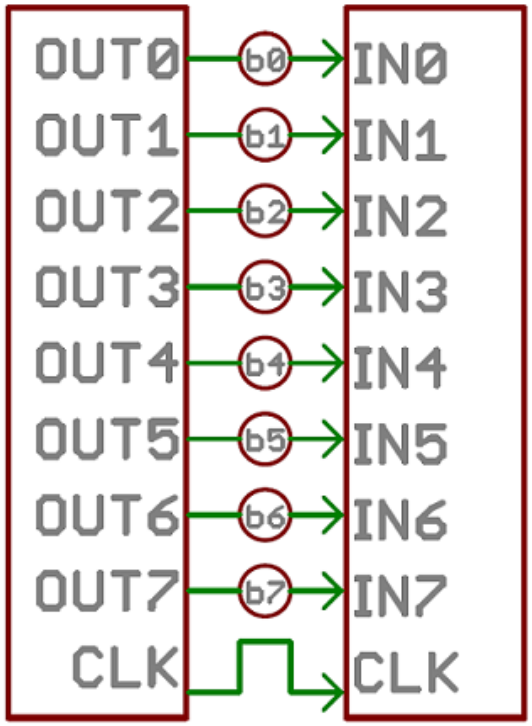
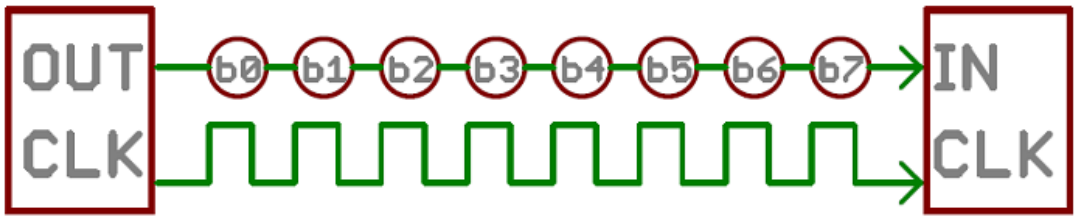
Serial Communication

Parallel vs. Serial Interfaces





- **Parallel**
  - Data bus (many pins)
  - Clock (source-synchronous or clock-forwarding)
- **Serial**
  - Single data wire (one pin)
  - Clock (source-synchronous or clock-forwarding)

Asynchronous Serial

- **Synchronous Serial**
  - Data lines associated with a clock signal
  - All devices on a serial bus share a common clock
  - Simpler and often faster interface
  - Requires extra wire(s) and pin(s)
  - Examples: SPI, I^2C
- **Asynchronous Serial**
  - No associated clock
  - Protocol must enforce additional signal transitions to facilitate data transfer, i.e. synchronization bits and optional parity bits in addition to data bits

Data Frame and Baud Rate

- **Frames created by appending bits to data**
  - Start and stop bit(s) indicate beginning and end of frame, performing synchronization function of a clock
  - Parity provides simple error detection (optional)
  - Protocol must specify endianness of data (LSB or MSB first)
- **Baud rate = speed of data transmission**
  - Ex: 9600 bits per second (bps)

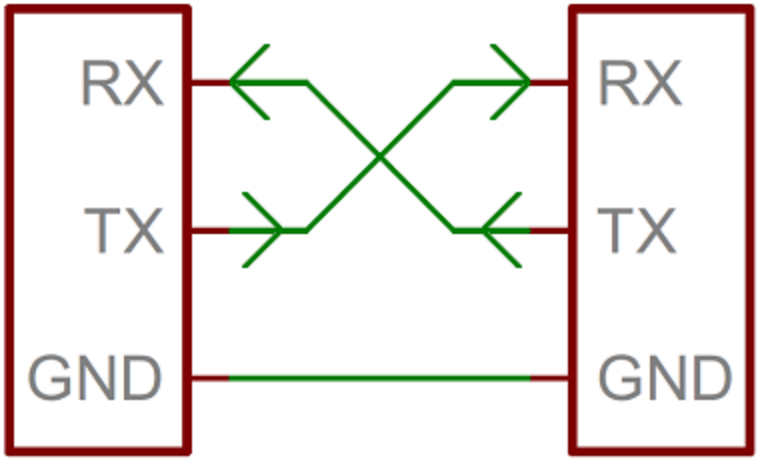
Data Frame Example

- **9600 8N1 Protocol**
  - Baud rate **9600** bps
  - **8** data bits
  - **N**o parity
  - **1** stop bit

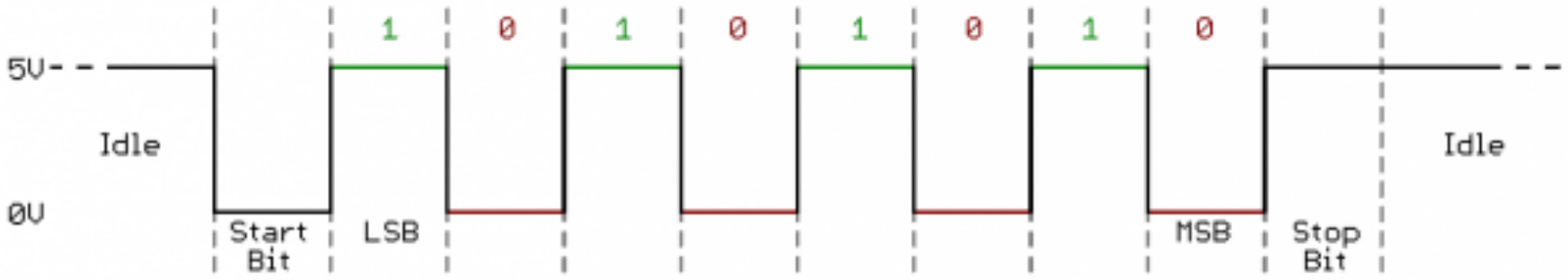
- **ASCII Characters ‘O’ and ‘K’**

Hardware Implementations

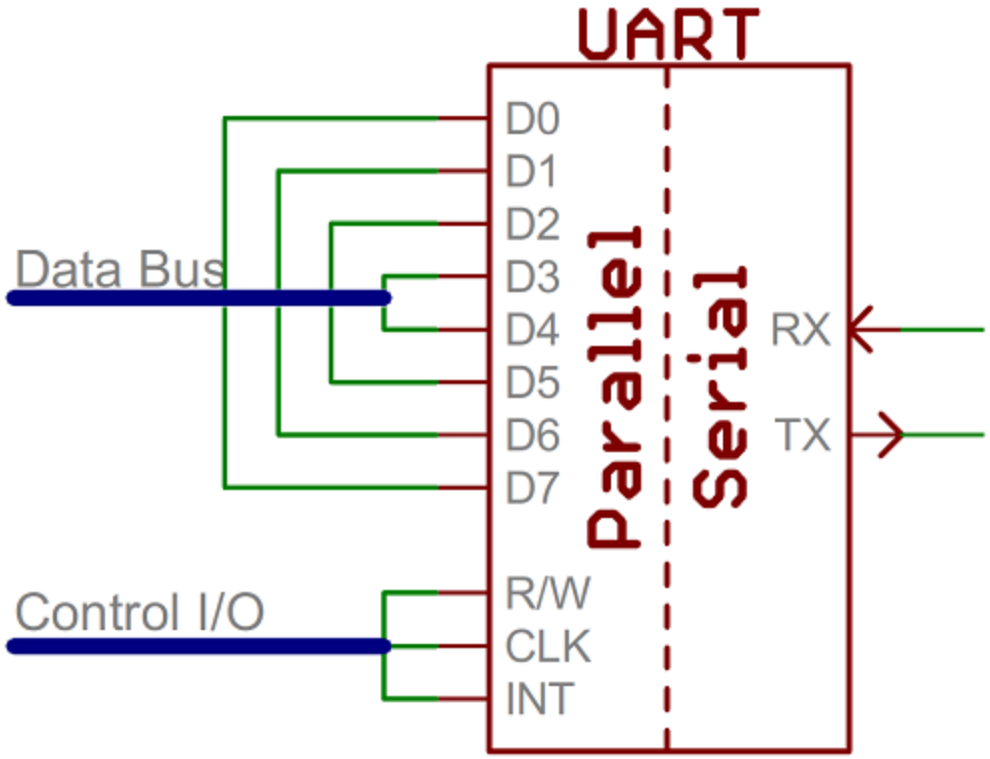
- **Unidirectional wires**
  - Simplex: only one connection
  - Half-duplex: only one device can transmit at a time
  - Full-duplex: both devices can transmit simultaneously



- **TTL (Transistor-Transistor Logic) Serial**



UART

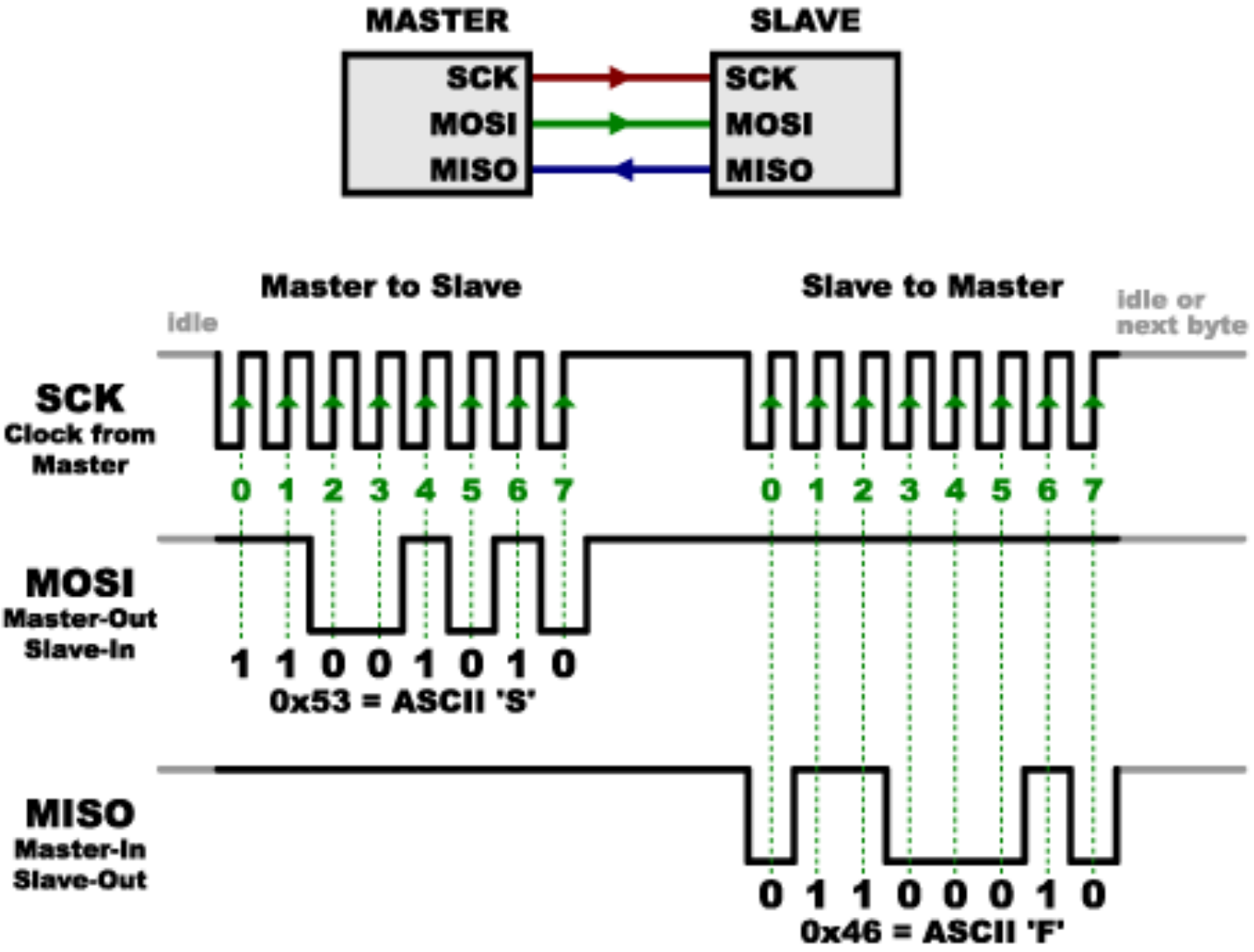


- **Universal Asynchronous Receiver/Transmitter**
  - Hardware block with parallel and serial interfaces plus control
  - Can be separate IC or integrated with microcontroller
  - Simplified UART interface shown
  - Software UART: CPU directly controls serial interface (bit-banging)

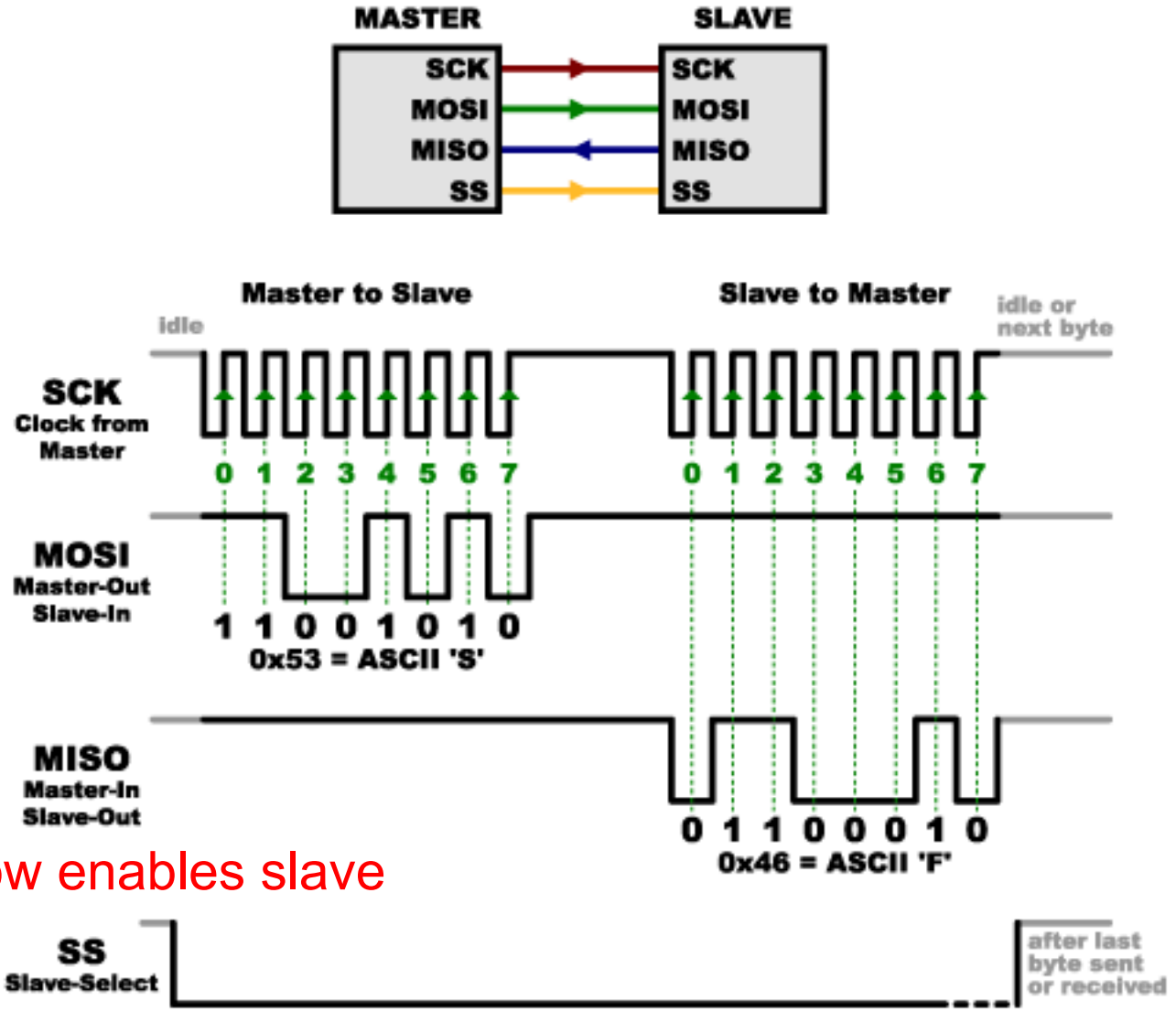
Serial Peripheral Interface (SPI)

- **Synchronous serial interface**
  - Only one device (master) generates the clock (usually called CLK or SCK)
  - Other devices are slaves
  - Always only one master (usually the microcontroller)
  - Can be multiple slaves
- **Data lines are unidirectional**
  - Master Out / Slave In (MOSI): from master to slave
  - Master In / Slave Out (MISO): from slave to master
  - Master generates prearranged number of clock cycles for slave to transmit data back, therefore master must know how many data bits to expect and when they will be valid

Simple SPI Example

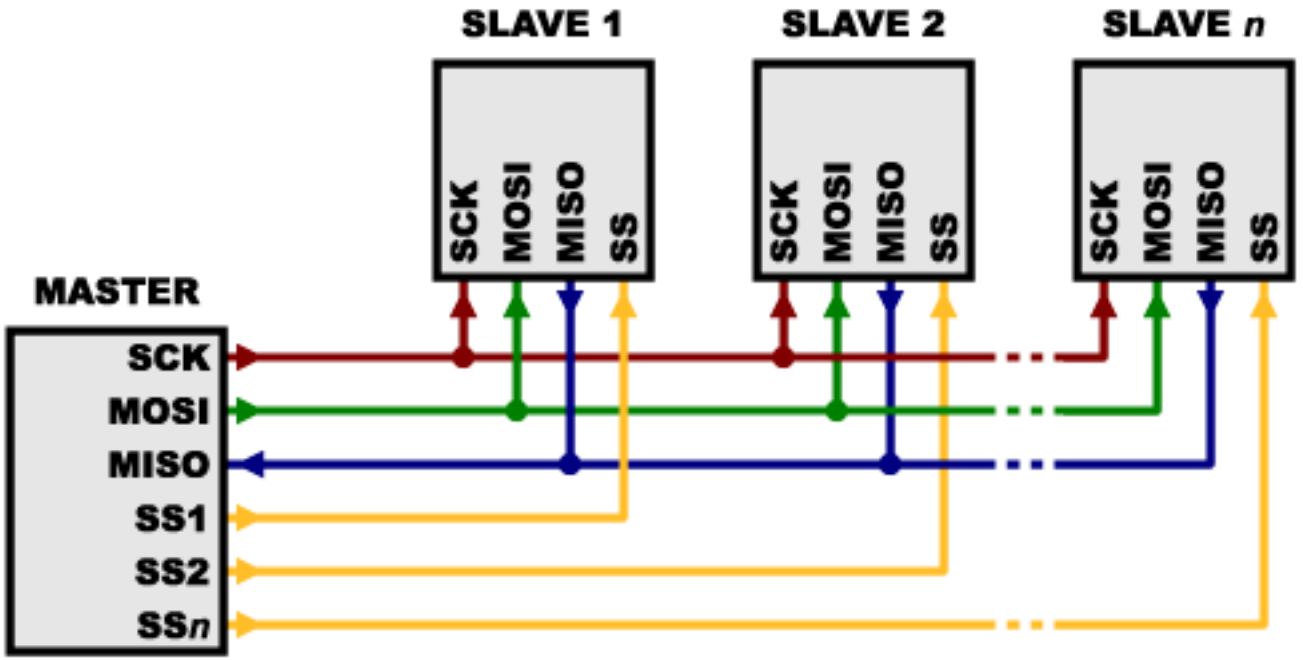


SPI Example with Slave Select



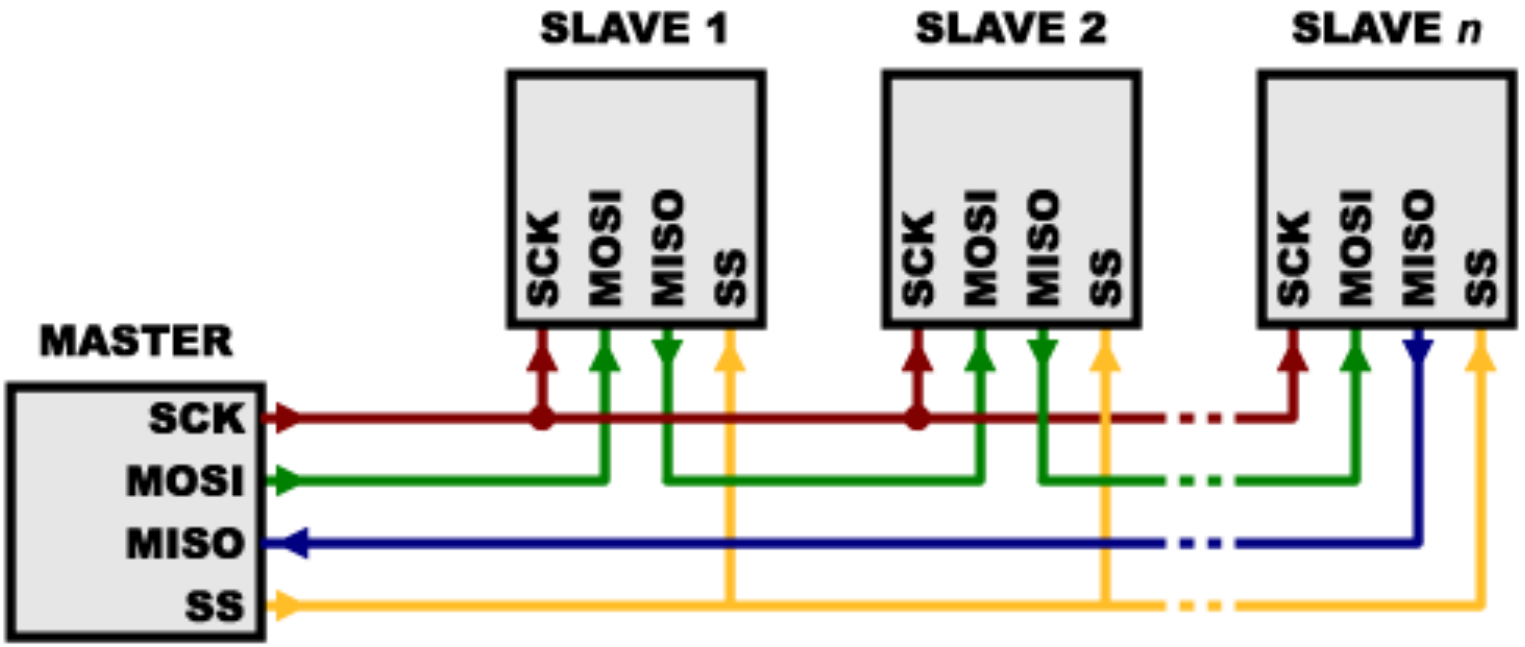
Active low enables slave

Multiple Slaves: Separate Slave Selects



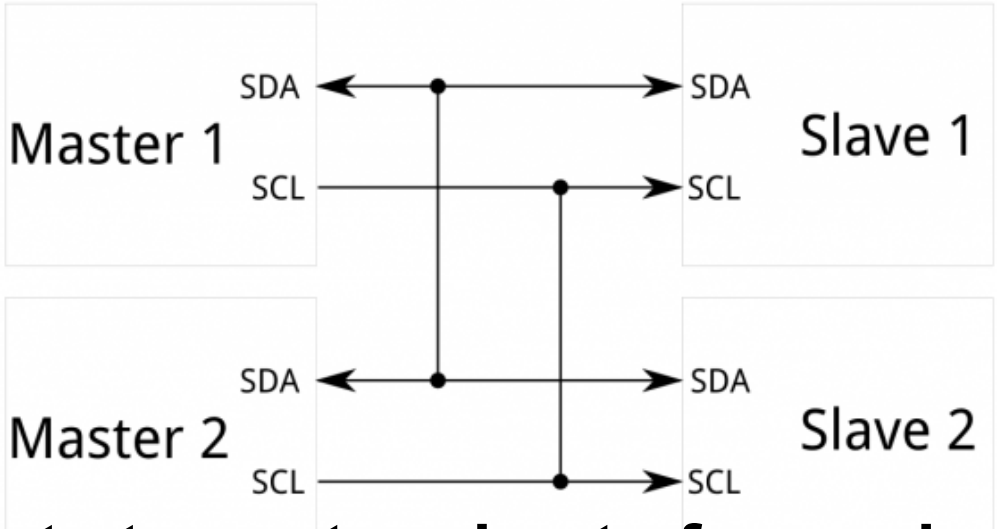
- **Master enables each slave by asserting unique slave select (SS) line**
  - High number of slaves requires many output pins on master or an external binary decoder chip

Multiple Slaves: Daisy Chain



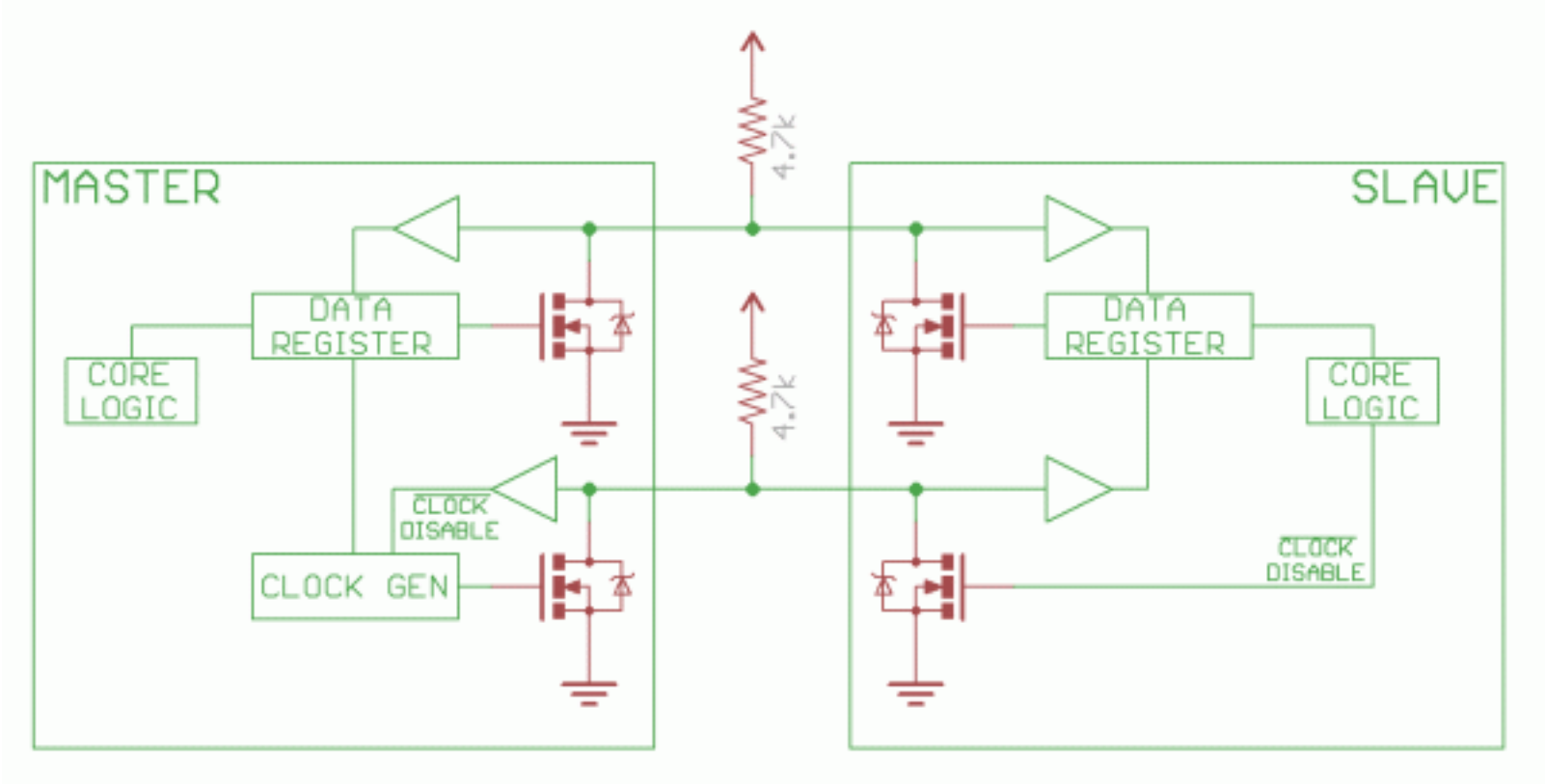
- **Some parts support daisy-chaining (MISO of one slave connects to MOSI of next slave)**
  - Slave select activates all slaves simultaneously
  - Data traverses all slave chips
  - Ex: addressable LED drivers

Inter-Integrated Circuit (I^2C) Protocol



- **I2C attempts to capture best of asynchronous serial and SPI**
  - Two wire interface (SDA/SCL) that can support up to 1008 slave devices
  - Multi-master systems are supported (with some limitations)
  - Data communication at 100 kHz or 400 kHz
  - 1 bit of meta data (ACK/NACK) for every 8 bits sent

Open-Drain Bus Drivers

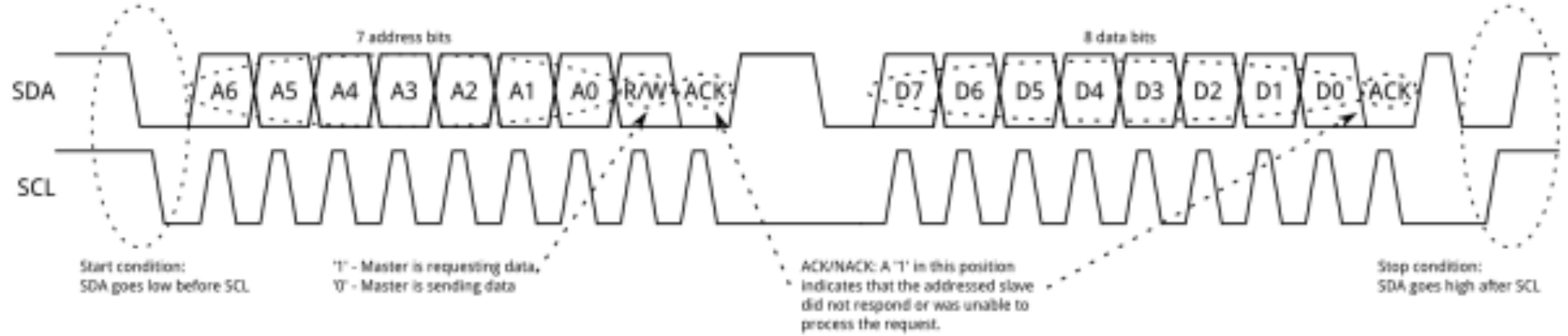


- **Pullup resistors ensure no damaged pins or excessive power consumption if bus contention occurs**

I²C Protocol

- **Messages segmented into two types of frame**
  - Address frame: master indicates which slave to communicate
  - Data frame(s): 8-bit packets transmitted from master to slave or vice versa
- **Sequence and polarity of transitions on SDA and SCL lines indicate start/stop and frame type**
  - Start Condition: SCL left high, SDA pulled low by master
  - Address Frame: always sent first; 7-bit address clocked out MSB first, followed by read/write (R/W) bit (0 = write, 1 = read)
  - Data Frame: master generates clock and either master or slave transmits data on SDA depending on R/W bit
  - Stop Condition: SDA goes high after SCL goes high

I²C Basic Example



- **NACK/ACK bit is 9th bit of all frames**
  - After first 8 bits sent, receiver pulls SDA low before 9th clock pulse (ACK)
  - If SDA left high, NACK is inferred and master decides next step
- **Advanced features include 10 bit addressing, repeated start conditions, and clock stretching**

References

- **Serial Communication**
  - https://learn.sparkfun.com/tutorials/serial-communication
- **SPI**
  - https://learn.sparkfun.com/tutorials/serial-peripheral-interface-spi
- **I^2C**
  - https://learn.sparkfun.com/tutorials/i2c