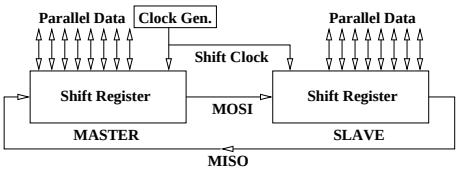
The Serial Peripheral Interface (SPI)

SPI is used widely for connecting peripherals to processors. It is a synchronous serial interface, like I^2C. SPI, like all serial interfaces, uses shift registers for converting parallel data to serial. As in the case of I^2C interface, there is the concept of a Master device and a slave device. The master device provides the clock to the slave device to shift data.

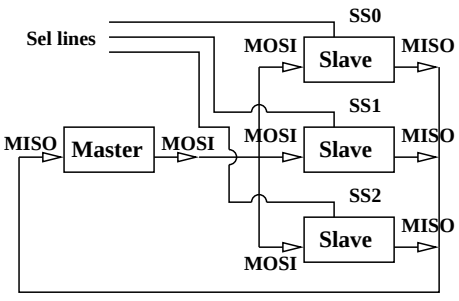
Let us first look at a simple version of SPI with a single master and a single slave.



The output of the master shift register is connected to the input of the slave shift register through a line called Master Out Slave In or MOSI. Similarly, the output of the slave shift register is connected to the input of the master shift register through a line called Master In Slave Out or MISO. The master also provides the shift clock to the slave. Thus, the simplest SPI interface uses 3 wires for communication. It is obvious that if 8 clock pulses are provided, The contents of the master shift register will end up in the slave shift register. *Simultaneously* the contents of the slave shift register will end up in the master shift register. Both Master and Slave can now read the data in parallel by reading their respective shift registers. Thus in SPI, roles of transmitter and receiver are not distinct. When 8 shift clocks are provided, the master and the slave have *exchanged* data. The only distinction between devices is the role of master or slave based on who provides the shift clock.

Obviously, the shift clock needs to have some logic to decide when to start shifting the data. Typically, the shift clock provides 8 clock pulses when a write occurs in the master shift register. If a master just wants to read data from the slave, it can do a dummy write to its SPI register (shift register) and then read back the contents after data exchange has occurred.

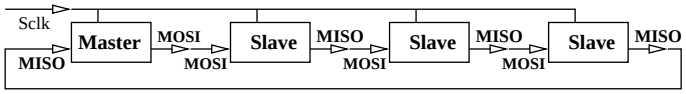
SPI uses one more wire compared to I^2C, but provides two way (duplex) communication in a single operation. How do we manage if we have multiple slave devices? There are two configurations for connecting multiple slave devices to a master.



In parallel configuration, all slaves have their MOSI and MISO lines in parallel. Slave devices have an extra input called slave select. The slave shift register shifts its data only if its slave select signal is asserted. At any time, the select signal of only one slave is activated. This results in a configuration similar to the single slave case.

The select signals are provided by the port pins of the master and a particular slave can be activated by asserting the corresponding port line. All devices have their clock inputs in parallel.

In series or daisy chained configuration, the MISO line of each slave drives the MOSI line of the next. The final MISO output loops back to the master device, forming a long shift register.



To send a byte from the master to the last device, several dummy writes will be required to shift the data out to the last device. This configuration does not use slave select signals. So it will not use up port pins but will reduce data transfer rate to the farther devices.

The master device is typically a microcontroller which provides a built in SPI module. The serial clock is generated by using a timer. Slave select lines, if required, can be generated from port pins.