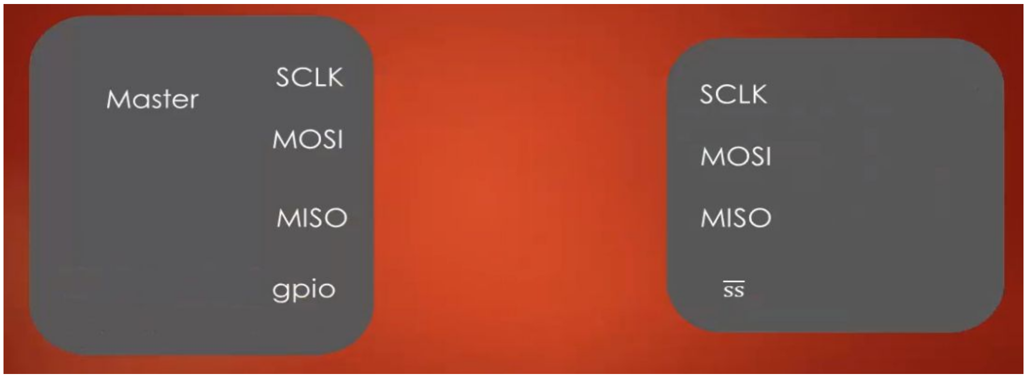
Operation

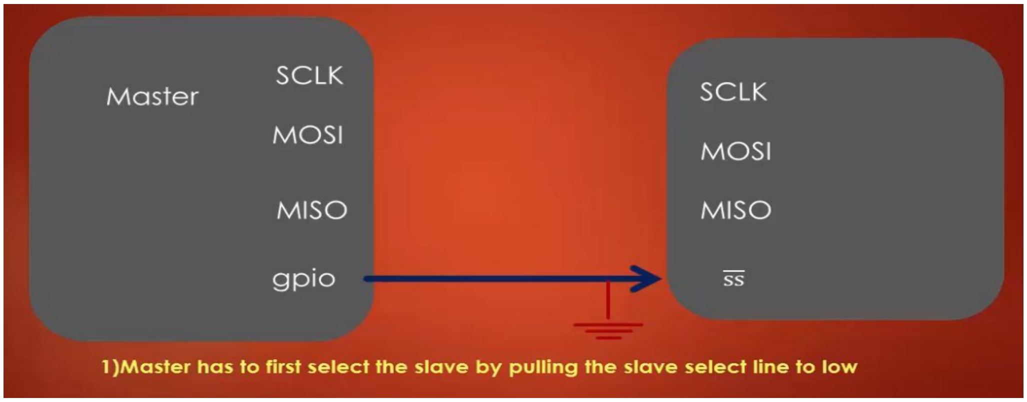
The SPI bus can operate with a single master device and with one or more slave devices. A slave cannot change it’s mode from slave to master and drive other slaves. This means that one central device initiates communication with all the slaves.



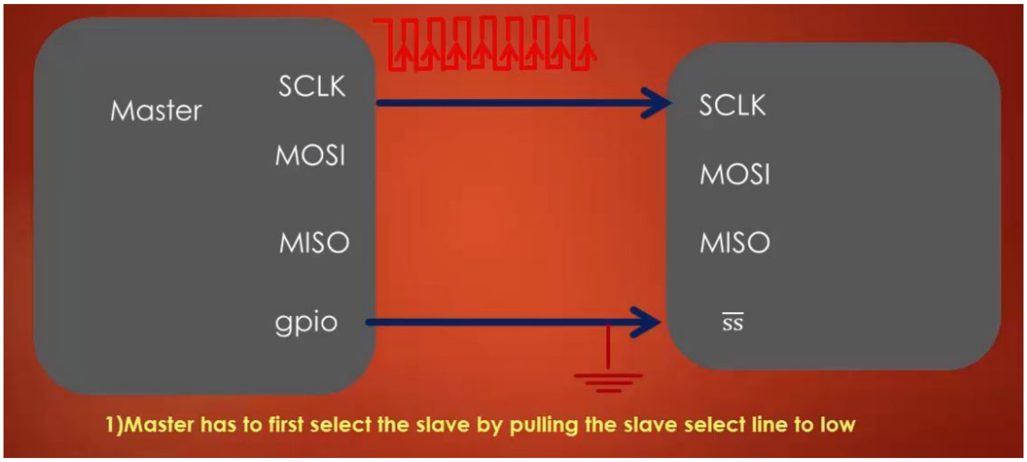


When master wishes to communicate to slave
it first select the slave by pulling down the corresponding slave select line.
And generates the clock signal usable by both master and slave
Master generates the information on mosi line and samples the miso line at the same time

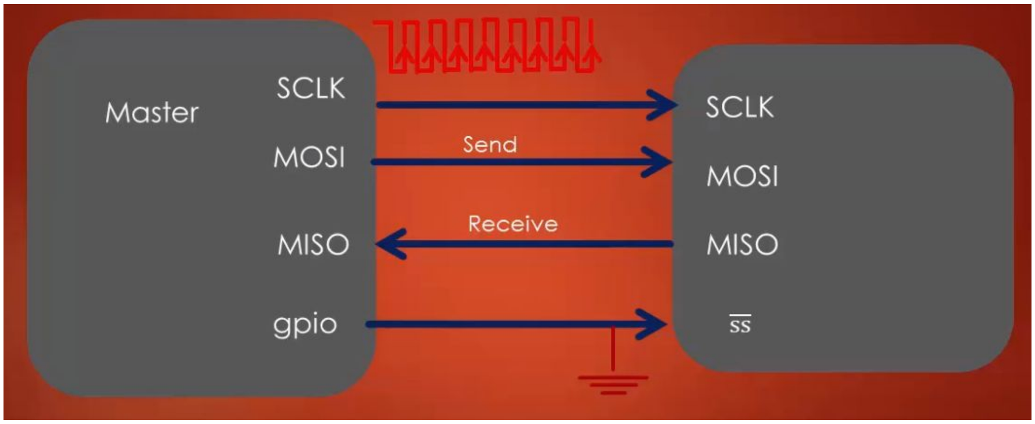




1)Master has to first select the slave by pulling the slave select line to low



1)Master has to first select the slave by pulling the slave select line to low

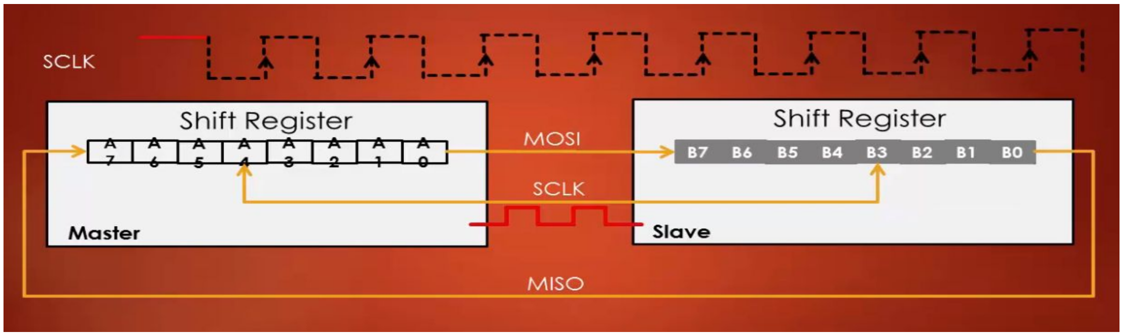


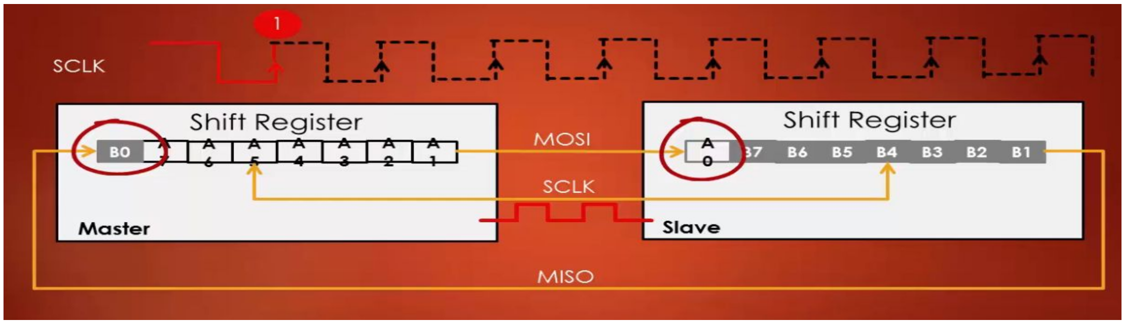
Data transmission

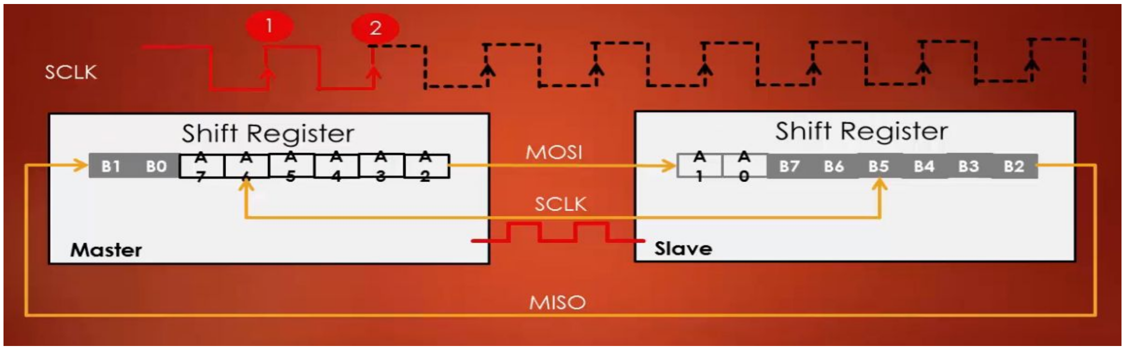
- To begin communication, the bus master configures the clock, using a frequency supported by the slave device
- The master then selects the slave device with a logic level 0 on the select line. If a waiting period is required, such as for an analog-to-digital conversion, the master must wait for at least that period of time before issuing clock cycles
- 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.
- Transmissions normally involve two shift registers of some given word size, such as eight 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. 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

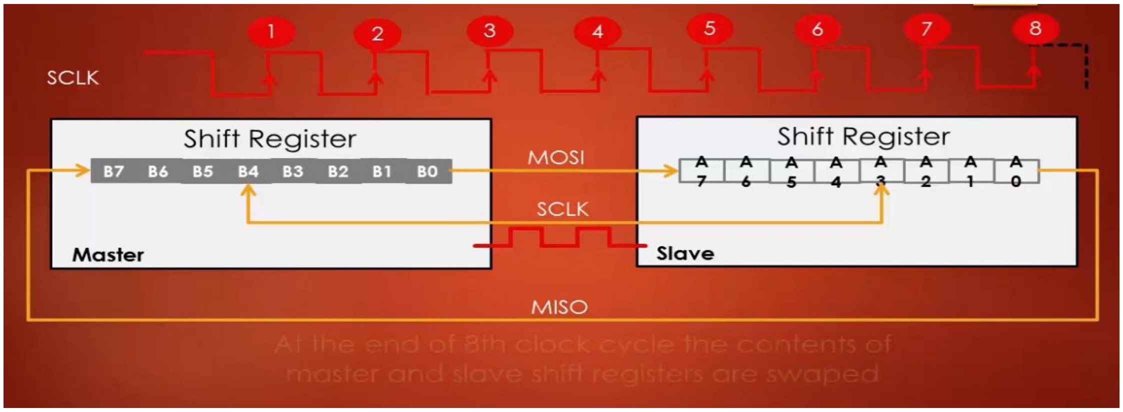
  How SPI hardware works during data transmission

  - Both master and slave spi engine consists of simple shift registers.
  - Master shifts out each bits of data on MISO line to the shift register of slave device which eventually causes the data to be shifted out at slave side on the MISO line.





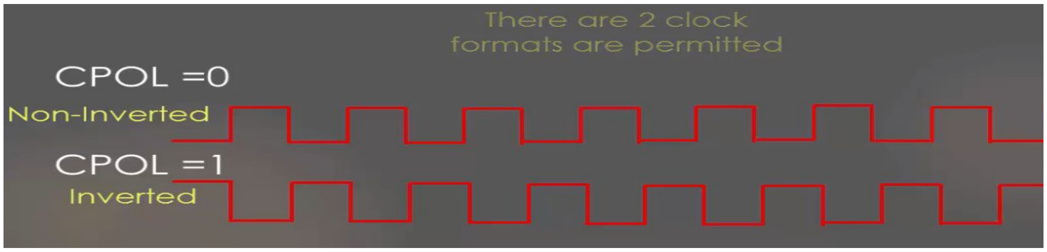


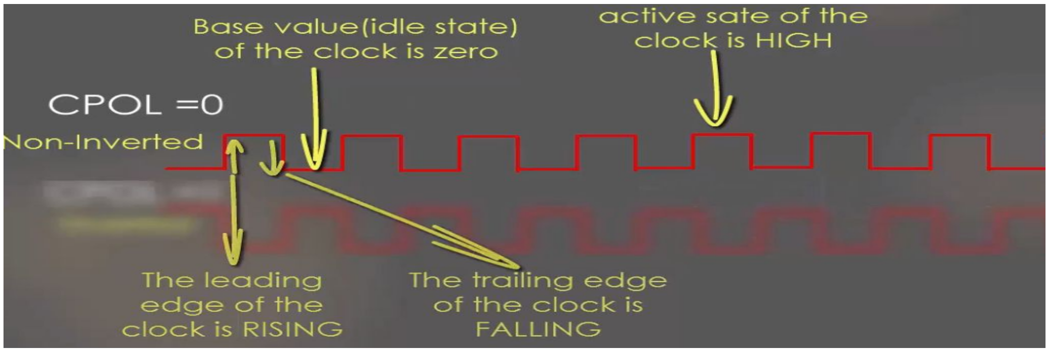


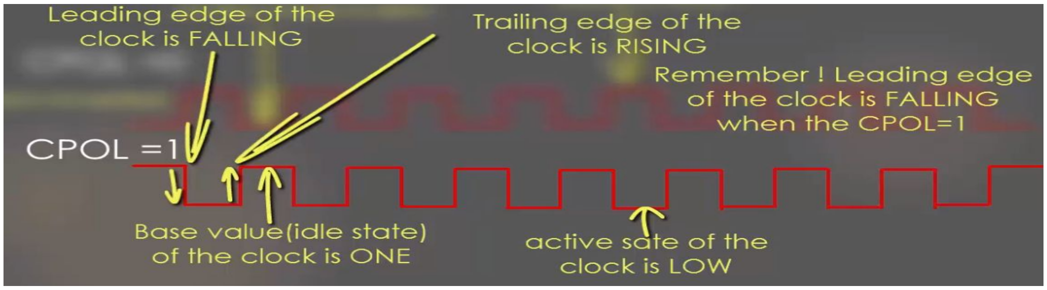
At the end of 8th clock cycle the contents of master and slave shift registers are swaped

Clock polarity

- CPOL stands for clock polarity, thsi paramatert decides the clock format to be used by SPI protocol. There are two permitted clock formats (inverted and non inverted) .
-
    - CPOL=0
        - is a non-inverted clock
        - Base value is zero (clock starts from zero / low)
        - idles at 0 , active state of clock is 1 (one / high)
        - each cycle consists of a pulse of 1
        - the leading edge is a rising edge, and the trailing edge is a falling edge.
    - CPOL=1
        - is an inverted clock
        - Base value is 1 (clock starts from high)
        - idles at 1, active state is 0
        - each cycle consists of a pulse of 0.
        - the leading edge isa falling edge, and the trailing edge is a rising edge.



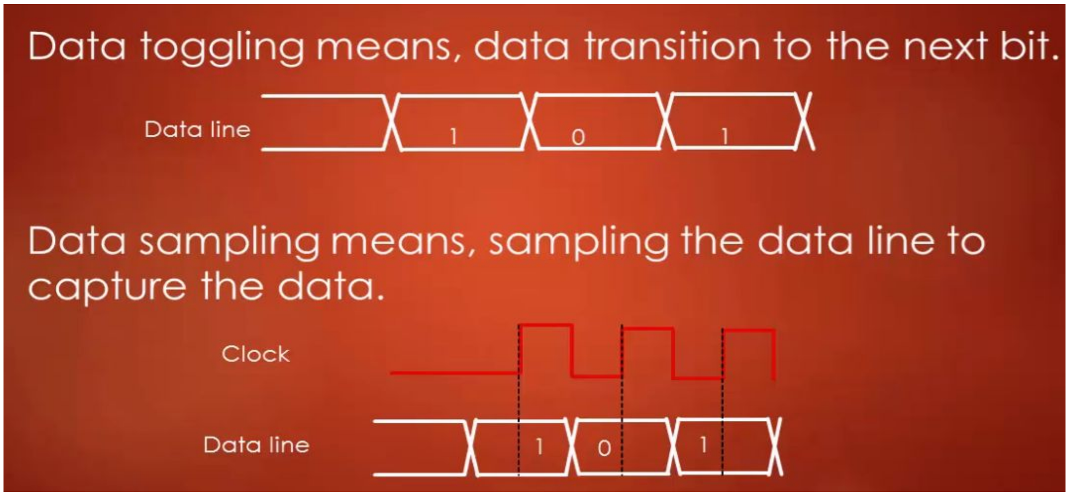




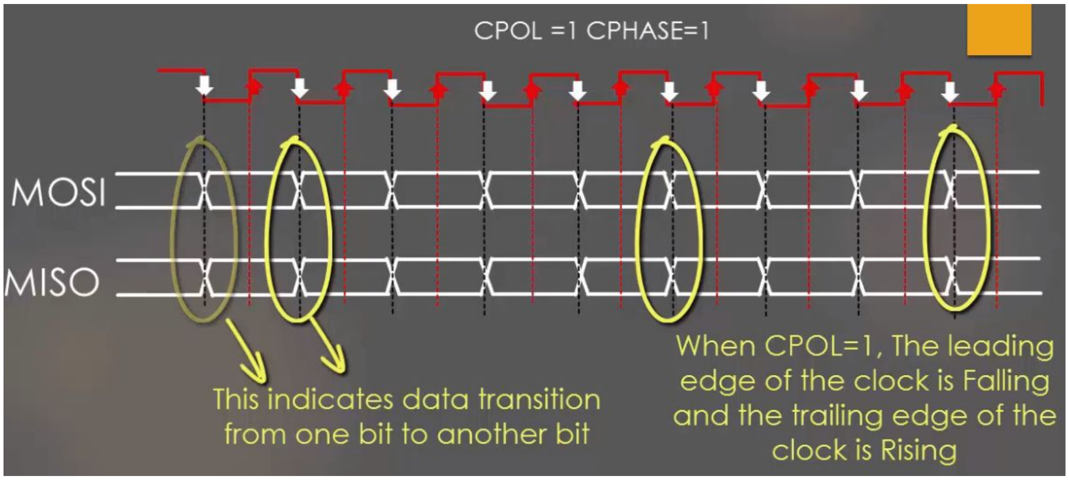
Clock Phase

- CPHA determines when the data has to be toggled and when the data has to be sampled on the spi data lines.
    - Data toggling means the data transition to next bit
    - Data sampling means sampling the data lins to get the data.

- For CPHA=0,
    - The data is always sampled on the leading edge of the clock cycle.
    - The data is toggled on trailing edge of the clock cycle .
    -
- CPHA=1,
    - Data is always sampled on the trailing edge of the clock
    - the "out" side changes the data on the leading edge of the current clock cycle, while the "in" side captures the data on (or shortly after) the trailing edge of the clock cycle.

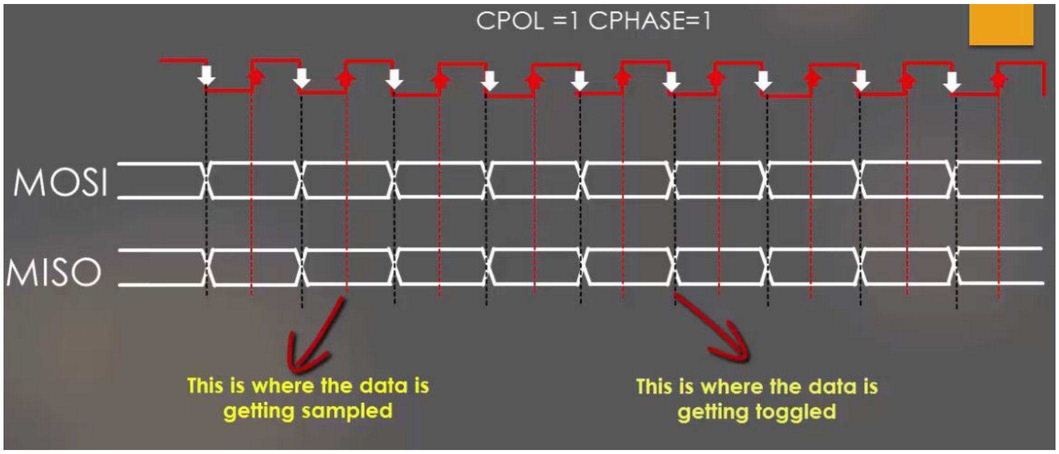

Data toggling means, data transition to the next bit.

Data sampling means, sampling the data line to capture the data.


CPOL =1 CPHASE=1

This indicates data transition from one bit to another bit

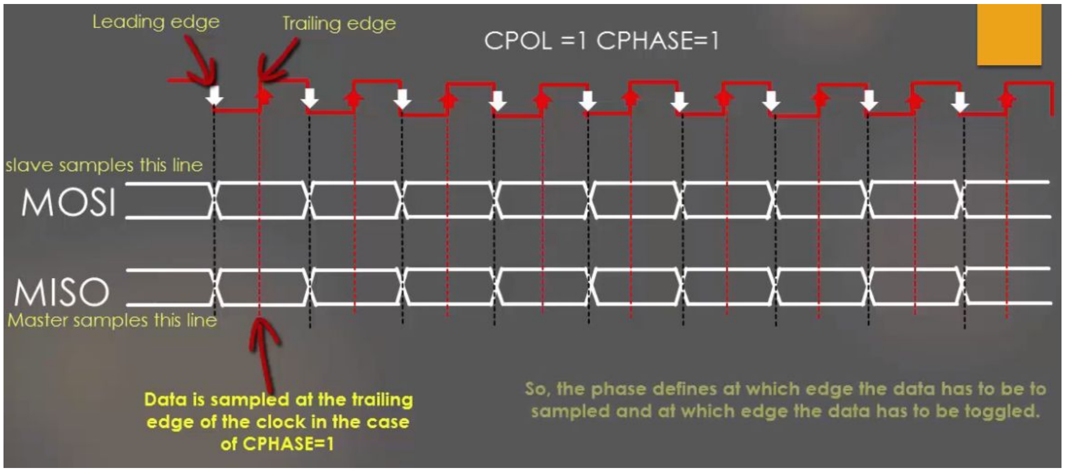
When CPOL=1, The leading edge of the clock is Falling and the trailing edge of the clock is Rising



CPOL =1 CPHASE=1

This is where the data is getting sampled

This is where the data is getting toggled



CPOL =1 CPHASE=1

Data is sampled at the trailing edge of the clock in the case of CPHASE=1

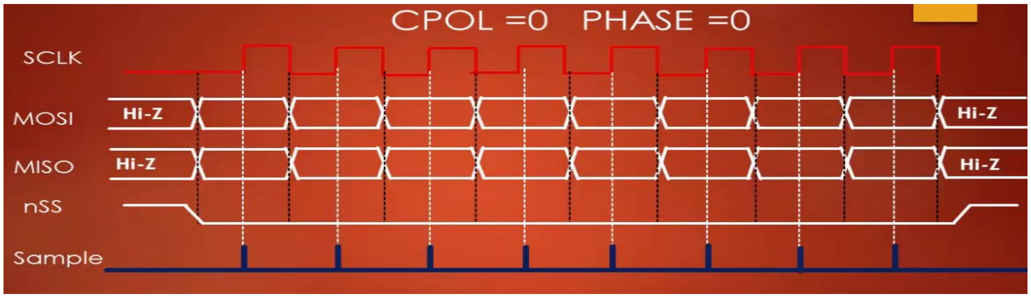
So, the phase defines at which edge the data has to be to sampled and at which edge the data has to be toggled.

Mode numbers

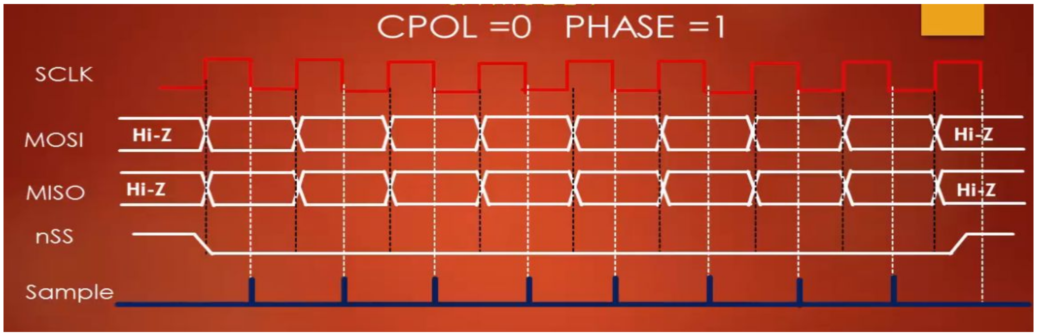
The combinations of polarity and phases are often referred to as modes which are commonly numbered according to the following convention, with CPOL as the high order bit and CPHA as the low order bit:

Mode	CPOL	CPHA
0	0	0
1	0	1
2	1	0
3	1	1

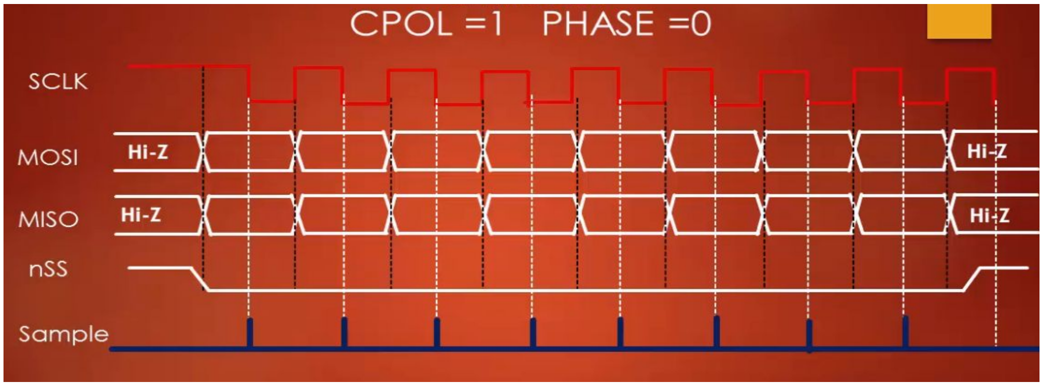
MODE 0



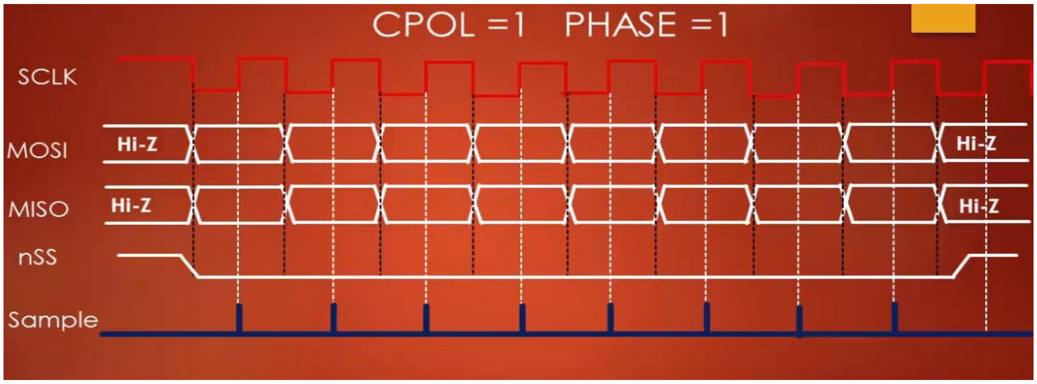
MODE 1



MODE 2



MODE 3



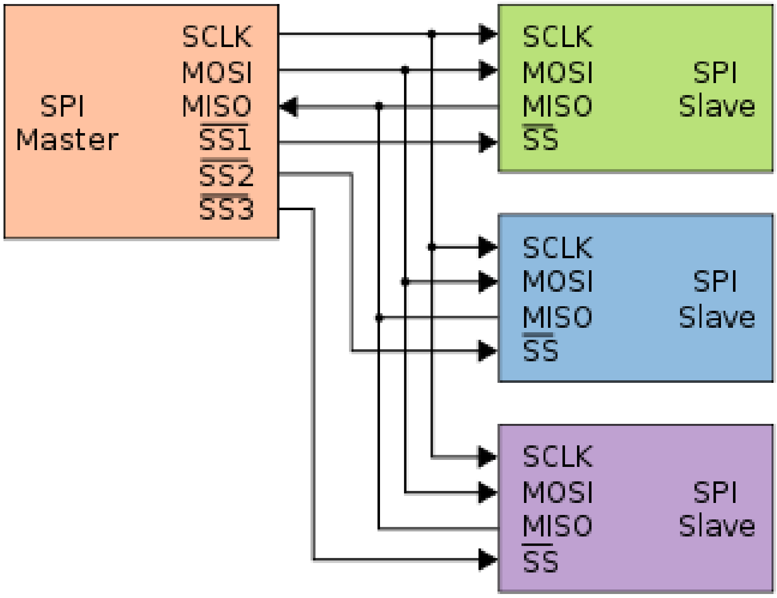
Configurations for SPI

- UNIdirectional SPI
- BIdirectional SPI
- DUAl SPI serial buses
- Three-wire serial buses
- Quad SPI serial buses

Other configuration

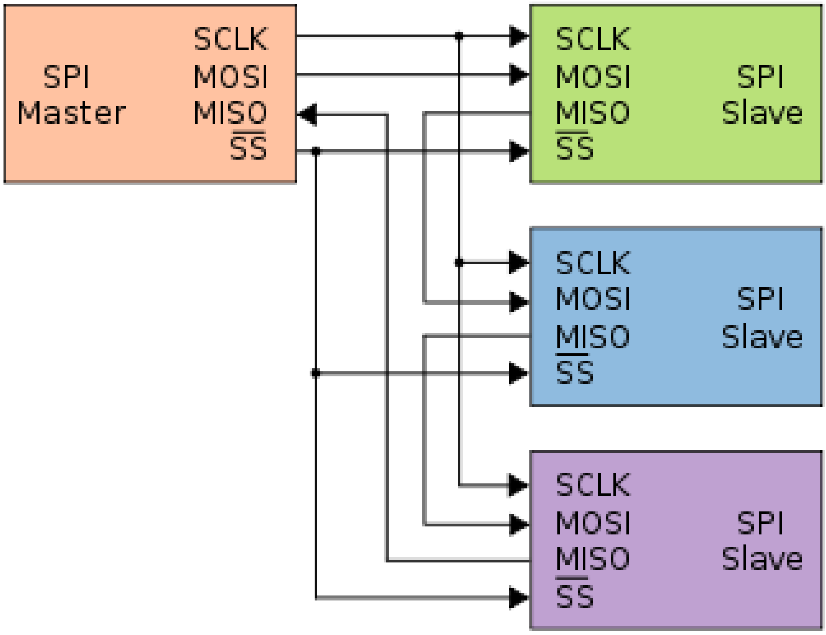
Independent slave configuration

there is an independent chip select line for each slave. A pull-up resistor between power source and chip select line is highly recommended for each independent device to reduce cross-talk between devices.This is the way SPI is normally used. Since the MISO pins of the slaves are connected together, they are required to be tri-state pins (high, low or high-impedance).



Daisy chain configuration:

the first slave output being connected to the second slave input, etc. The SPI port of each slave is designed to send out during the second group of clock pulses an exact copy of the data it received during the first group of clock pulses. The whole chain acts as a communication shift register; daisy chaining is often done with shift registers to provide a bank of inputs or outputs through SPI. Each slave copies input to output in the next clock cycle until active low SS line goes high. Such a feature only requires a single SS line from the master, rather than a separate SS line for each slave.



SAMPLE CODE

```
/*
 * Simultaneously transmit and receive a byte on the SPI.
 * Polarity and phase are assumed to be both 0, i.e.:
 *   - input data is captured on rising edge of SCLK.
 *   - output data is propagated on falling edge of SCLK.
 * Returns the received byte.
 */

uint8_t SPI_transfer_byte(uint8_t byte_out)
{
    uint8_t byte_in = 0;
    uint8_t bit;
    for (bit = 0x80; bit; bit >>= 1) {
              write_MOSI((byte_out & bit) ? HIGH : LOW);    // Shift-out a bit to the MOSI line
              delay(SPI_SCLK_LOW_TIME);                     // Delay for at least the peer's setup
time
               write_SCLK(HIGH);                            // Pull the clock line high
               if (read_MISO() == HIGH)                     // Shift-in a bit from the MISO line
               byte_in |= bit;
               delay(SPI_SCLK_HIGH_TIME);                   // Delay for at least the peer's hold time
               write_SCLK(LOW);                             // Pull the clock line low
    }
    return byte_in;
}
```

Advantages:

- Full duplex communication in the default version of this protocol
- provide good signal integrity and high speed
- Not limited to any maximum clock speed
- Not limited to 8-bit words
- Extremely simple hardware interfacing
- Simple software implementation

Disadvantages:

- Requires more pins on IC packages
- No hardware slave acknowledgment (the master could be transmitting to nowhere and not know it)
- No error-checking protocol is defined
- Only handles short distances compared to RS-232, RS-485, or CAN-bus.
- Some variants like dual SPI, quad SPI, and three-wire serial buses defined below are half-duplex.