第十讲
总线与接口
Lecture 10
Bus and Interface

明玉瑞 Yurui Ming

yrming@gmail.com

声明
Disclaimer

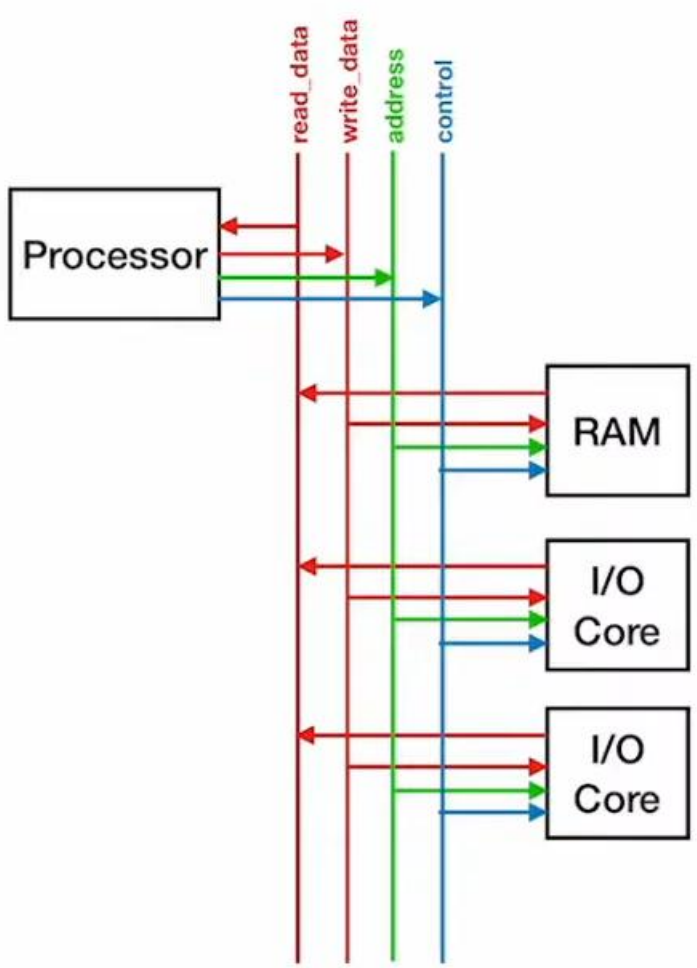
- 本讲义在准备过程中由于时间所限，所用材料来源并未规范标示引用来源。所引材料仅用于教学所用，作者无意侵犯原著者之知识版权，所引材料之知识版权均归原著者所有；若原著者介意之，请联系作者更正及删除。

  The time limit during the preparation of these slides incurs the situation that not all the sources of the used materials (texts or images) are properly referenced or clearly manifested. However, all materials in these slides are solely for teaching and the author is with no intention to infringe the copyright bestowed on the original authors or manufacturers. All credits go to corresponding IP holders. Please address the author for any concern for remedy including deletion.

总线
Bus

- CPU与其它模块或外设进行交互，通常是通过总线进行。一般与高速设备的交互直接称为通过总线，与低速设备的交互称为通过接口。总线包括数据总线、地址总线与控制总线。

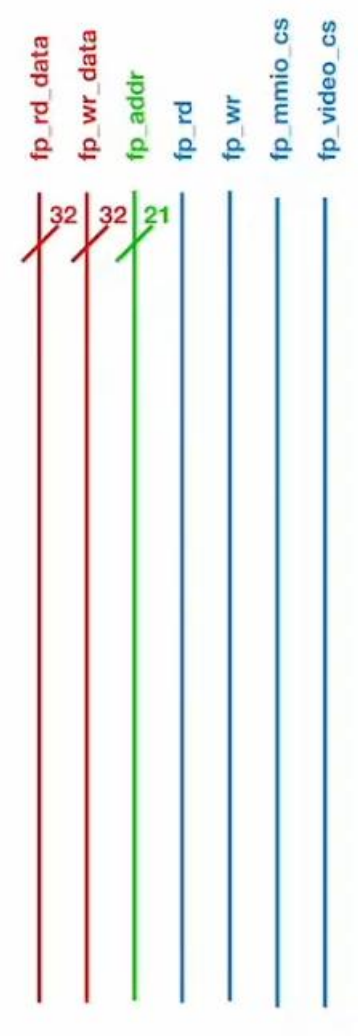
  The interactions between other modules or peripherals are via system buses. Usually, the mechanism of CPU and high-speed peripheral interaction is called system bus, while the mechanism of the interaction between CPU and low-speed peripherals is called interface. The types of buses include data bus, address bus and control bus.



总线
Bus

- 对于总线的图示，一般单根直线表示承载一位信息，如果总线上有一个斜线，表示承载多位信息。有时会将位数也标识上去。注意各类型总线的位数相互独立，通常情况下，CPU的位数与数据总线的位数一致。

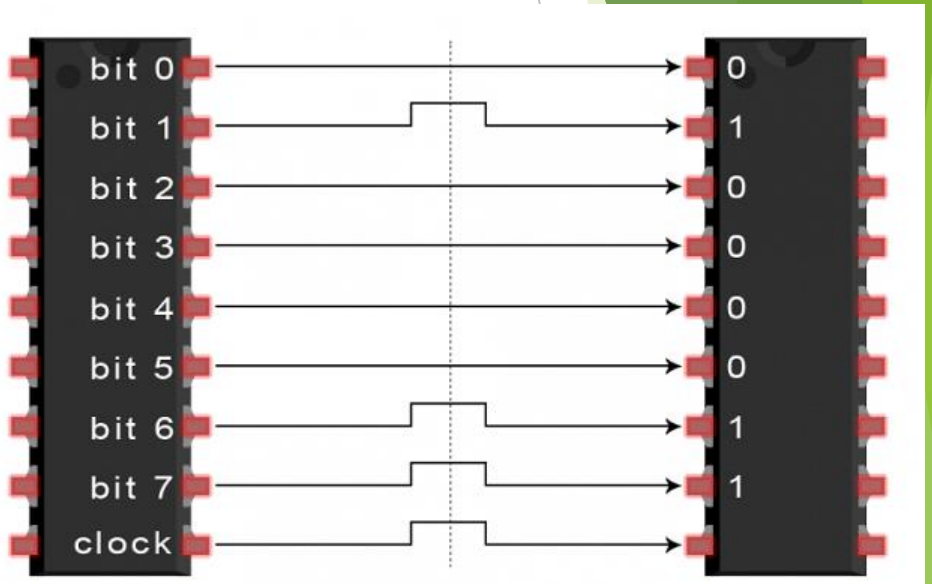
  For the bus diagram, a single line indicates it carries one bit information. A slash on the line indicates it carries multiple bits information. Sometimes, the bit width is also labelled in the diagram. Notably, the different types of buses are independent of each other. Generally, the width of the data bus is in accordance with the bits of the CPU.



总线
Bus

- 总线本身只是承载者电信号。与人们之间的通讯类似，通过总线连接的设备如果期望相互通讯，则必须遵循彼此能理解的规范或通讯协议，一般称为总线协议或接口协议。不同的通讯协议反过来可能约束总线本身，如总线的位宽，制造总线的物理介质、长度等。依据数据传输方式，总线又分为并行总线与串行总线。

  The bus itself only caries the electrical signal. Communication between electrical equipment is similar to human communication. Both parties connected by the bus expected to communicate with each other must comply with the same standard or communication protocols, which is also called bus protocols or interface protocols. It should be mentioned that the protocol might put constraint on the bus itself, such as the width of the bus, the materials made of the bus, the length of the bus it can run. According to the way of transferring the data, bus can be categorized as parallel bus and serial bus.



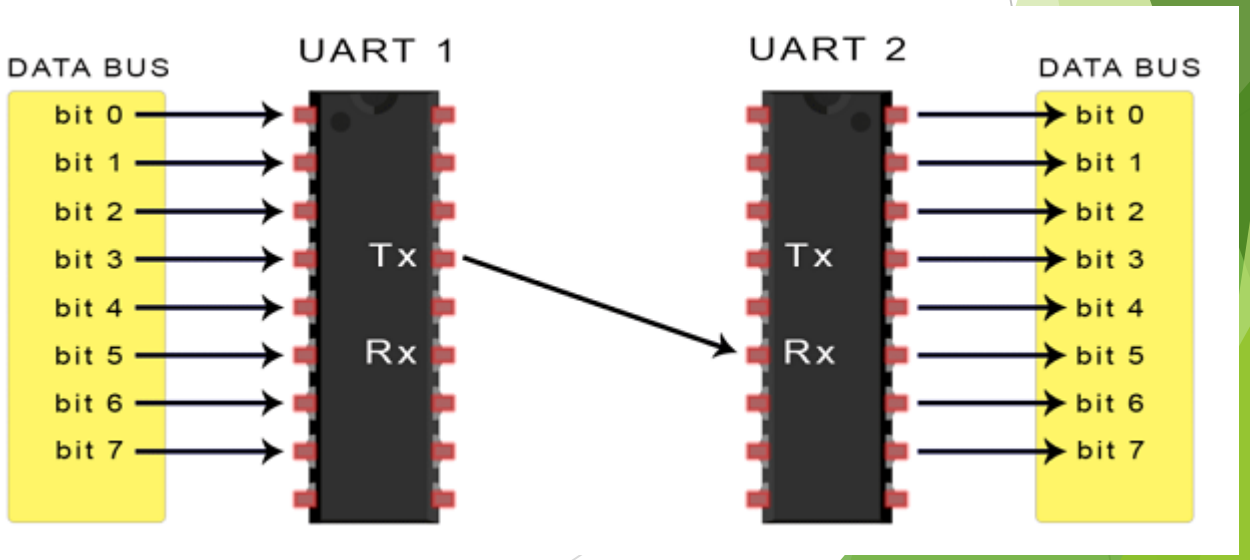
UART协议
UART Protocol

- 两个具有UART接口的设备可以通过其直接相互通信。发送端UART将来自控制设备（例如 CPU）的并行数据转换为串行数据，然后串行发送到接收端UART。数据从发送端UART的Tx引脚传输到接收UART的Rx引脚。

  Two UARTs communicate directly with each other in UART communication. The sending UART converts parallel data from the control device (such as the CPU) to serial data and sends it serially to the receiving UART. The data travels from the sending UART's Tx pin to the receiving UART's Rx pin:

- UART协议的全称叫通用异步收发协议。从名字我们就可以知道，UART是异步串行通信的，这意味着没有时钟信号。

  The acronym UART stands for Universal Asynchronous Receiver/Transmitter. From the name, we know UART is an asynchronous serial communication, which means there is no clock signal.



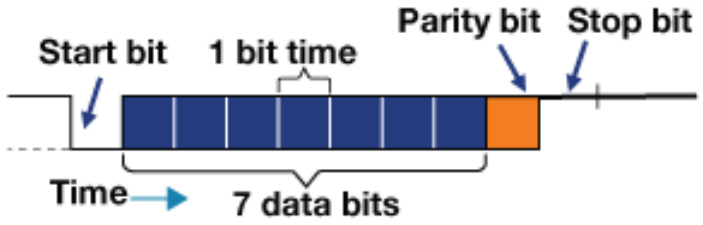
UART协议
UART Protocol

- 基于UART的数据传输速度(bps)以波特率（以每秒位数为单位）来度量。两个UART必须以大致相同的波特率运行，接收端UART在检测到起始位时会以给定的波特率读取起始位。由于两个设备之间没有共同的时钟信号，为了保证数据接收的准确性，发送和接收UART之间的最大波特率差异为10%。

  The baud rate, which is measured in bits per second, is used to measure the data transmission speed (bps) of UART. The two UARTs must operate at about the same baud rate, so the receiving UART will read the start bit at a certain baud rate when it detects it. Due to lack of common clock signal, to ensure the accuracy of data transmission, it only allows a maximum baud rate difference of 10% between the sending and receiving UARTs.

- 下图显示了UART的时序图。每个块代表一个位。通常传输以逻辑低开始。

  Figure 1 shows the timing diagram of a UART. Each block represents a single bit. Usually, data transmission begins with logic low.



UART协议
UART Protocol

- 实际上，要传输数据，发送器和接收器以下配置必须相同：

  To transmit data, both the transmitter and the receiver must agree with five common configurations.

波特率	衡量将要传输的数据的传输速度
Baud Speed	The transmission speed of how fast data is to be transmitted.
数据长度	接收器将保存到其寄存器中的数据的位数。
Data Length	The agreed number of bits that the receiver will save into its registers.
开始位	一个低电平信号，让接收器知道何时要传输数据。
Start Bit	A low signal that lets the receiver know when data is about to be transferred.
停止位	让接收器知道最后一位业已发送的高电平信号。
Stop Bit	A high signal that lets the receiver know when the last bit (most significant bit) has been sent.
校验位	用于检查发送的数据是否正确或损坏的校验信号。
Parity Bit	Either a high or low signal used for checking if data sent was correct or corrupted.

UART协议
UART Protocol

- 传输过程：

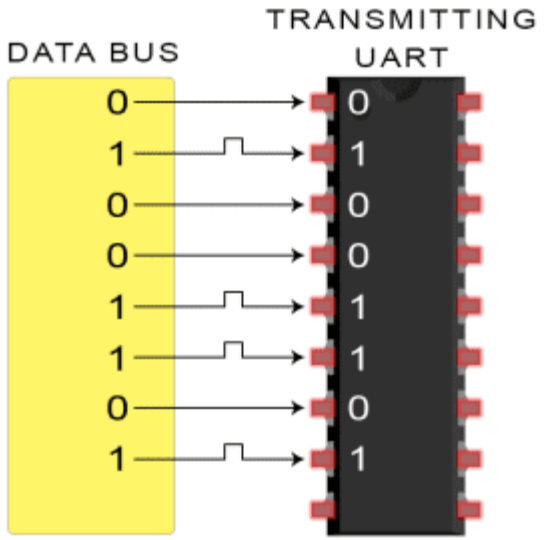
  1. 发送UART从数据总线并行接收数据；

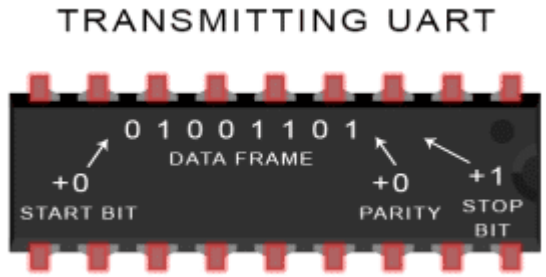
  2. 发送端 UART 将起始位、奇偶校验位和停止位添加到数据帧中；

- Transmission steps：

  1. The sending UART receives data from the data bus in parallel;

  2. The starting bit, parity bit, and stop bit are added to the data frame by the sender UART:





UART协议
UART Protocol

- 传输过程：

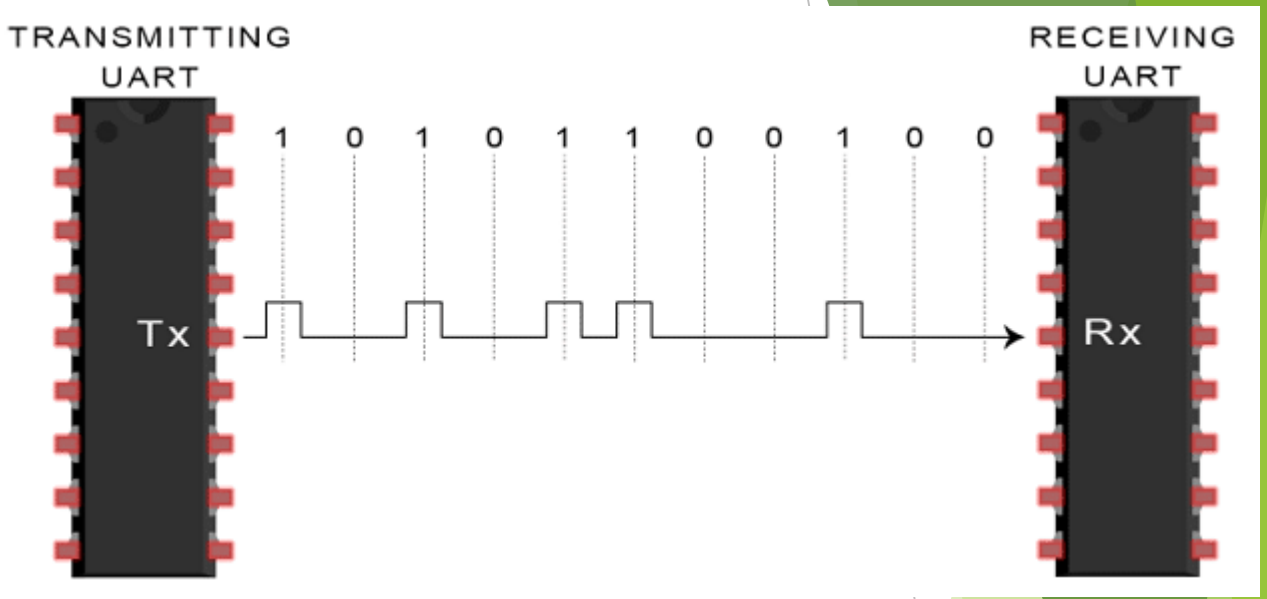
  3. 完整的数据包被从发送端UART串行传输到接收端UART。接收端UART以预先配置的波特率对数据线进行采样；

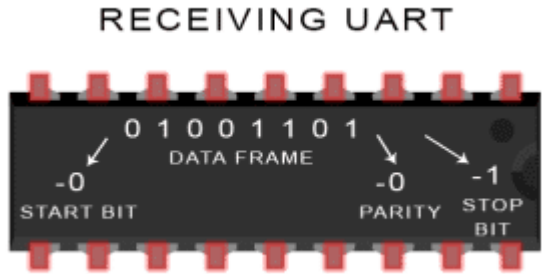
  4. 接收端UART丢弃数据帧的起始位、奇偶校验位和停止位，并将串行数据转换为并行数据，发送到数据总线

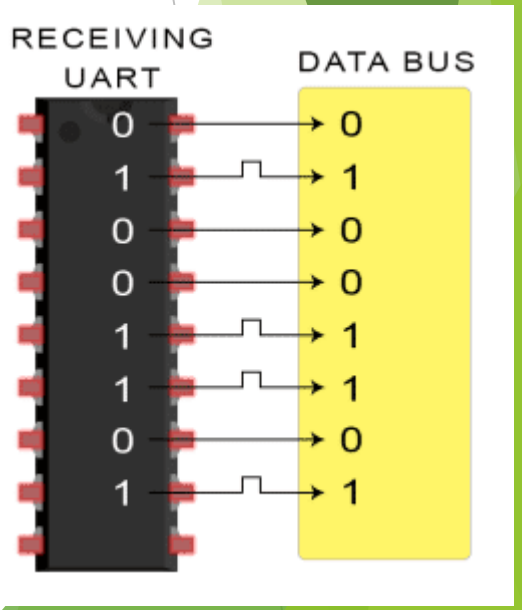
- Transmission steps：

  3. From the sending UART to the receiving UART, the full data packet is transferred serially. The data line is sampled at a pre-configured baud rate by the receiving UART;

  4. The receiving UART discards the data frame‘s start, parity, and stop bits and translates the serial data to parallel data and sends it to the data bus



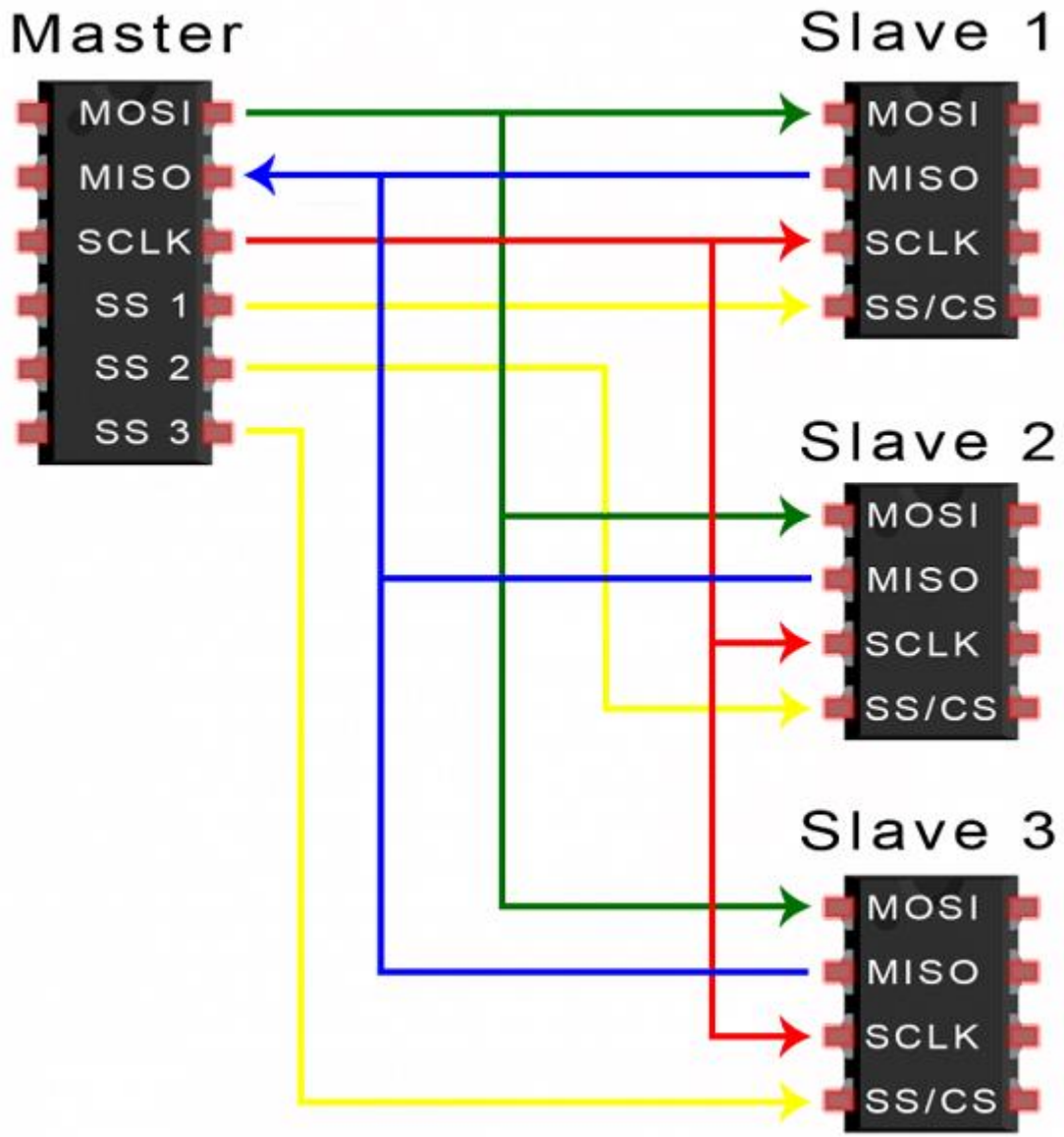




SPI协议
SPI Protocol

- SPI是一种广泛使用的设备通信协议，使用SPI协议进行通信的设备分为两部分：主机（或主设备/系统）和从机（或从设备/系统）。主机（通常是微控制器）是控制设备，而从机（通常是传感器、显示器或存储芯片）接收来自主机的指令。一套SPI通讯有四根信号线：主输出/从输入、主输入/从输出、时钟信号SCLK，片选信号SS/CS。

  SPI is a widely used device communication protocol. The SPI device is split into two parts: a host and a slave system. The host (typically a microcontroller) is the control device, while the slave (commonly a sensor, display, or memory chip) receives instructions from the host. There are four signal lines in a set of SPI communication: MOSI (Master Output/Slave Input), MISO (Master Input/Slave Output), Clock signal SCLK (Clock), Chip select signal SS/CS (Slave Select/Chip Select).



SPI协议
SPI Protocol

- 由于设备共享时钟信号，SPI是一种同步通信机制。但由于每个时钟周期只传输一位数据，所以基于SPI协议的数据传输速度取决于时钟信号的频率。因为时钟信号是由主设备设置生成，SPI通信始终由主设备开始。

  Since in SPI devices share clock signals, SPI is a synchronous communication mechanism. Because each clock cycle transmits one bit of data, the data transmission speed is determined by the clock signal's frequency. The SPI communication is always begun by the host because the clock signal is generated by the host setup.

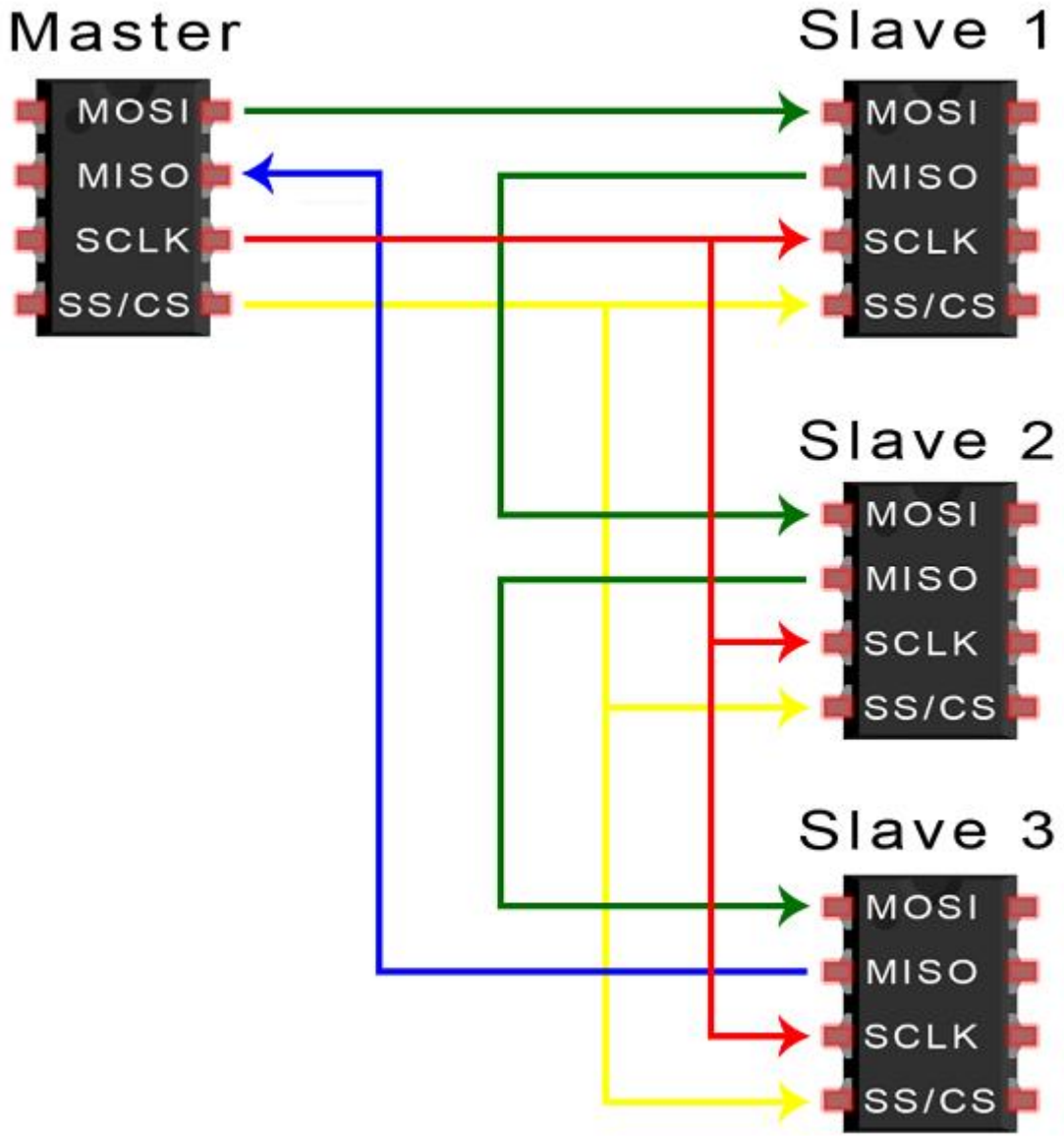
- SPI理论上是一种全双工通信协议，主设备通过MOSI以串行方式向从机发送数据，而从设备可以通过 MISO 向主设备发送数据，两者可以同时进行。

  SPI is a full-duplex communication protocol in theory. The master sends data to the slave in a serial fashion via MOSI, while the slave can send data to the master via MISO, and the two can be done simultaneously.

SPI协议
SPI Protocol

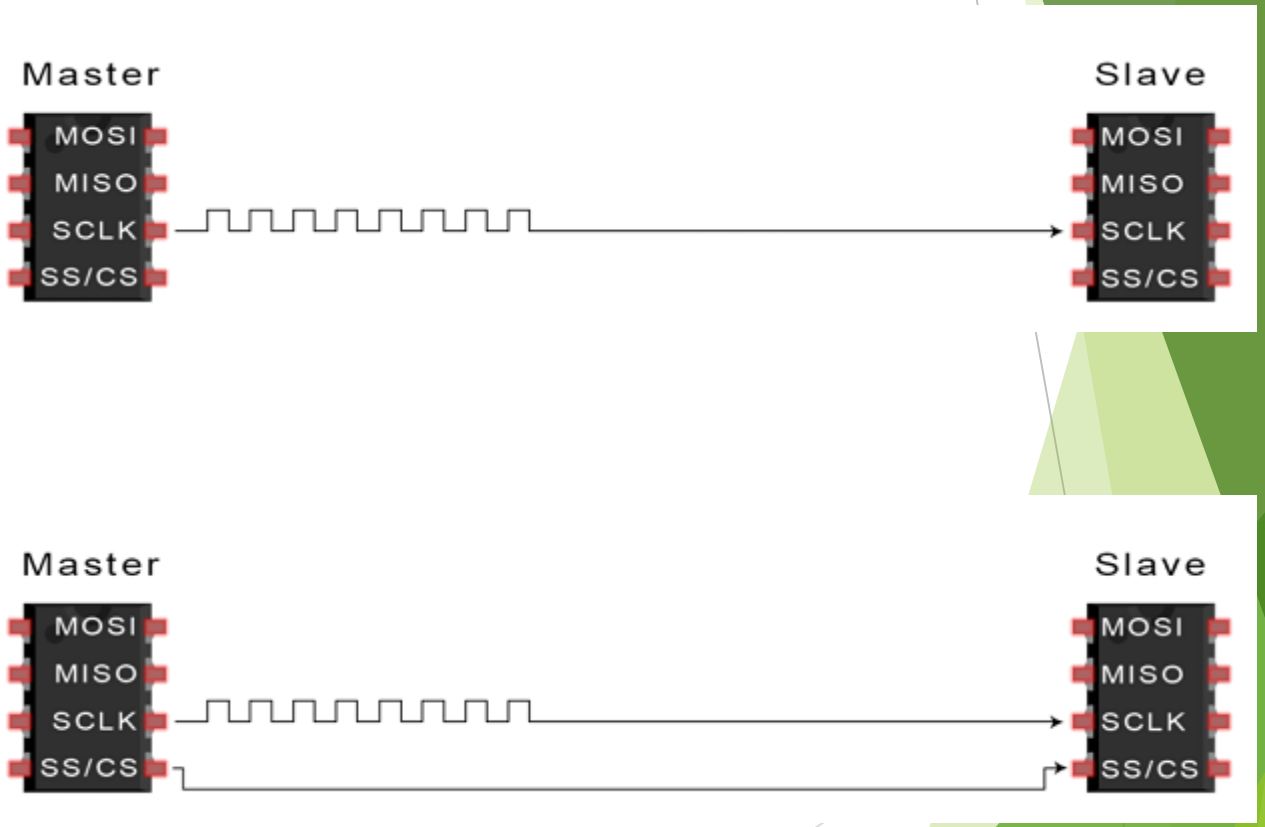
- 当主设备设备有多个片选引脚时，可以挂多个从设备。此时片选引脚可以只有一位。但当主设备只有一个片选引脚时，同样可以挂多个从设备，但要求片选引脚有足够的位宽。

  When the host device has multiple chip select pins, multiple slave devices can be connected. At this time, the chip select pin can have only one bit. However, when the host device has only one chip select pin, multiple devices can also be connected, but the chip select pin is required to have sufficient bit width.



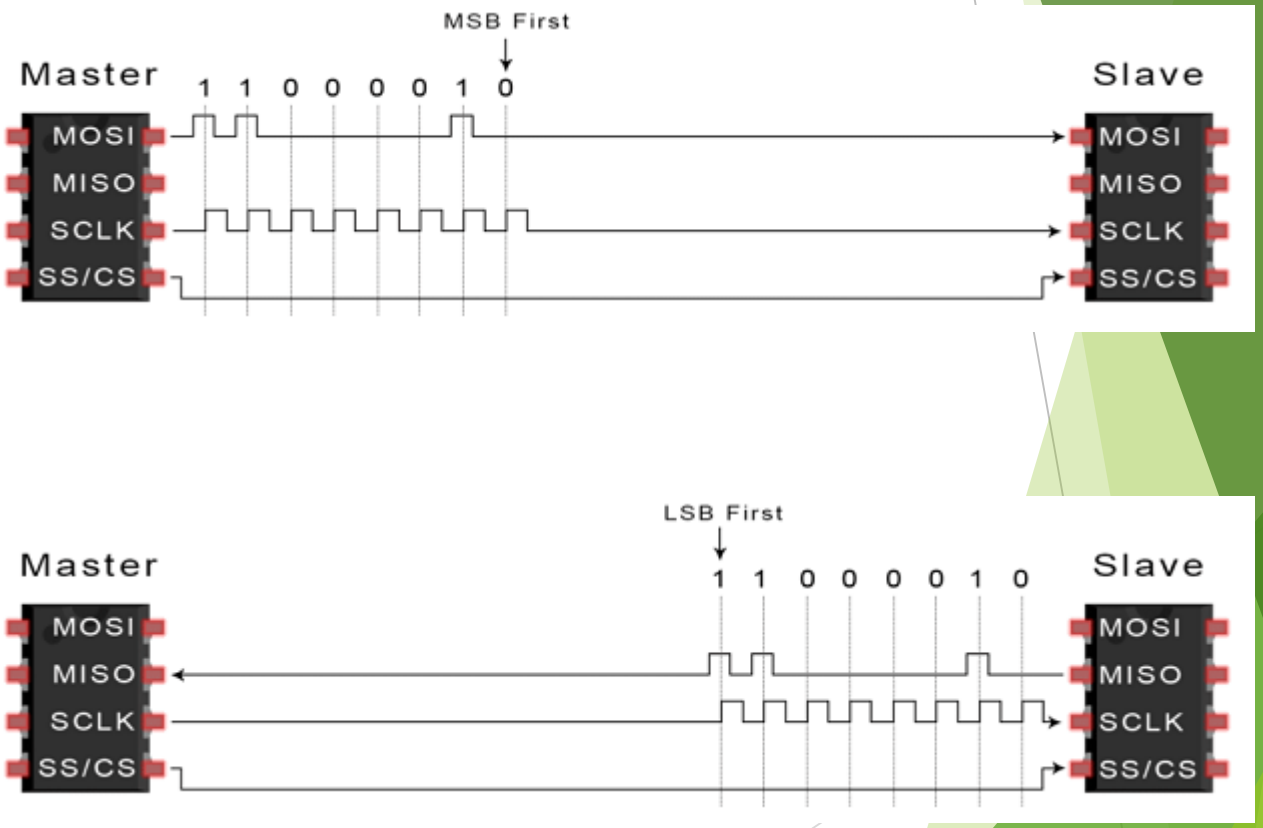
SPI协议
SPI Protocol

- 传输步骤
  1. 主设备发起时钟信号；
  2. 主设备拉低SS/CS引脚从而激活从设备；
- Steps in the transmission
  1. The clock signal from the host
  2. To activate the slave, the master pushes down the SS / CS pin



SPI协议
SPI Protocol

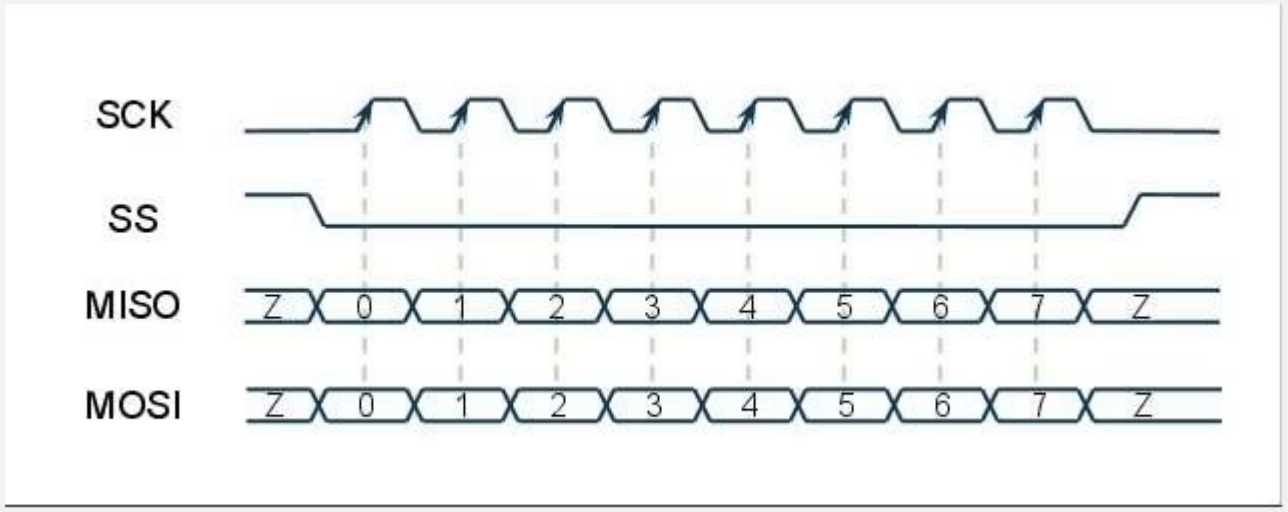
- 传输步骤
  3. 从设备通过MOSI接收主设备的数据；
  4. 如果需要响应，从设备将通过MISO 将数据返回给主设备。
- Steps in the transmission
  3. The slave receives data from the master through MOSI
  4. The slave will return the data to the master via MISO if a response is necessary.



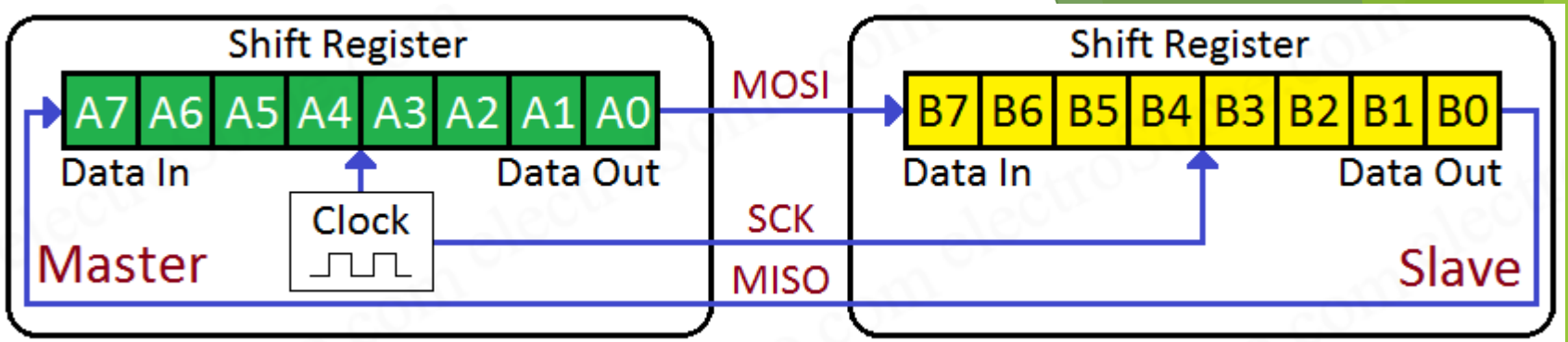
SPI协议
SPI Protocol

- 总的来说，当MISO或MOSI准备就序时，置低对应从设备的片选信号管脚，此时主设备与从设备在时钟信号的上升沿，对数据信号MISO与MOSI进行采样，其电路时序如下图所示：

  Generally, when the data is put into MISO or MOSI and ready to transmit, the low activated SS which corresponds the slave device is enabled. The data on MISO and MOSI are sampled at the rising edge of the clock signal. The overall sequential diagram is as follows:



SPI协议
SPI Protocol



- 注意，只所以说在时钟信号的上升沿，对数据信号MISO与MOSI进行采样，是因为主设备和从设备的实现都包含一个串行移位寄存器。主机通过将字节写入其SPI移位寄存器来开始传输字节。当寄存器通过MOSI信号线将字节传输到从设备时，从设备将其移位寄存器的内容通过MISO信号线传输回主设备。如果只需要写操作，主设备会忽略它接收到的字节。相反，如果主设备只想从从设备读取一个字节，它必须传输一个空字节以启动从设备传输，即主从设备都同时执行写入和读取操作。

Notably, we say the data on MISO and MOSI are sampled at the rising edge of the clock signal simultaneously, it is because both masters and slaves are implemented to contain a serial shift register. The master starts a transfer of a byte by writing it to its SPI shift register. As the register transmits the byte to the slave on the MOSI signal line, the slave transfers the contents of its shift register back to the master on the MISO signal line. If only a write operation is desired, the master just ignores the byte it receives. Conversely, if the master just wishes to read a byte from the slave, it must transfer a dummy byte in order to initiate a slave transmission.

I²C协议
I²C Protocol

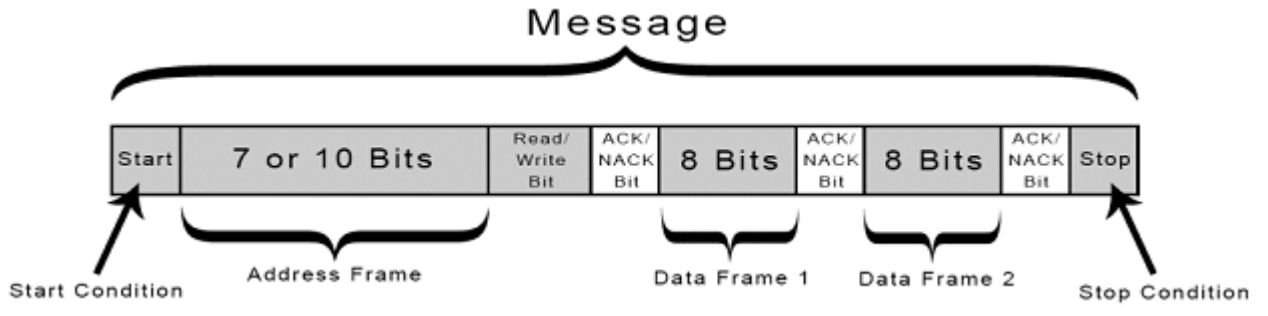
- I²C总线是由飞利浦发明的，是一种简单的双向同步串行总线，只需要两根电缆进行数据发送。其将SPI和UART的优点结合在一个设备中，多个从设备可以连接到单个主设备（例如 SPI），或者多个主设备可以管理一个或多个从设备。当希望多个微控制器将数据记录到单个存储卡或在单个 LCD 上显示文本时，这是一个不错的选择。

  Philips invented the I2C bus, which is a simple bidirectional two-wire synchronous serial bus. Only two cables are required to send data. It combines the benefits of SPI and UART in a single device. Numerous slaves can be connected to a single master (such as SPI), or multiple masters can govern one or more slaves. When numerous microcontrollers need to log data to a single memory card or display text on a single LCD, this is a good option.

- I2C 协议中的两个引脚是发送和接收数据的 SDA（串行数据线）和用作时钟的 SCL（串行时钟线）引脚。

  The two pins in an I2C protocol are the SDA (Serial Data Line) which transmits and receives data, and the SCL (Serial Clock Line) pin, which functions as a clock.

I²C协议
I²C Protocol



- I²C数据以消息的形式传输，每个消息包含从设备的二进制地址帧和一个或多个数据帧，其中包括开始和停止条件、读/写位和数据帧。

  I²C data is transmitted in the form of numerous messages, each of which contains the slave's binary address frame and one or more data frames, which include start and stop conditions, read/write bits, and data frames.

- 因为I²C缺少像SPI这样的片选线，因此每个I²C从设备都有一个唯一的7位或10位序列，用于识别主设备和从设备。主机为每个从机提供从机地址进行通信，每个从机将其与自己的地址进行比较。如果地址匹配，它将向主机发送一个低级ACK位。如果它们不匹配，则不采取任何措施，并且SDA线保持高电平。

  Because I2C lacks a chip choose line like SPI, so each slave device has a unique 7-bit or 10-bit sequence that is used to identify the master and slave devices. The master provides each slave the slave address to communicate, and each slave compares it to its own address. It will transmit a low-level ACK bit to the host if the addresses match. If they don't match, no action is taken, and the SDA line remains high.

I²C协议
I²C Protocol

- 读/写位：如果主设备向从设备提供数据，则为低电平，如果从设备正在请求数据，则为高电平。

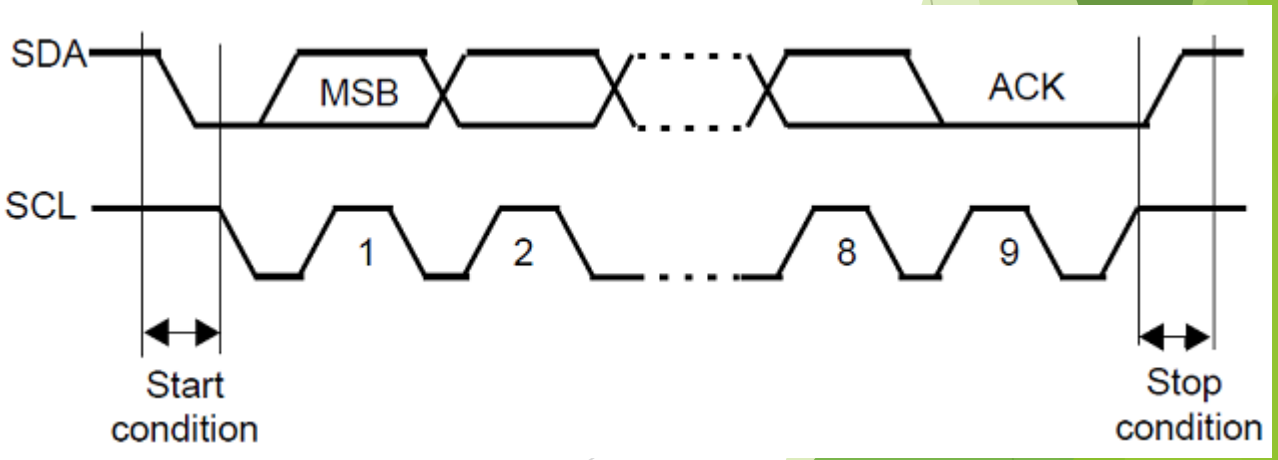
  Read/write bit: which is low level if the master is providing data to the slave and high level if the slave is requesting data.

- ACK/NACK：在消息中的每一帧之后都有一个ACK/NACK位。接收设备将返回一个ACK位，表示地址帧或数据帧已成功接收。

  ACK/NACK: An ACK/NACK bit follows each frame in the message. The receiving device will return an ACK bit to indicate that the address frame or data frame was successfully received.

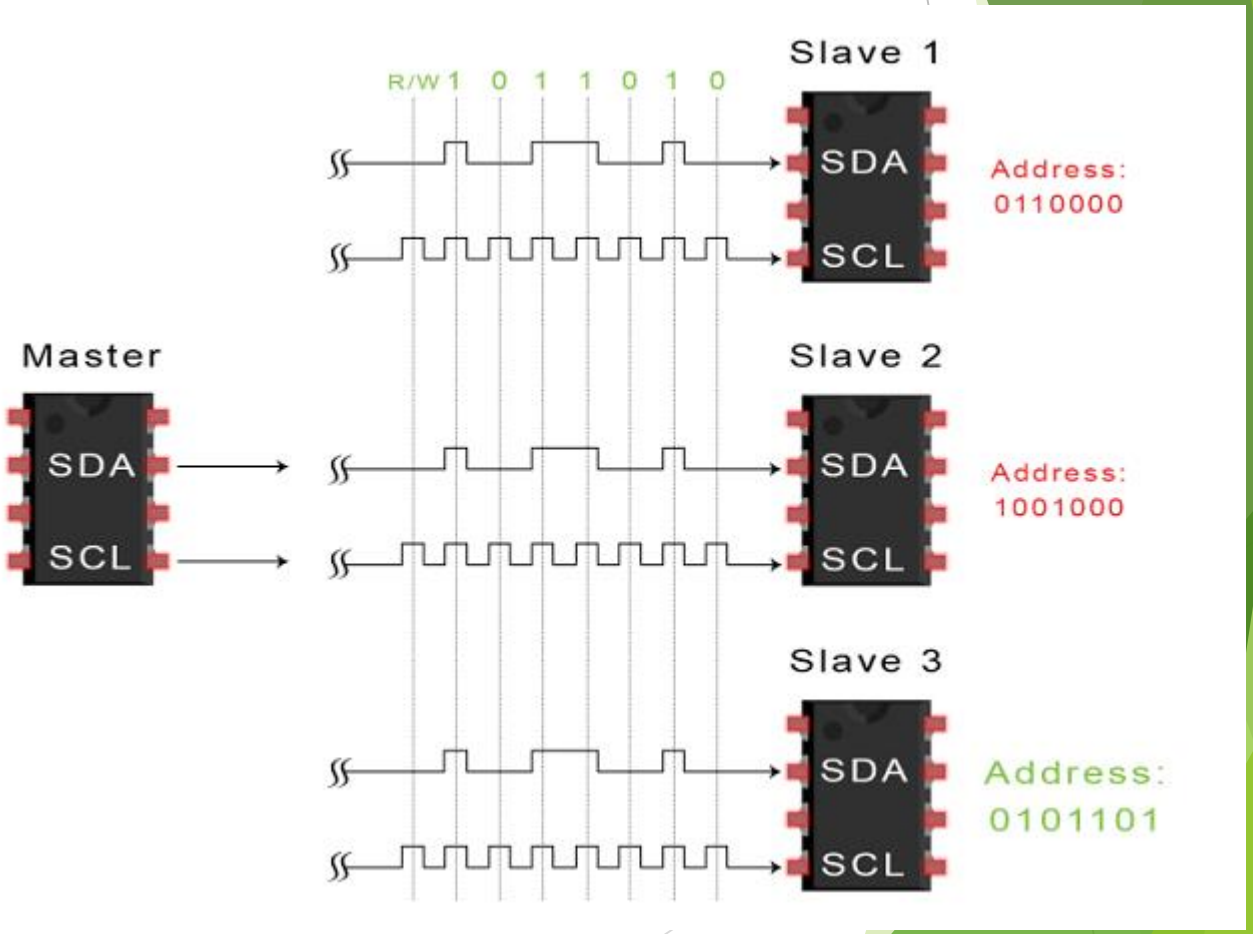
- 启动条件：当SCL为高电平时，SDA由高电平切换为低电平。停止条件：当SCL为高电平时，SDA由低电平切换为高电平。

  Start condition: When SCL is high, SDA switches from high to low. Stop condition: When SCL is high, SDA switches from low to high.



I²C协议
I²C Protocol

- 传输步骤
  1. 当SCL线为高电平时，主机通过将SDA线从高电平切换到低电平来启动总线通信；
  2. 主机将要与之通信的从机的读/写位和7位或10位地址传送到总线；
- Steps in the transmission
  1. The host initiates bus communication by switching the SDA line from high to low when the SCL line is high.
  2. The host delivers the read/write bit and the 7-bit or 10-bit address of the slave to be communicated with to the bus



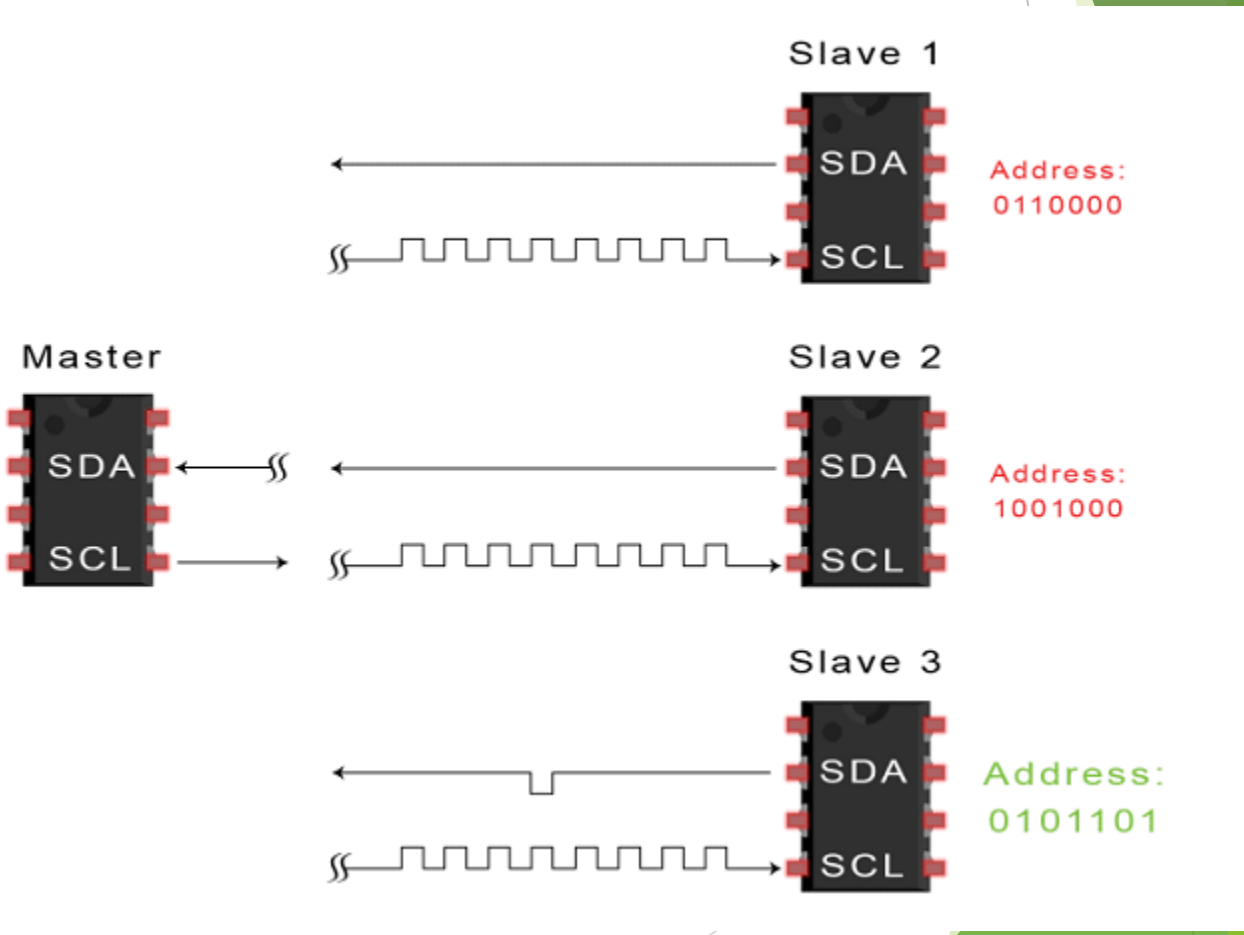
I²C协议
I²C Protocol

- 传输步骤

3. 每个从设备都会根据自己的地址检查主设备的地址。如果地址匹配，从设备通过将SDA线拉低来发送ACK位。如果主设备地址与从设备地址不匹配，从设备将SDA线推高。

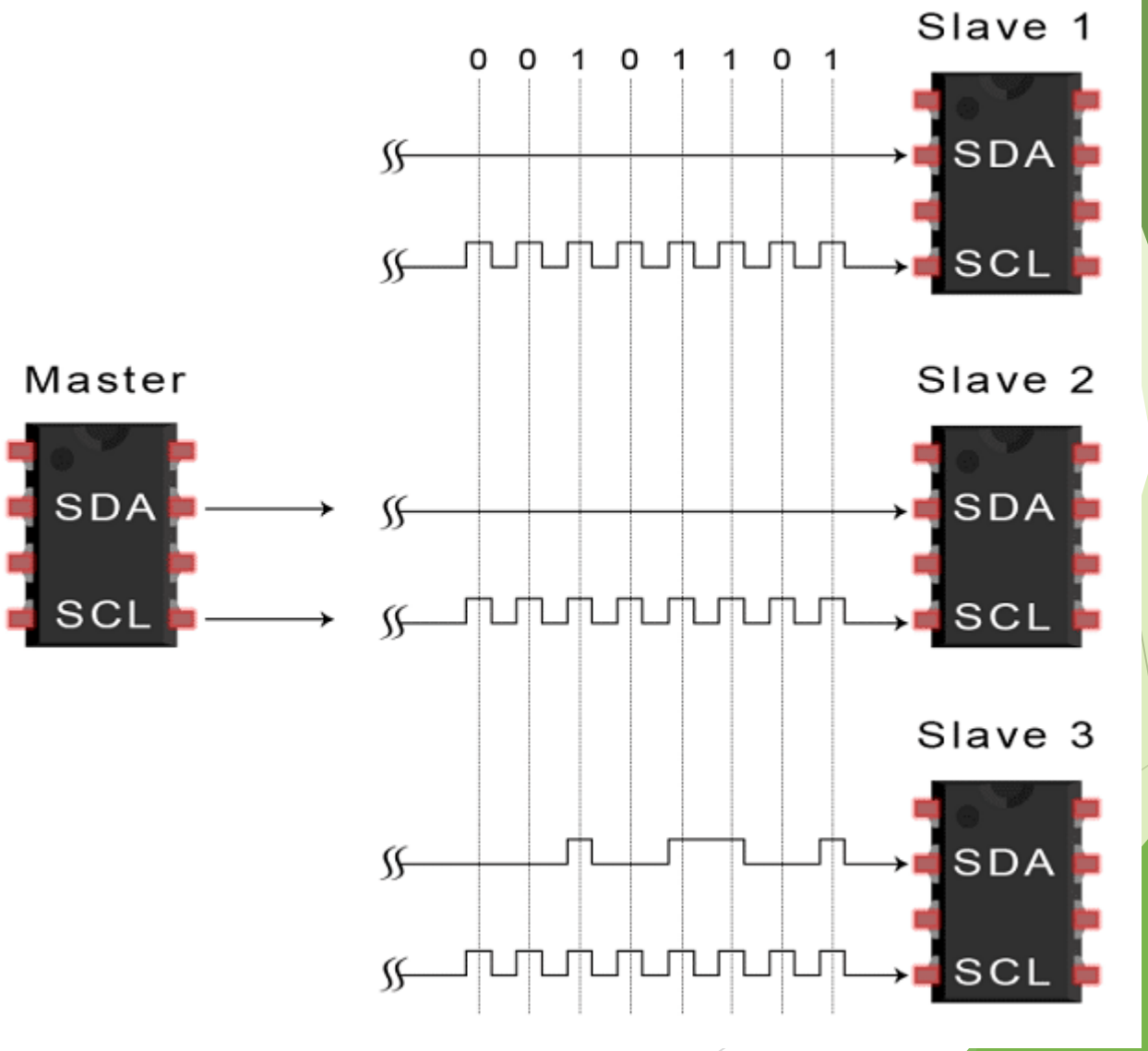
- Steps in the transmission

3. Each slave checks the master's address against its own address. The slave sends an ACK bit by pulling the SDA line low if the addresses match. The slave pushes the SDA line high if the master's address does not match the slave's address.



I²C协议
I²C Protocol

- 传输步骤
  4. 主机发送或接收数据帧。
- Steps in the transmission
  4. Data frames are sent or received by the host.



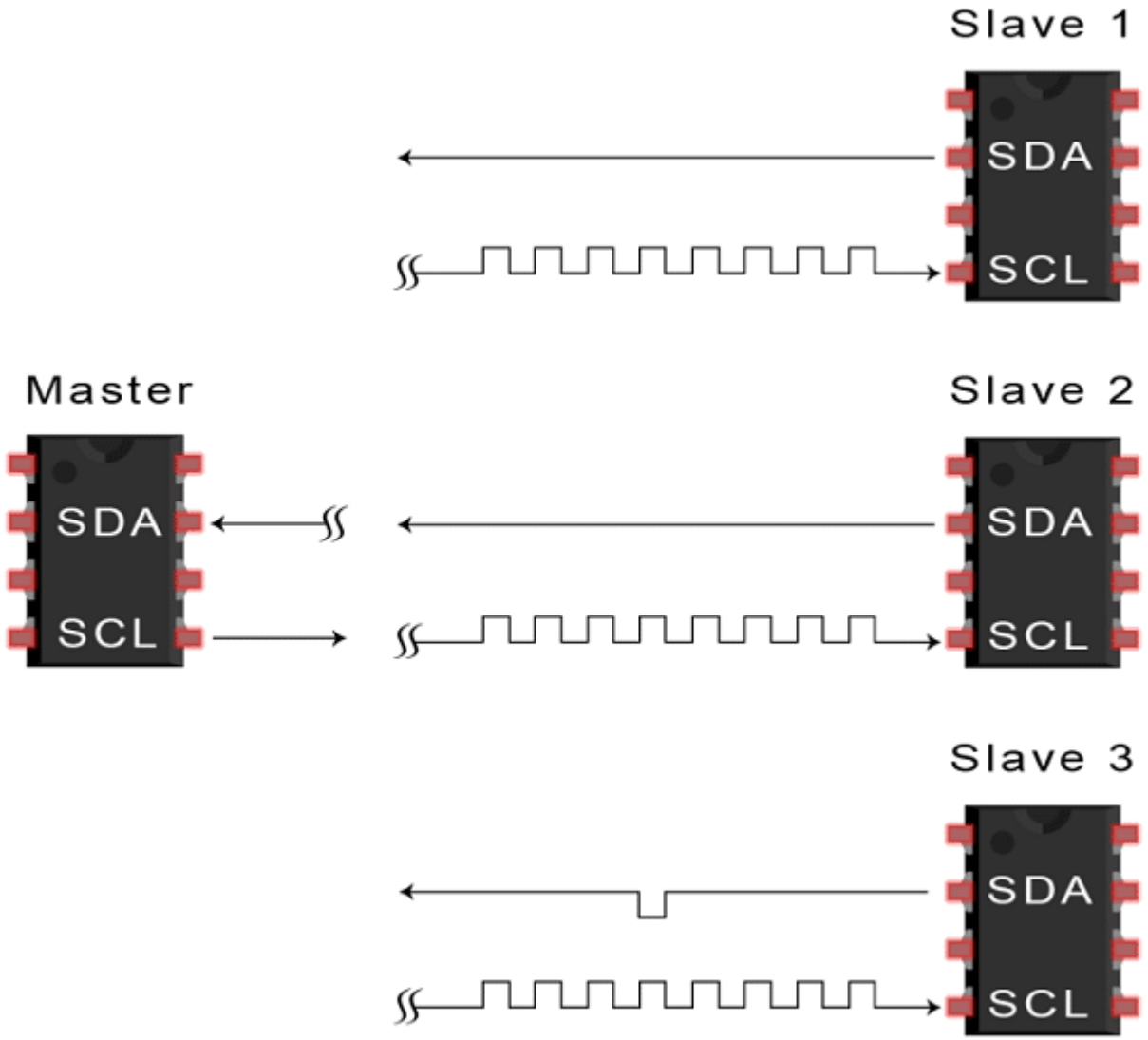
I²C协议
I²C Protocol

- 传输步骤

5. 接收设备在每一个数据帧传输完毕后，再向发送方发送一个ACK位，以保证该帧被成功接收；
6. 然后主设备将SCL切换为高电平，随后将 SDA切换为高电平，向从设备发送停止条件。

- Steps in the transmission

4. The receiving device sends another ACK bit to the sender after each data frame is transmitted to ensure that the frame was successfully received.
5. The master then switches SCL to high level and subsequently SDA to high level, sending a stop condition to the slave.



I²C协议
I²C Protocol

- I²C协议的一般情况下的时序图如下所示。注意，由于SDA在SCL处于高电平时被采样，因此，当SCL线的时钟信号为高电平时，SDA线的数据必须保持稳定，只有当SCL线的时钟信号为低电平时，SDA线的电平状态才允许改变。

  The timing diagram of I²C is shown as below. Notably, due the data on SDA is sampled when SCL is at high level, therefore, it's noted that data in SDA line must keep stable when clock signal in SCL line is at high level, and level state in SDA line is only allowed to change when clock signal in SCL line is at low level.