Bridge of Life Education

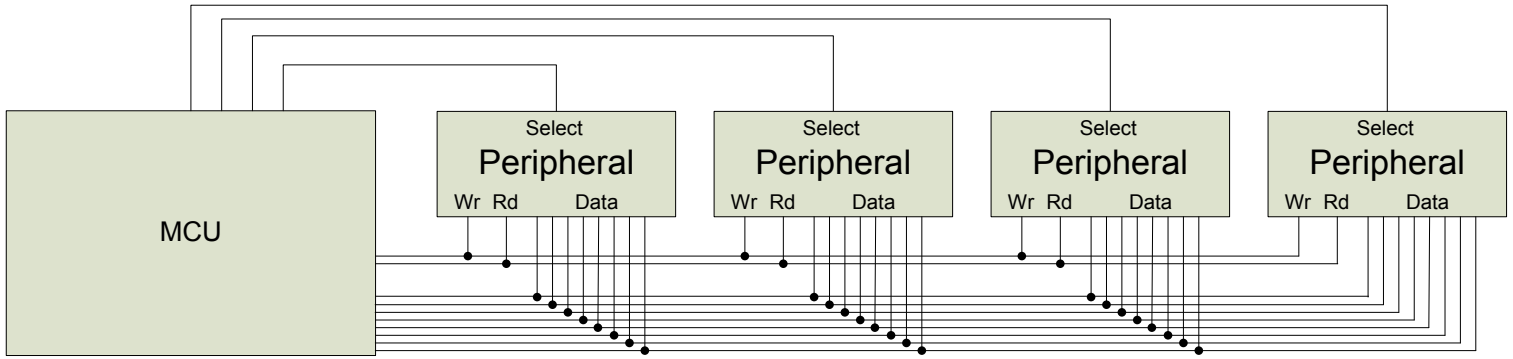
SOC Design
Peripheral – Serial Bus

Jiin Lai

Topics

- Serial Communication Basics
- UART, SPI, I2C Introduction - Protocol

Parallel Buses



- All devices use buses to share data, read, and write signals
- MCU uses individual select lines to address each peripheral
- MCU requires fewer pins for data, but still one per data bit
- MCU can communicate with only one peripheral at a time

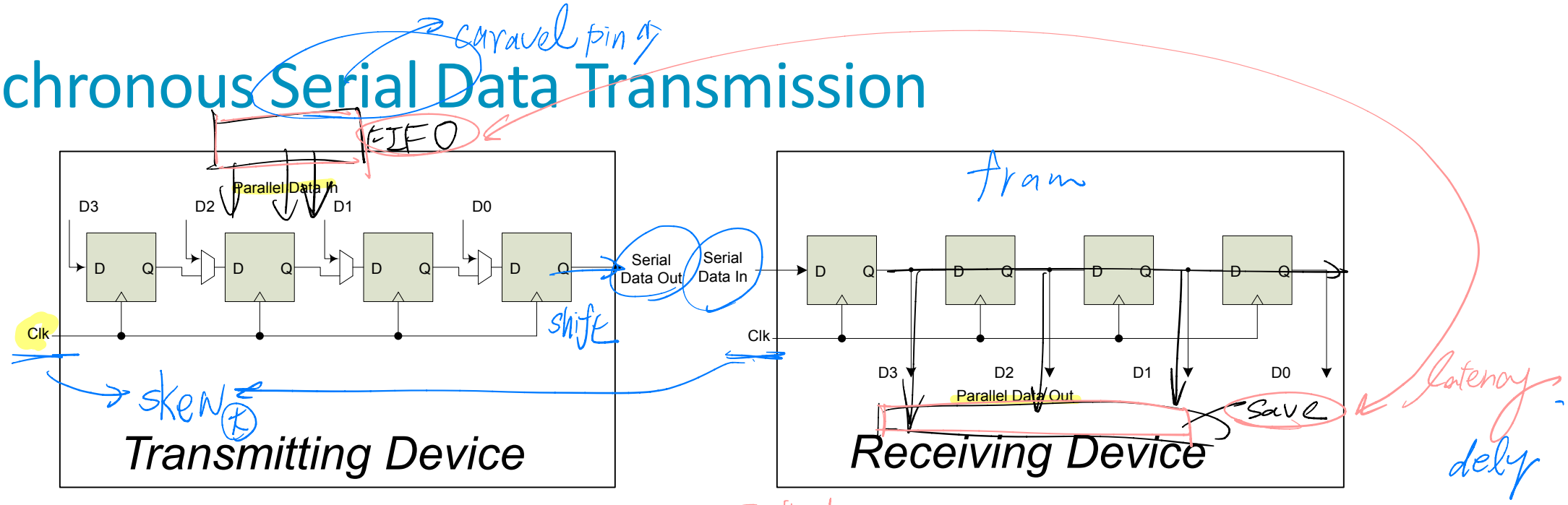
Types of Serial Communication

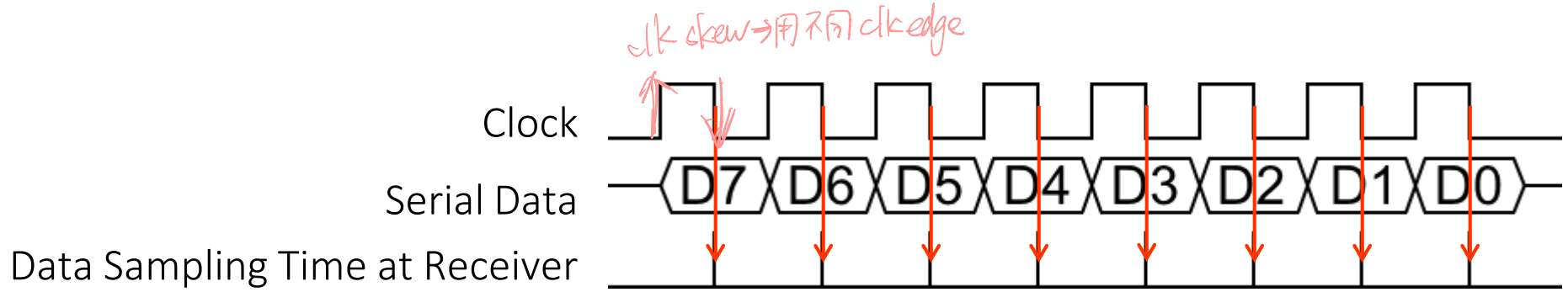
- **Synchronous** Serial Transmission
  - A common clock is shared by both the sender and the receiver
  - More efficient transmission, since one wire is dedicated to data transfer
  - Costly if an extra clock wire is required

- **Asynchronous** Serial Transmission
  - The sender does not have to send a clock signal
  - Both sender and receiver agree on **timing parameters** in advance
  - **Special bits** are added to synchronize transmission

- Full-Duplex, Half-Duplex

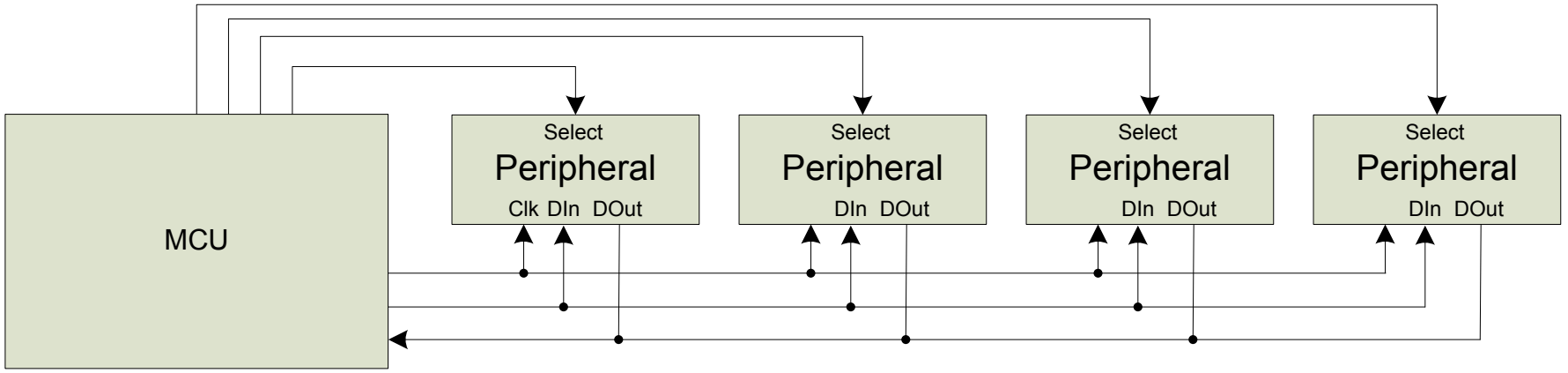
Synchronous Serial Data Transmission





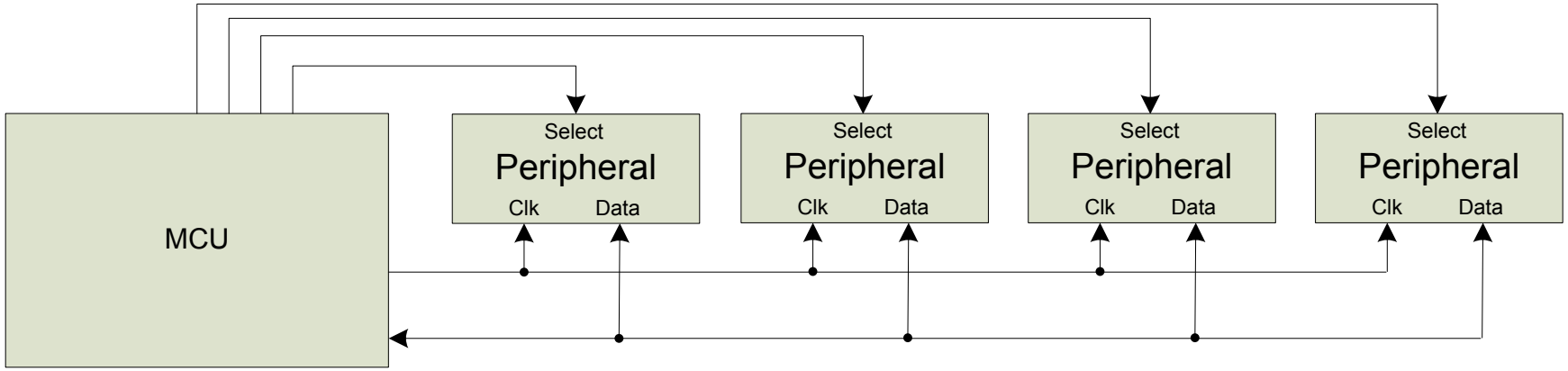
- Use shift registers and a clock signal to convert between serial and parallel formats
- Synchronous: an explicit clock signal is along with the data signal

Synchronous Full-Duplex Serial Data Bus



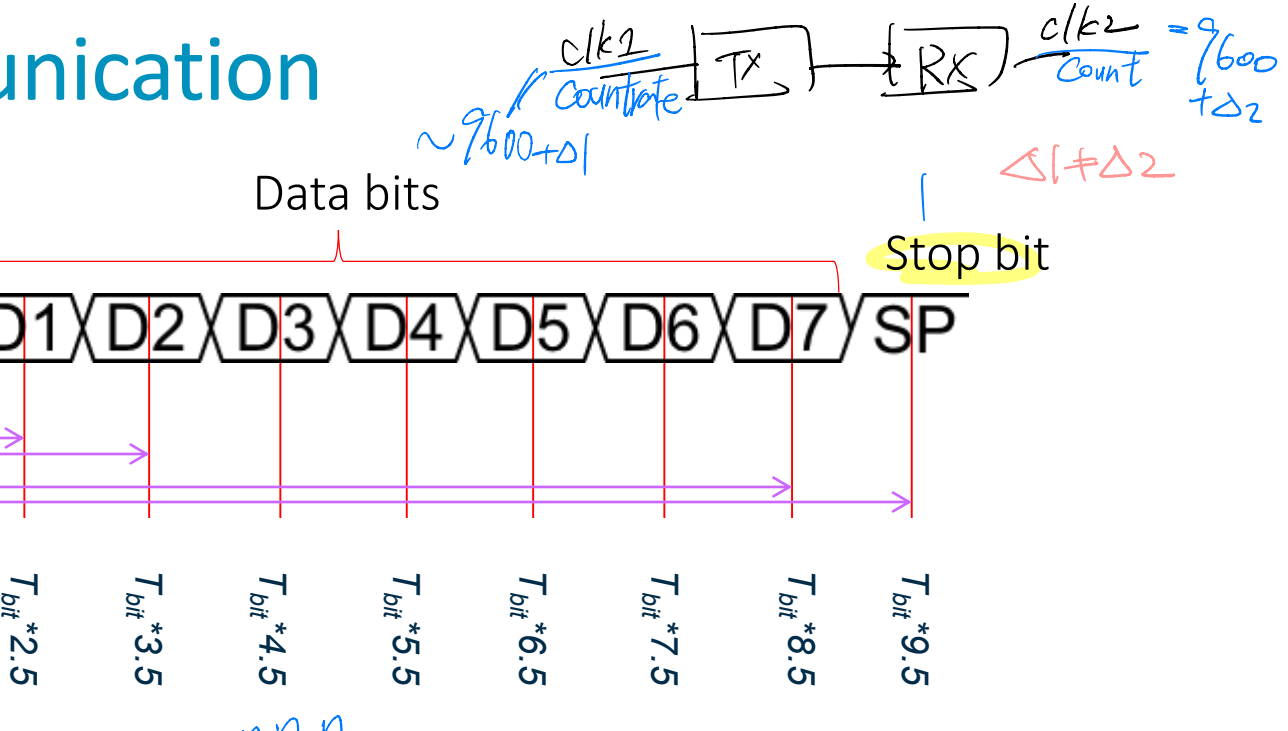
- Use two serial data lines - one for reading, one for writing.
  - Allows simultaneous send and receive full-duplex communication

Synchronous Half-Duplex Serial Data Bus



- Share the serial data line
- Doesn’t allow simultaneous send and receive - is half-duplex communication

Asynchronous Serial Communication

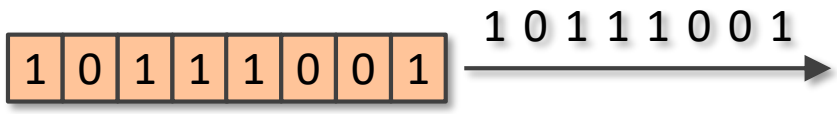


- Eliminate the clock line!
- Transmitter and receiver must generate clock locally
- Transmitter must add start bit (always same value) to indicate start of each data frame
- Receiver detects leading edge of start bit, then uses it as a timing reference for sampling data line to extract each data bit N at time Tbit*(N+1.5)
- Stop bit is also used to detect some timing errors

Serial v.s. Parallel Communication

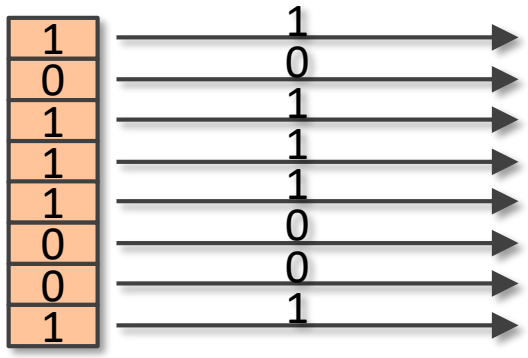
Serial

- Less wire cost
- More reliability
- Higher clock rate
- Limited throughput
- UART, SPI, I2C, USB, Ethernet, PCI Express, etc.

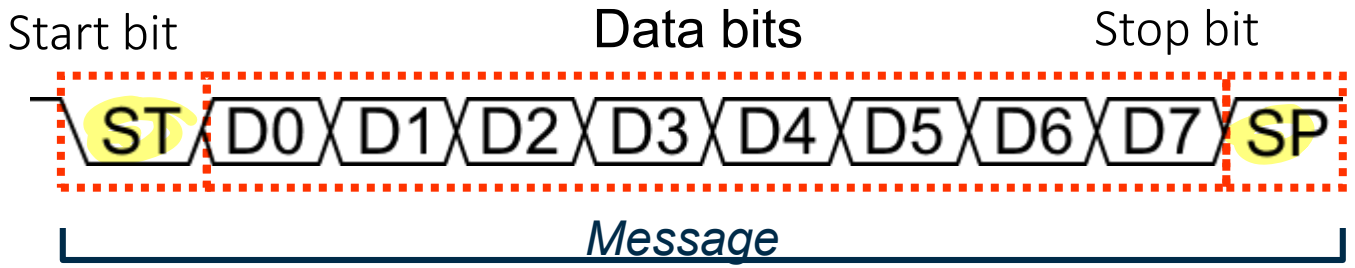


Parallel

- More wire cost
- Less reliable (more connector contacts to fail)
- Slower clock rate
- Higher throughput



Serial Communication Specifics



Network protocal

- Data frame fields
  - Start bit (one bit)
  - Data (LSB first or MSB, and size – 7, 8, 9 bits)
  - Optional parity bit is used to make total number of ones in data even or odd
  - Stop bit (one or two bits)
- All devices must use the same communications parameters
  - E.g. communication speed (300 baud, 600, 1200, 2400, 9600, 14400, 19200, etc.)
- Sophisticated network protocols have more information in each data frame
  - Medium access control – when multiple nodes are on bus, they must arbitrate for permission to transmit
  - Addressing information – for which node is this message intended?
  - Larger data payload
  - Stronger error detection or error correction information
  - Request for immediate response (“in-frame”)

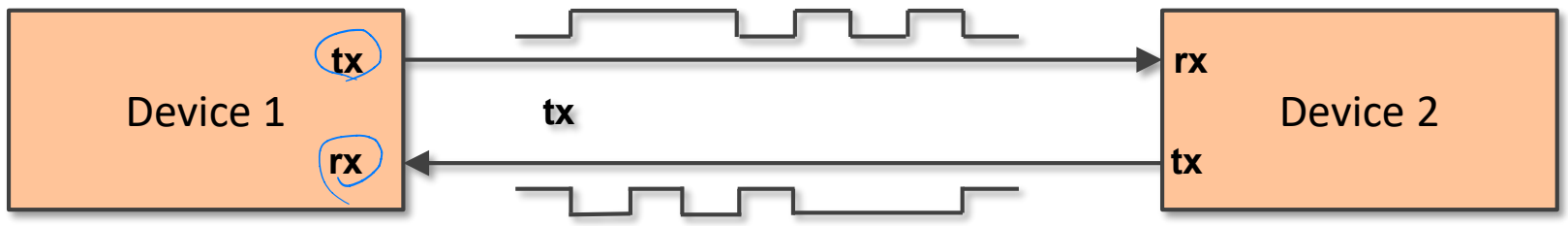
Error Detection parity

- Can send additional information to verify data was received correctly
- Need to specify which parity to expect: even, odd or none.
- Parity bit is set so that total number of “1” bits in data and parity is even (for even parity) or odd (for odd parity)
  - 01110111 has 6 “1” bits, so parity bit will be 1 for odd parity, 0 for even parity
  - 01100111 has 5 “1” bits, so parity bit will be 0 for odd parity, 1 for even parity
- Single parity bit detects if 1, 3, 5, 7 or 9 bits are corrupted, but doesn’t detect an even number of corrupted bits
- Stronger error detection codes (e.g. Cyclic Redundancy Check) exist and use multiple bits (e.g. 8, 16), and can detect many more corruptions. CRC
  - Used for CAN, USB, Ethernet, Bluetooth, etc.

UART

UART Overview

- UART
  - Asynchronous communication, no clock wire required, pre-agreed baud rate
  - Separate transmission and receiving wires
- UART communication
  - the transmitter and receiver must agree on timing parameters beforehand (baud rate, character length, parity, stop bits)
  - Converts data from parallel to serial
  - Sequential data is transferred through serial cable
  - Receives the sequential data and reassembles it back to parallel



UART Protocol

- Data transfer starts with a starting bit by driving logic to low for one clock cycle.
- In the next eight clock cycles, eight bits are sent sequentially from the transmitter.
- Optionally, one parity bit can be added to improve transfer reliability.
- In the end, the data wire is pulled up high to indicate completion of the transfer.

Transfer one byte without parity bit

Transfer one byte with parity bit

Terms

- **Baud Rate**: number of bit transfers per second (clock cycle)
  - 9600, 19200, 38400, 57600, 115200, 230400, 460800, 921600, 1000000, 1500000
- Data bits: the actual data bits in a transmission.
  - data packets are 5-9 bits. ASCII (0 to 127 – 7 bits), Extended ASCII (8bits),
  - A packet: start (1), data bits (5-9), parity (0-1), stop (1-2)
- Stop bits: signal the end of communication for a single packet (1, 1.5, and 2 bits)
  - two devices to become slightly out of sync., give the computers some room for error in the clock speeds
- Parity: error checking, four types of parity: even, odd, marked, and spaced
  - Marked and spaced parity does not actually check the data bits, but simply sets the parity bit high for marked parity or low for spaced parity. enables the device to determine if noise is corrupting the data or if the transmitting and receiving devices' clocks are out of sync.

https://www.analog.com/en/analog-dialogue/articles/uart-a-hardware-communication-protocol.html

Advantages / Disadvantages

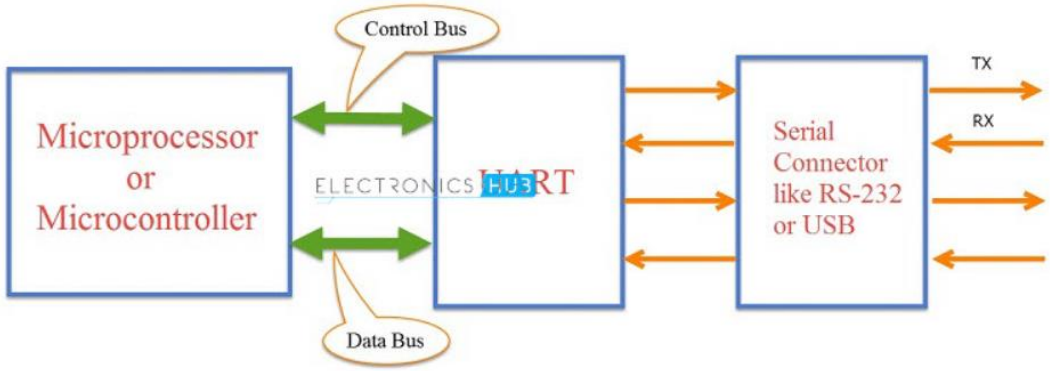
ADVANTAGES

- Only uses two wires
- No clock signal is necessary
- Has a parity bit to allow for error checking
- The structure of the data packet can be changed as long as both sides are set up for it
- Well documented and widely used method

DISADVANTAGES

- The size of the data frame is limited to a maximum of 9 bits
- Doesn’t support multiple slave or multiple master systems
- The baud rates of each UART must be within 10% of each other

UART Use Cases



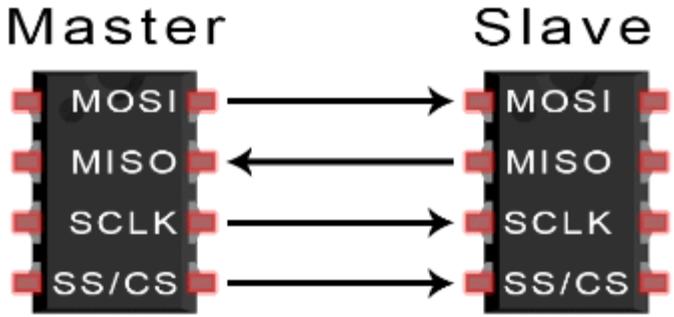
- Debugging: Capturing messages from the system.
- Manufacturing function-level tracing: Logs are very important in manufacturing.
- Testing/verification: Verifying products before they leave the manufacturing process helps deliver the best quality products possible to customers.

SPI - Serial Peripheral Interface

https://www.circuitbasics.com/basics-of-the-spi-communication-protocol

SPI Overview

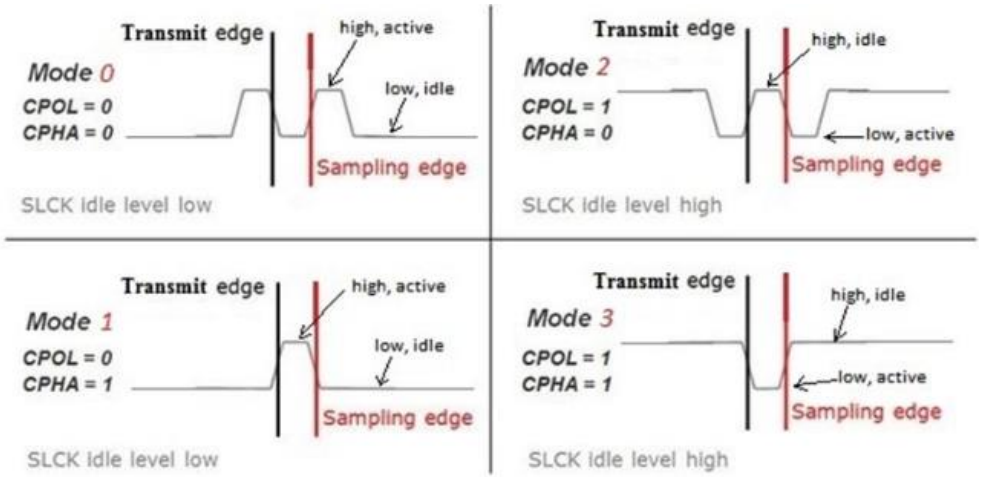
- Synchronous communication, clock wire required
- Separate transmit and receive wires – 4 wire serial bus
  - MOSI (Master Output/Slave Input)
  - MISO (Master Input/Slave Output)
  - SCLK (Clock)
  - SS/CS (Slave Select/Chip Select) – Active Low
- Data can be transferred in a continuous stream without interruption.
- Devices communicate in Master/Slave Mode.



Clock Polarity (CPOL), Clock Phase (CPHA)

- CPOL represents the polarity of the clock
- CPHA represents the phase of each data bit's transmission cycle relative to SCLK.

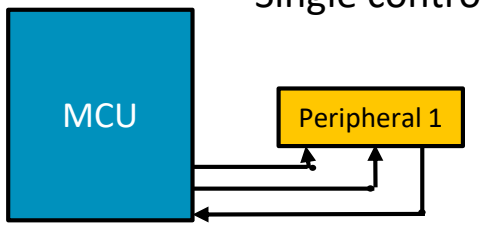
SPI mode	Clock polarity (CPOL)	Clock phase (CPHA)	Data is shifted out on	Data is sampled on
0	0	0	falling SCLK, and when $\overline{CS}$ activates	rising SCLK
1	0	1	rising SCLK	falling SCLK
2	1	0	rising SCLK, and when $\overline{CS}$ activates	falling SCLK
3	1	1	falling SCLK	rising SCLK



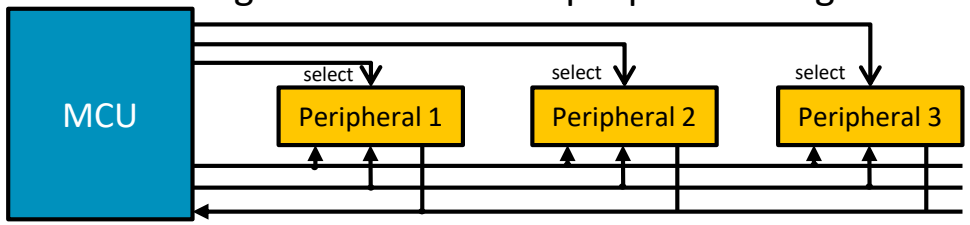
Configuration

- All chips share bus signals
  - Clock SCK
  - Data lines MOSI (Master out, Slave in) and MISO (Master in, Slave out)
- Each peripheral has its own chip select line (CS)
  - Controller (MCU) only asserts the CS line of the peripheral with which it wants to communicate

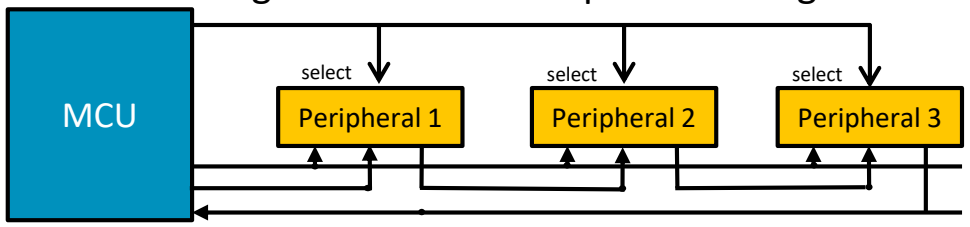
Single controller single target



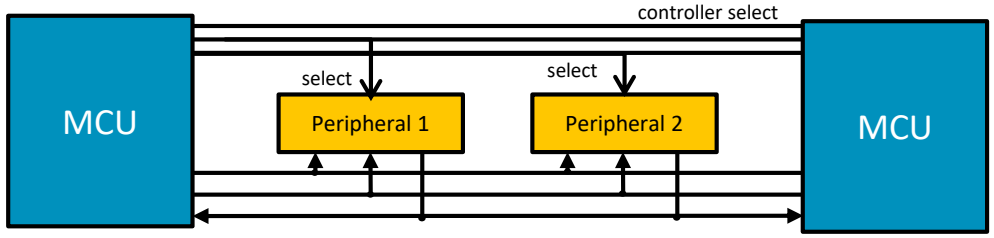
Single controller multiple parallel targets



Single controller multiple serial targets



Multiple controllers multiple parallel targets



Advantage / Disadvantage

ADVANTAGES

- No start and stop bits, so the data can be streamed continuously without interruption
- No complicated slave addressing system like I2C
- Higher data transfer rate than I2C (almost twice as fast)
- Separate MISO and MOSI lines, so data can be sent and received at the same time

DISADVANTAGES

- Uses four wires (I2C and UARTs use two)
- No acknowledgement that the data has been successfully received (I2C has this)
- No form of error checking like the parity bit in UART
- Only allows for a single master

SPI Flash – Infineon S25FL064L

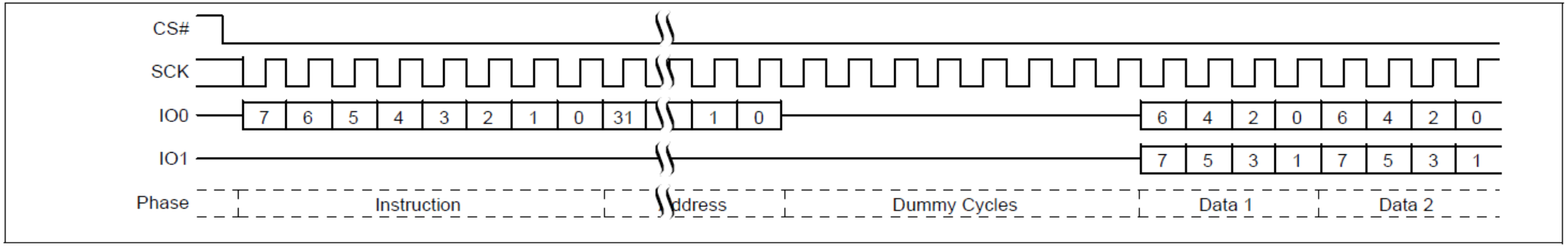


Figure 17 Dual Output Read command

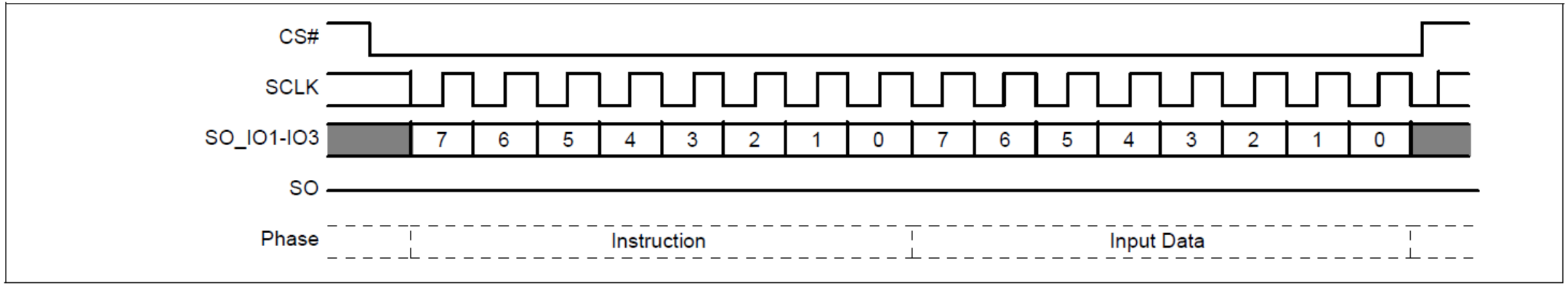


Figure 14 Single Bit Wide Input command

https://www.infineon.com/dgdl/Infineon-S25FL064L_64-Mbit_(8-Mbyte)_3.0_V_FL-L_SPI_Flash_Memory-DataSheet-v07_00-EN.pdf?fileId=8ac78c8c7d0d8da4017d0ee2d2846996&utm_source=cypress&utm_medium=referral&utm_campaign=202110_globe_en_all_integration-files&redirId=File_3_302

SPI Usages

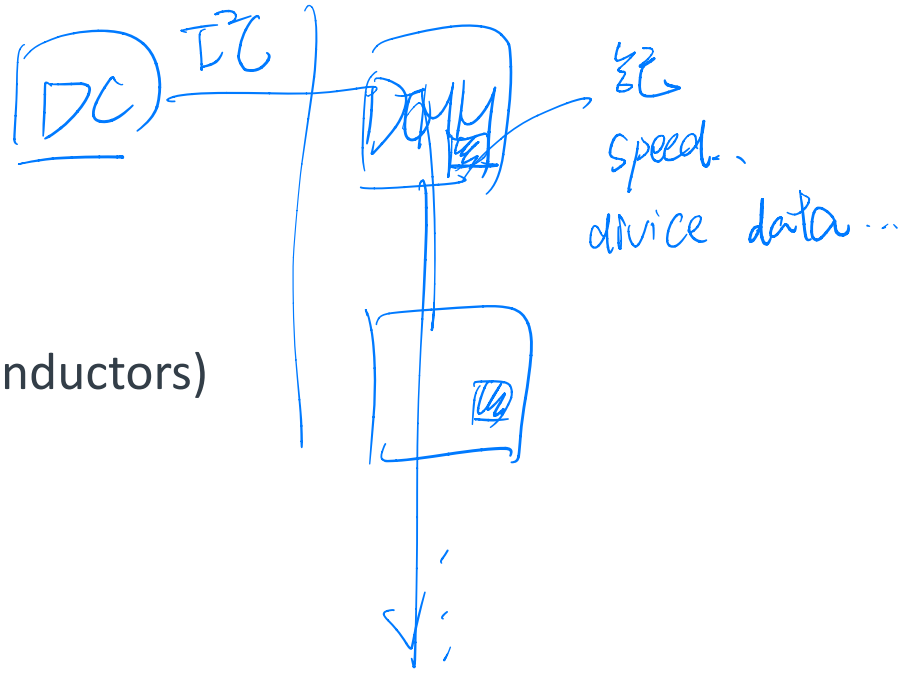
- Sensors: temperature, pressure, ADC, touchscreens, video game controllers
- Control devices: audio codecs, digital potentiometers, DACs
- Camera lenses: Canon EF lens mount
- Communications: Ethernet, USB, USART, CAN, IEEE 802.15.4, IEEE 802.11
- Memory: flash and EEPROMs
- Real-time clocks
- LCDs, sometimes even for managing image data
- Any MMC or SD card (including SDIO variant)
- Shift registers for additional I/O

I2C

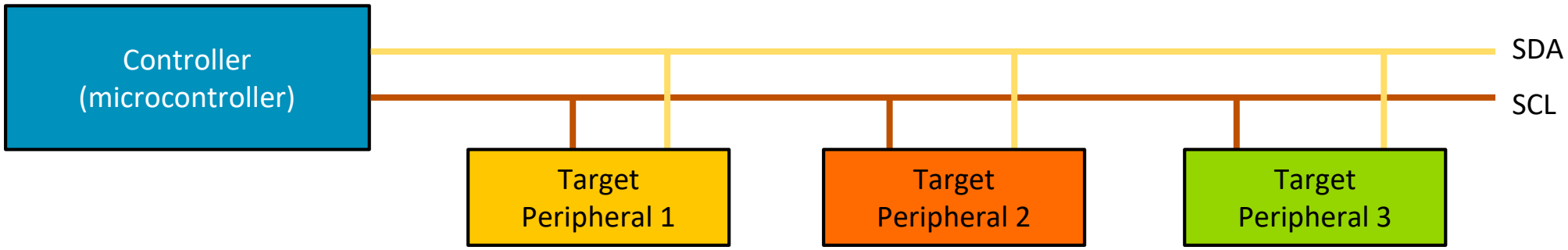
https://www.circuitbasics.com/basics-of-the-i2c-communication-protocol

NXP I2C Manual: https://www.nxp.com/docs/en/application-note/AN10216.pdf

I²C Bus Overview

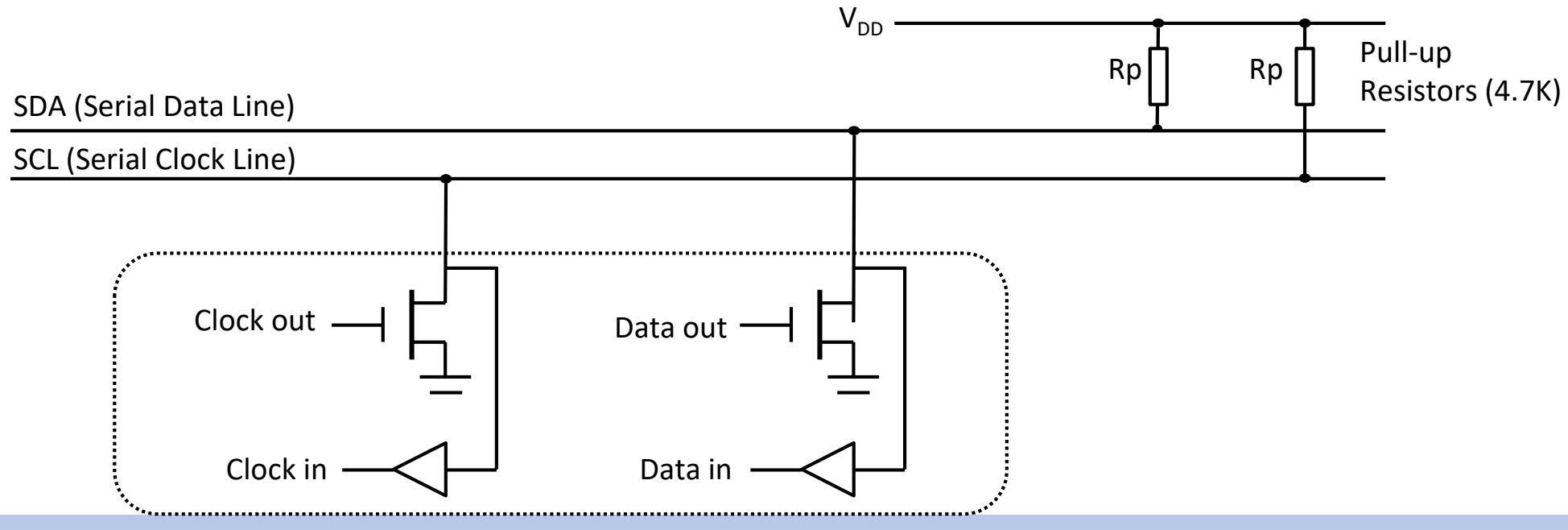


- Inter-integrated Circuit (I2C) bus:
  - Multi-controller serial single-ended computer bus
  - Invented by Philips semiconductor division (today: NXP Semiconductors)
  - Communicates with low-speed peripherals
  - Two signal lines
    - SCL: Serial clock
    - SDA: Serial data



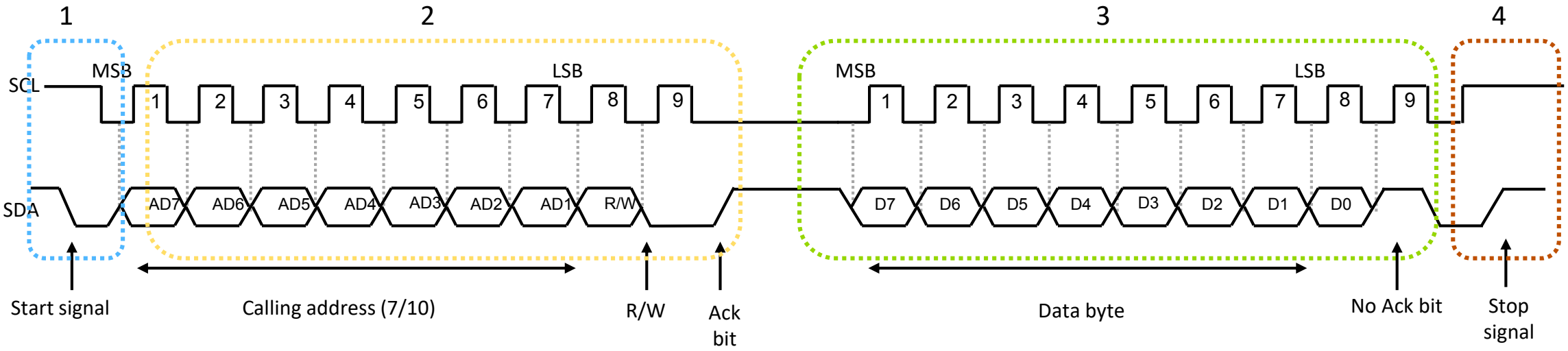
I²C Bus Connections

- Bus is typically controlled by controller device; targets respond when addressed.
- Resistors pull up lines to VDD
- Open-drain transistors pull lines down to ground
- Controller generates SCL clock signal
  - Can range up to 400 kHz, 1 MHz, or more



I²C Message Format

- **Message-oriented data transfer with four parts:**
  1. Start condition
     - SDA line switches from high to a low *before* the SCL line switches from high to low.
  2. Target address transmission
     - Address (7/10)
     - Command (read H or write L)
     - Acknowledgement by receiver
  3. Data fields
     - Data byte
     - Acknowledgement by receiver
  4. Stop condition
     - SDA line switches from low to high voltage after the SCL switches from low to high.



I2C Devices

I²C Device Categories

- TV Reception
- Radio Reception
- Audio Processing
- Infrared Control
- DTMF
- LCD display control
- Clocks/timers
- General Purpose I/O
- LED display control
- Bus Extension/Control
- A/D and D/A Converters
- EEPROM/RAM
- Hardware Monitors
- Microcontroller

Advantage / Disadvantage

ADVANTAGES

- Only uses two wires
- Supports multiple masters and multiple slaves
- ACK/NACK bit gives confirmation that each frame is transferred successfully
- Hardware is less complicated than with UARTs
- Well known and widely used protocol

DISADVANTAGES

- Slower data transfer rate than SPI
- The size of the data frame is limited to 8 bits
- More complicated hardware needed to implement than SPI

Reference

- Pyserial
  - https://github.com/pyserial/pyserial
- Reference
- https://learn.sparkfun.com/tutorials/serial communication/all
- https://www.omnicalculator.com/other/baud rate
- https://caravel harness.readthedocs.io/en/latest/uart.html
- bol edu/caravel soc (github.com)