Communication Protocols I

Team Emertxe

Communication Protocols I

- Introduction
- UART
- SPI
- I^2C
- CAN

Introduction

Introduction

- What do mean by Communication?
- Mode of Communications
- Type of Communications
- Why Protocols?

UART

UART

- Introduction
- Interface
- Hardware Configurations
- Frame Format

UART
Introduction

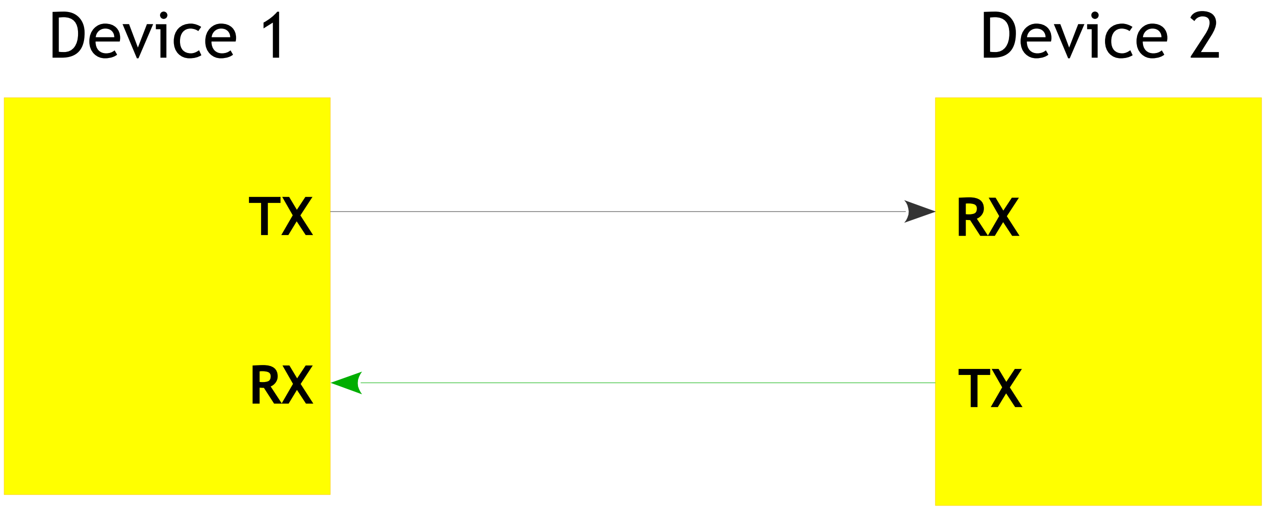
- Asynchronous
- Duplex - Any
- Master / Slave

UART
Interface

- RX
- TX

UART

Hardware Configuration



UART

Frame Format

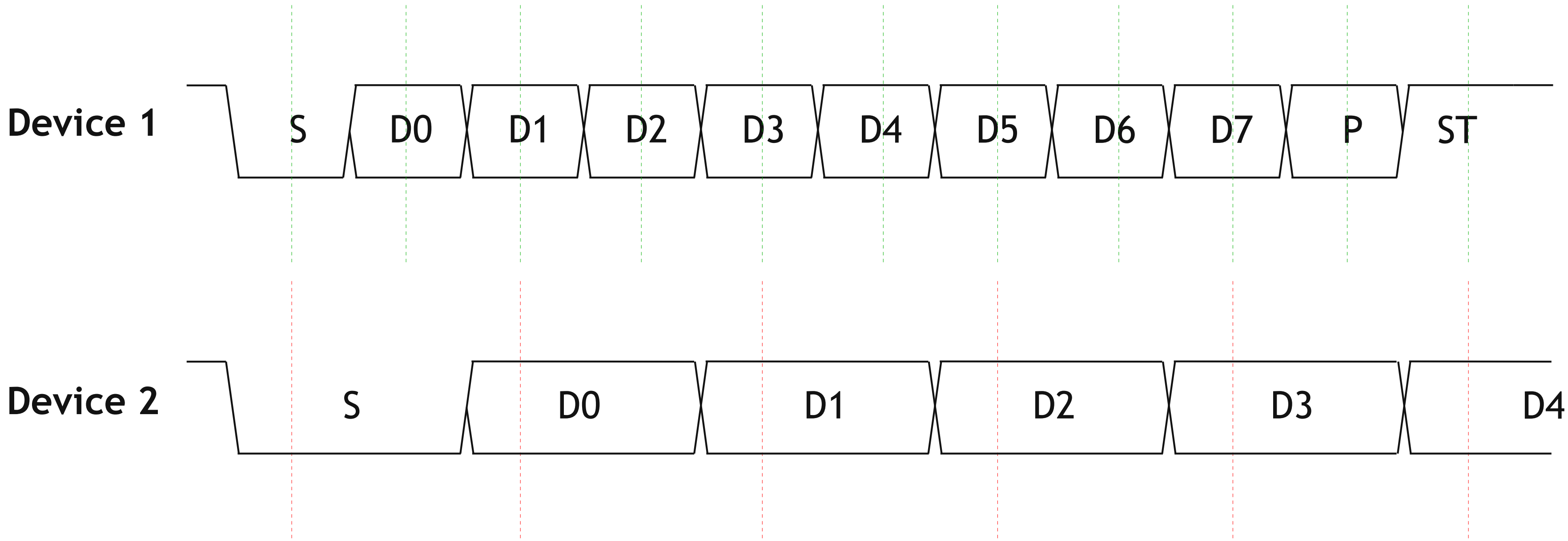
- Data part can be 5 to 9 bits
- Stop could be 2 bits
- Parity could be 0 or 1 bit

UART

Baud Rate

- Number of symbols per second (In this context the a symbol is a bit)
- So, sometimes referred as Bit Rate (No of bits per second)
- The frequency of the data transfer
- Both transmitter and receiver has to agree upon a common frequency for data integrity

UART
Baud Rate

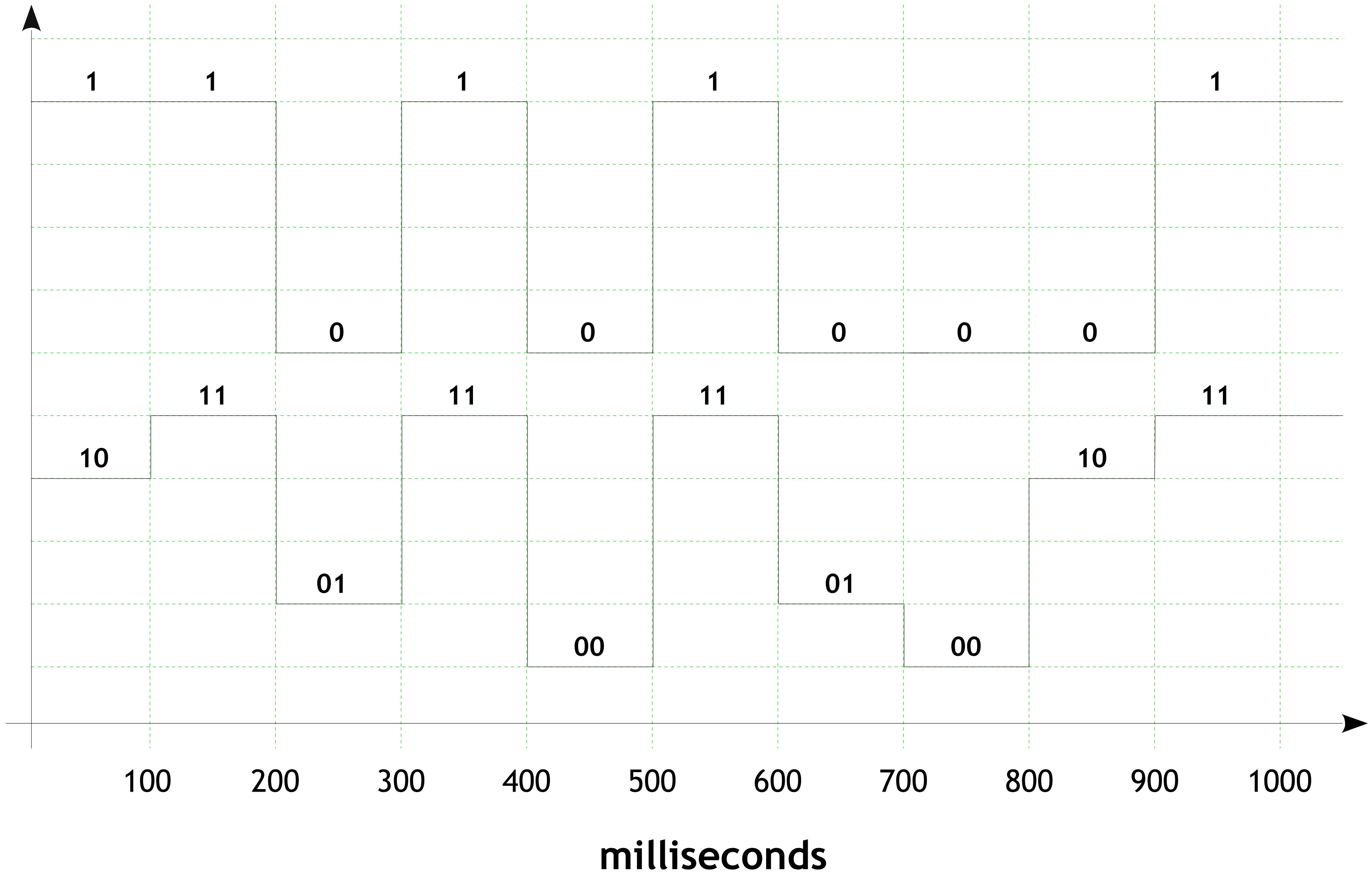


Transmitter Sample Frequency

Receiver Sample Frequency

UART

Baud Rate vs Bit Rate



Serial Peripheral Interface

Serial Peripheral Interface

- Introduction
- Interface
- Hardware Configurations
- Data Transmission
  - Data Validity

EMERTXE

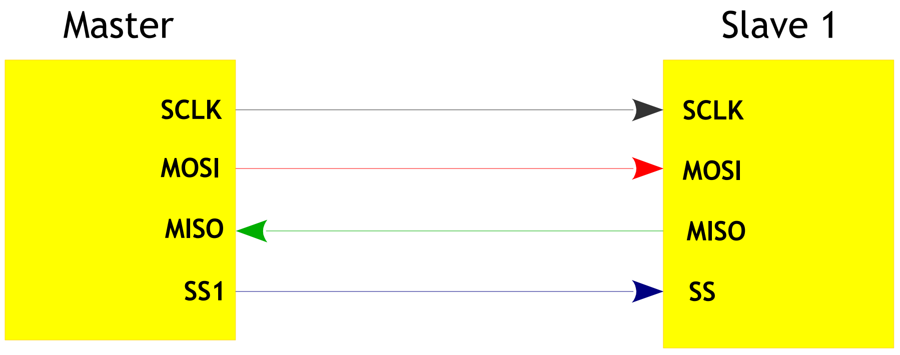
SPI
Introduction

- Synchronous
- Full Duplex
- Master / Slave

SPI
Interface

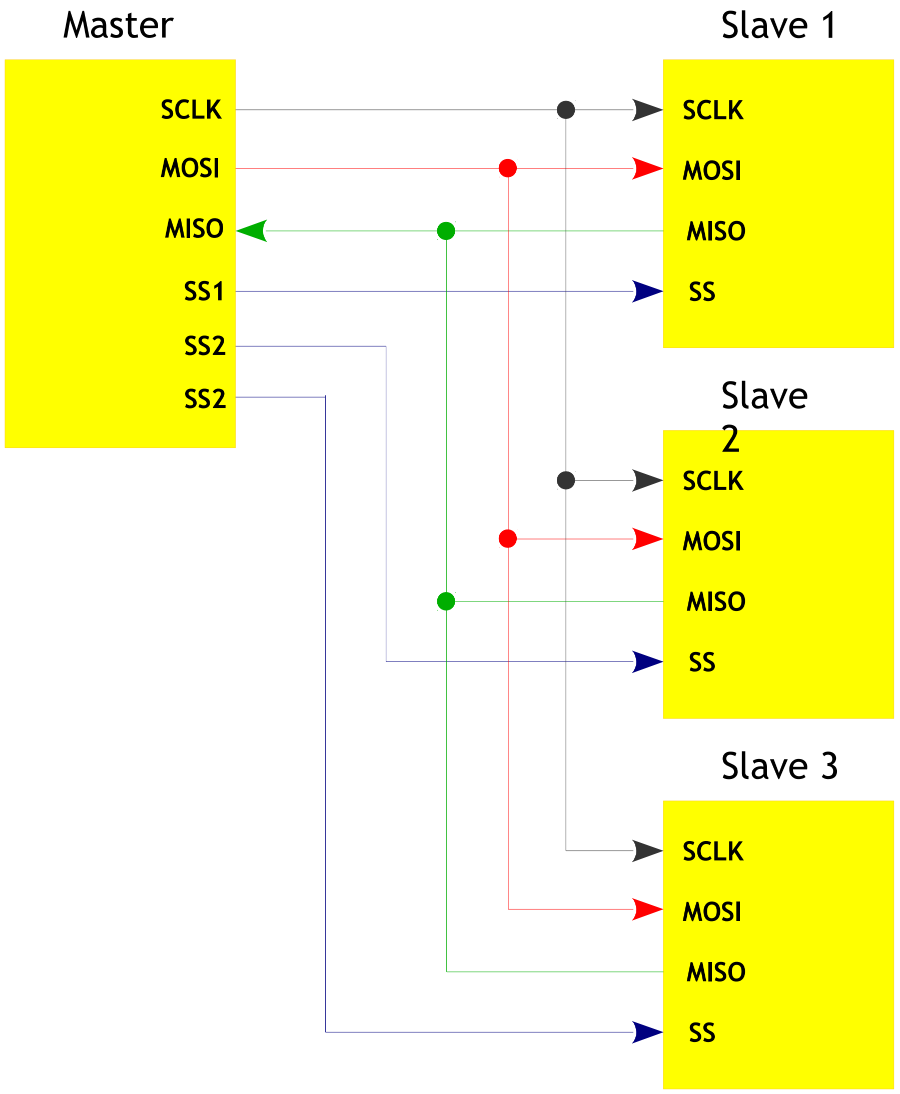
- SCLK
- MOSI
- MISO
- nSS

SPI
Hardware Configuration



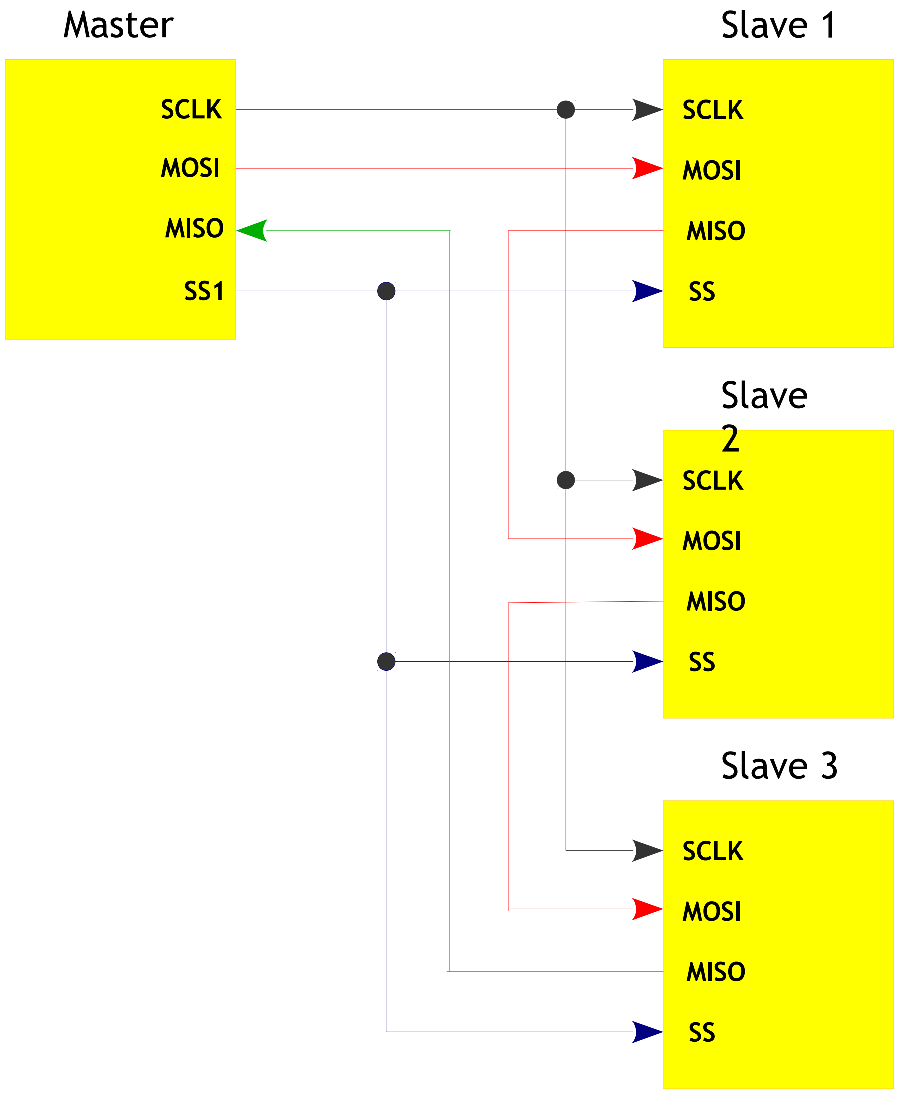
Single Master and Single Slave

SPI
Hardware Configuration



Single Master and Three Slaves

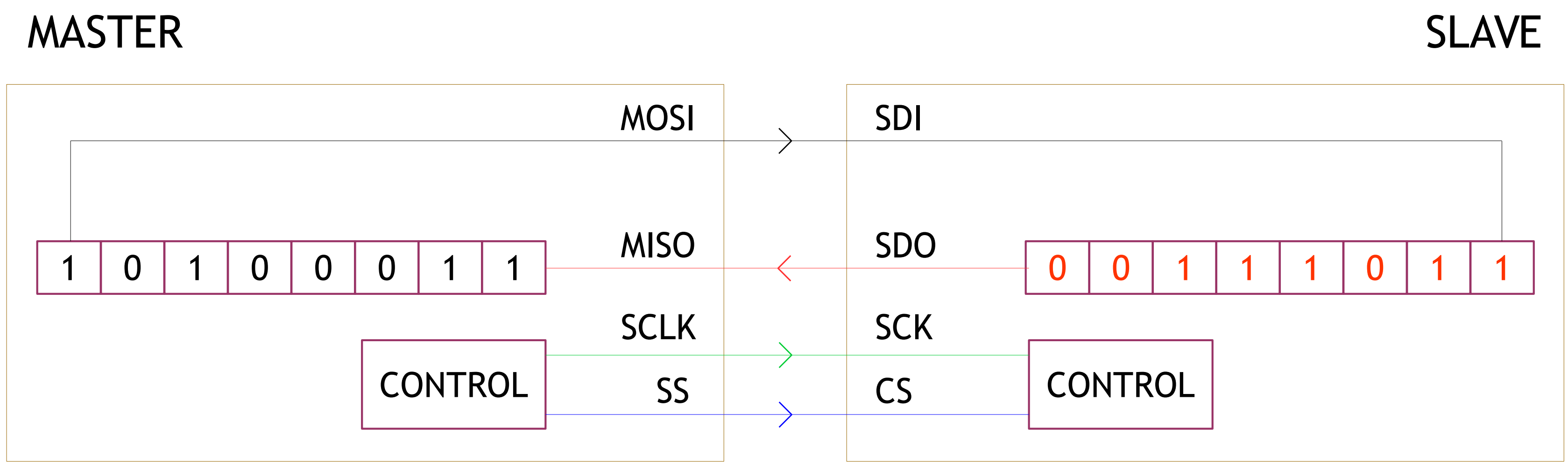
SPI
Hardware Configuration



Single Master and Three Daisy Chained Slaves

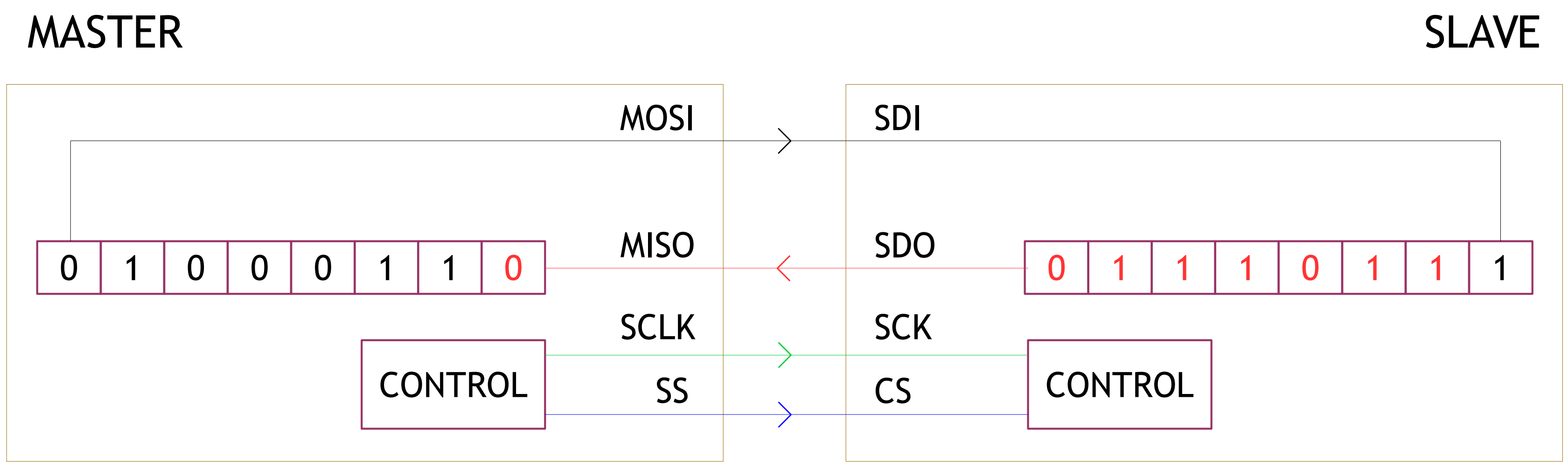
SPI

Data Transmission



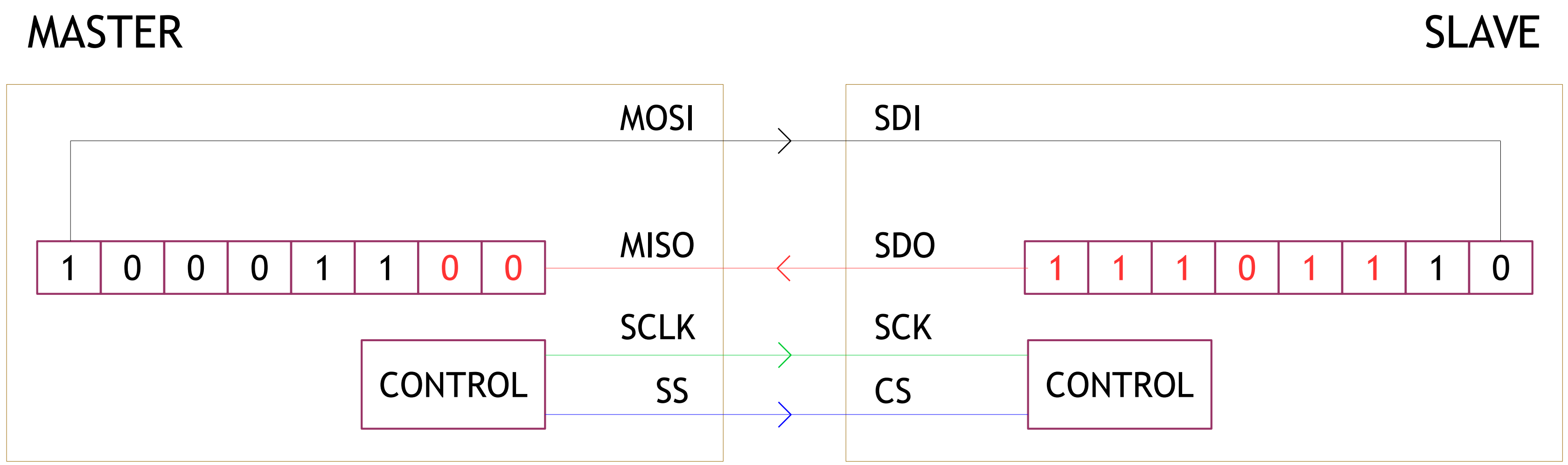
SPI

Data Transmission

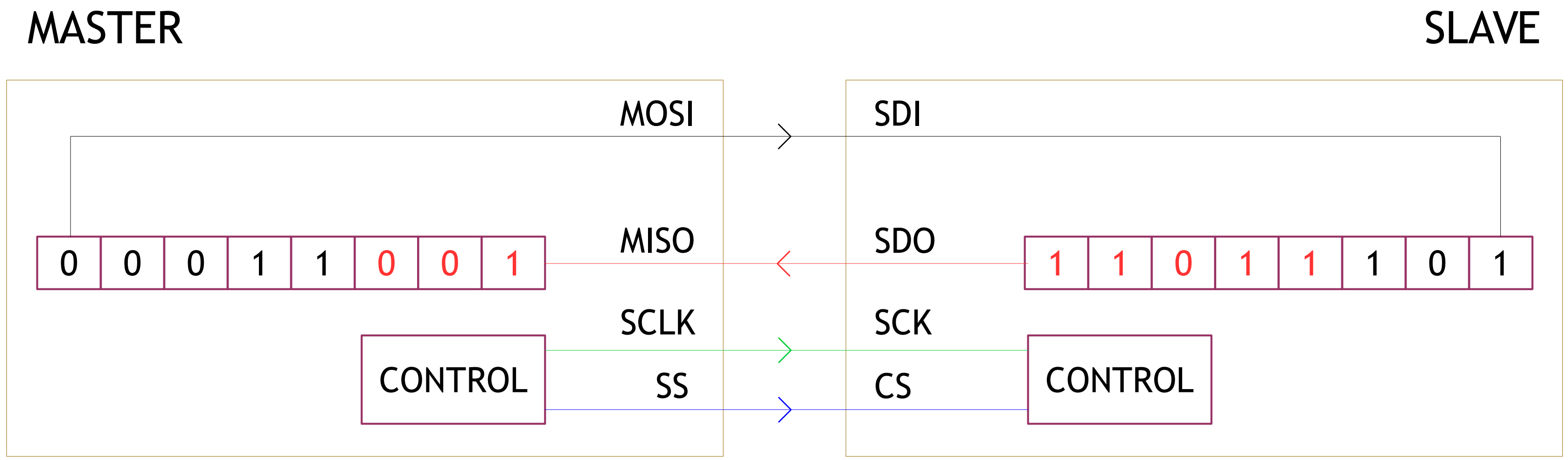


SPI

Data Transmission

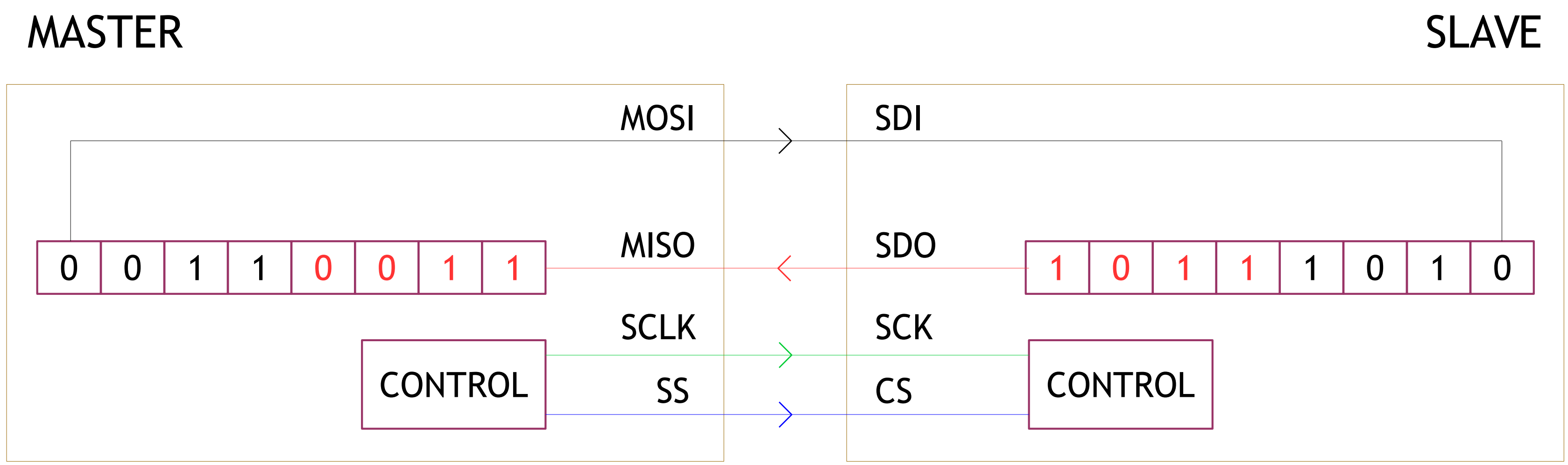


SPI
Data Transmission

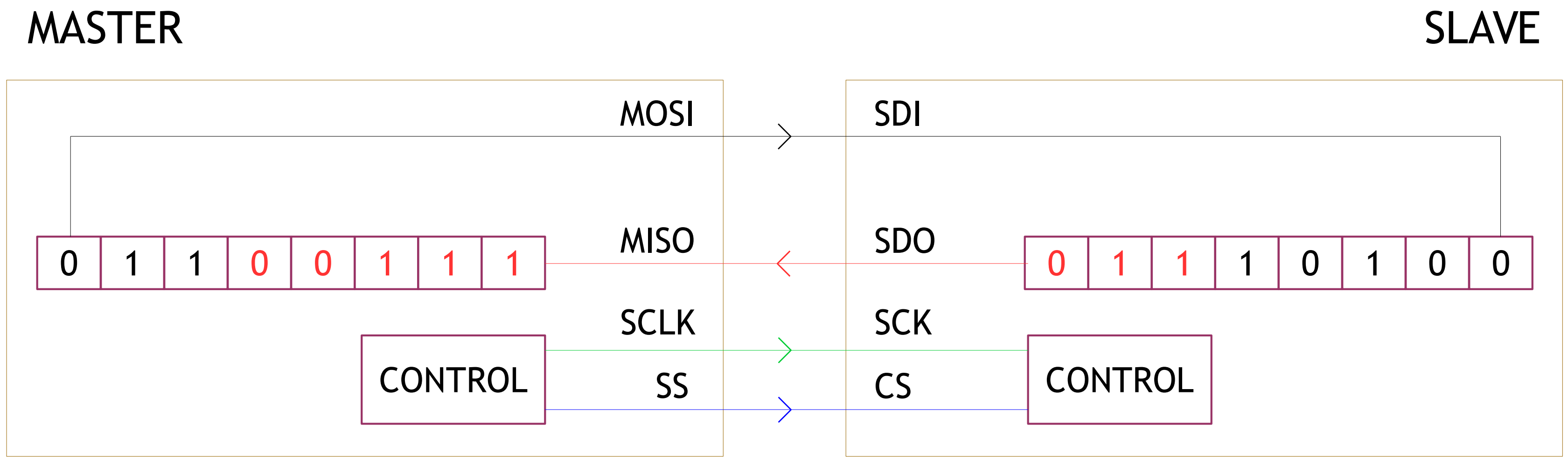


SPI

Data Transmission

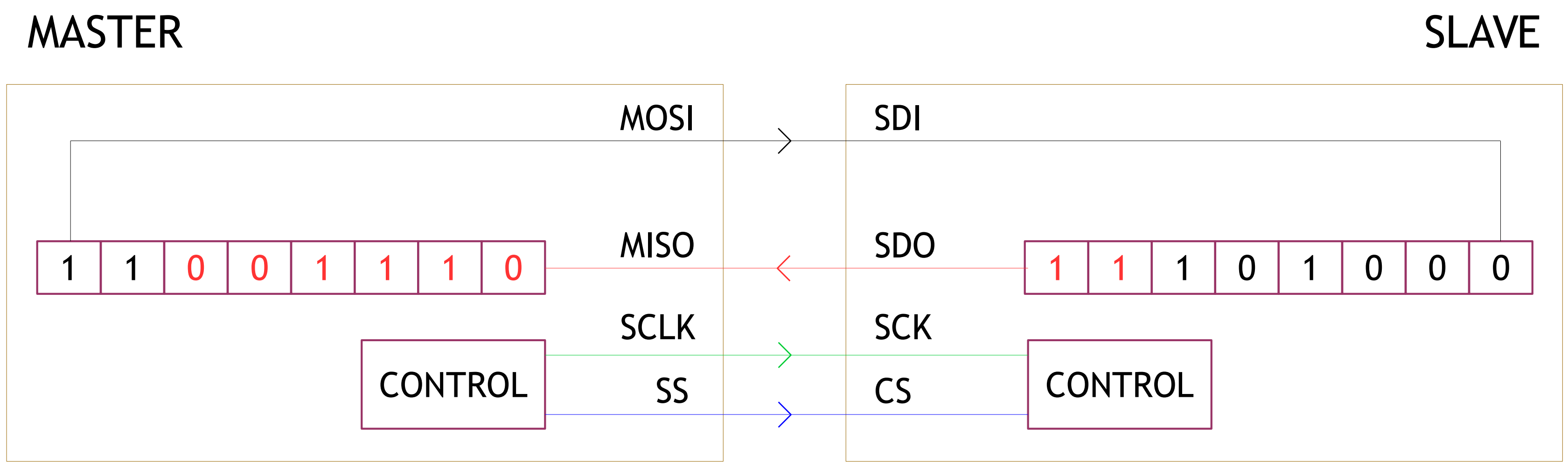


SPI
Data Transmission



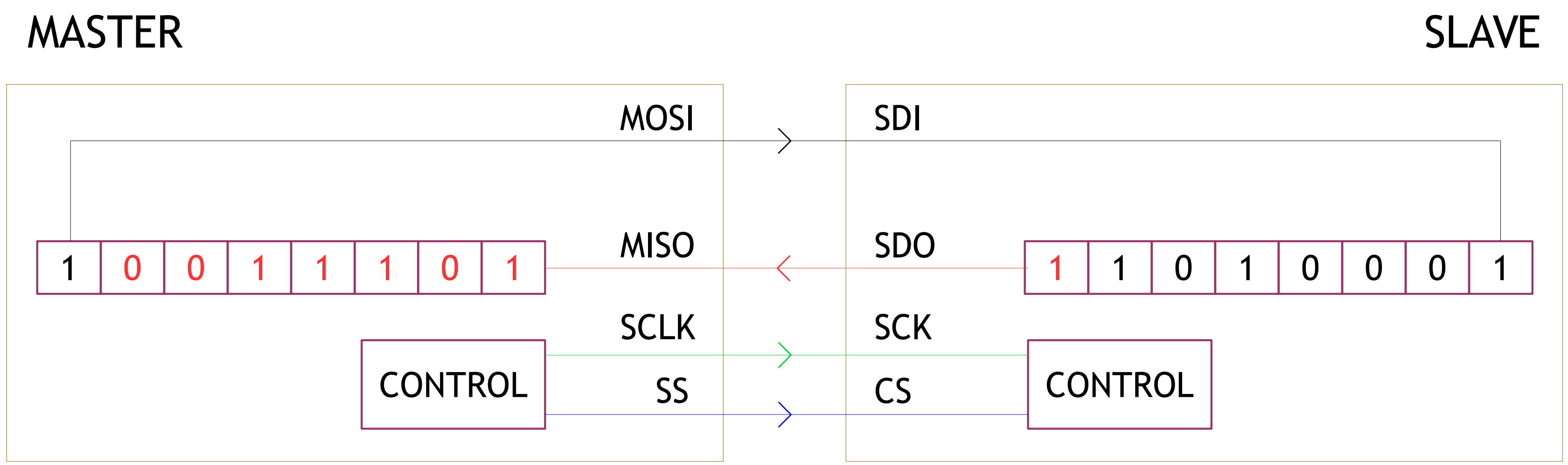
SPI

Data Transmission



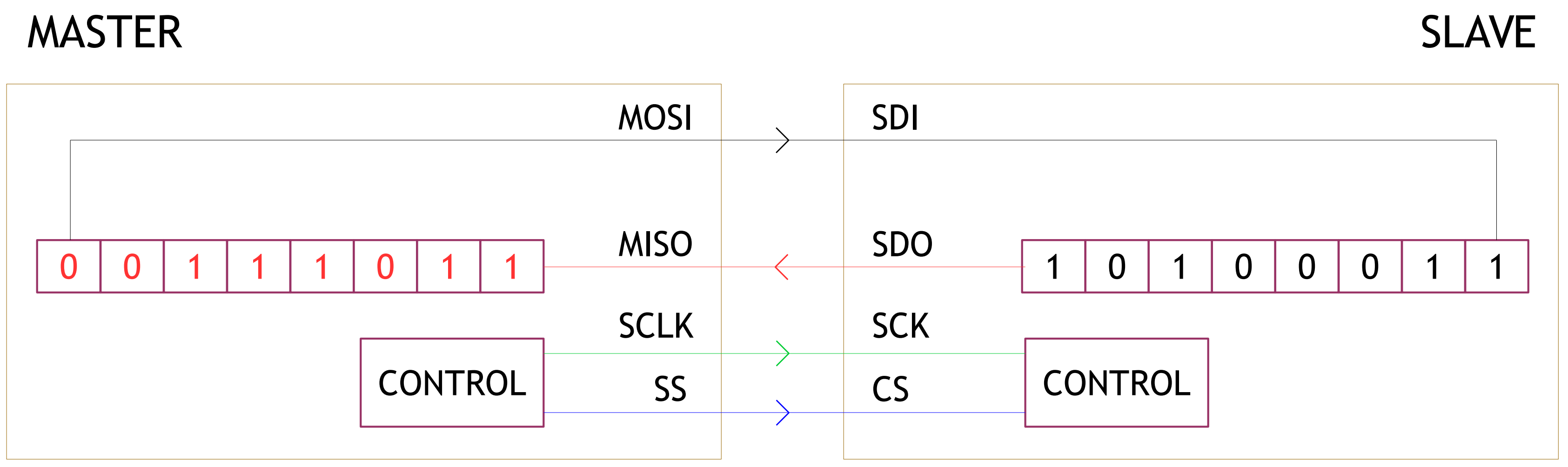
SPI

Data Transmission

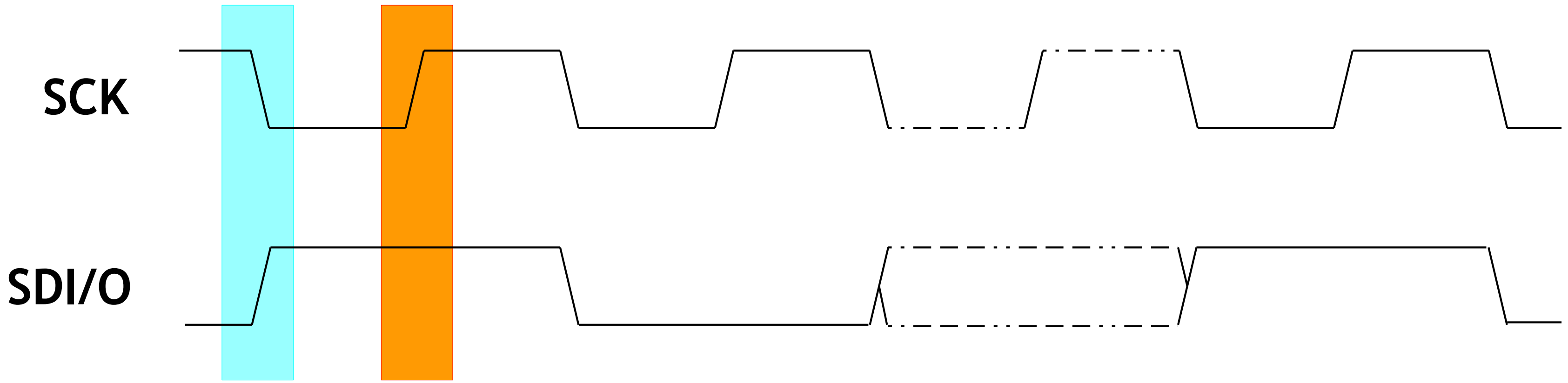


SPI

Data Transmission



SPI
Data Validity



Data Write

Data Read

Inter Integrated Circuits

Inter Integrated Circuits

- Introduction
- Bus Features
- The Protocol
- Bus Speeds

I^2C
Introduction

- Synchronous
- Half Duplex
- Multi Master / Slave

I^2C

Bus Features

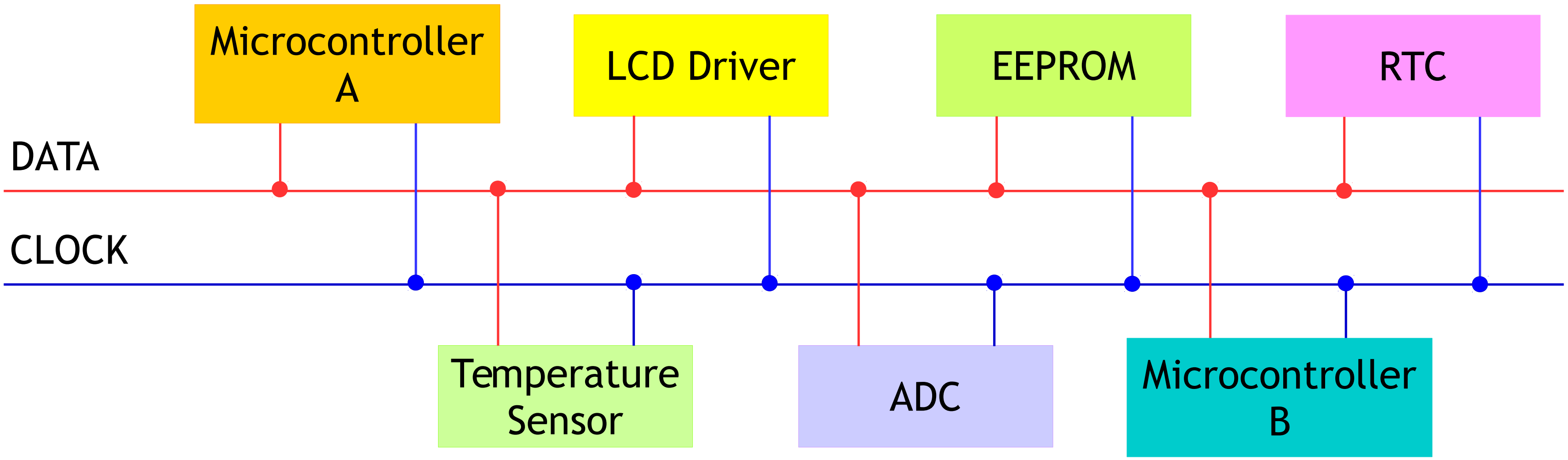
- Two Line Interface
- Software Addressable
- Multi Master with CD
- Serial, 8 bit Oriented, Bidirectional with 4 Modes
- On Chip Filtering

I^2C
Protocol

- Example
- Signals
- A Complete Data Transfer

I^2C

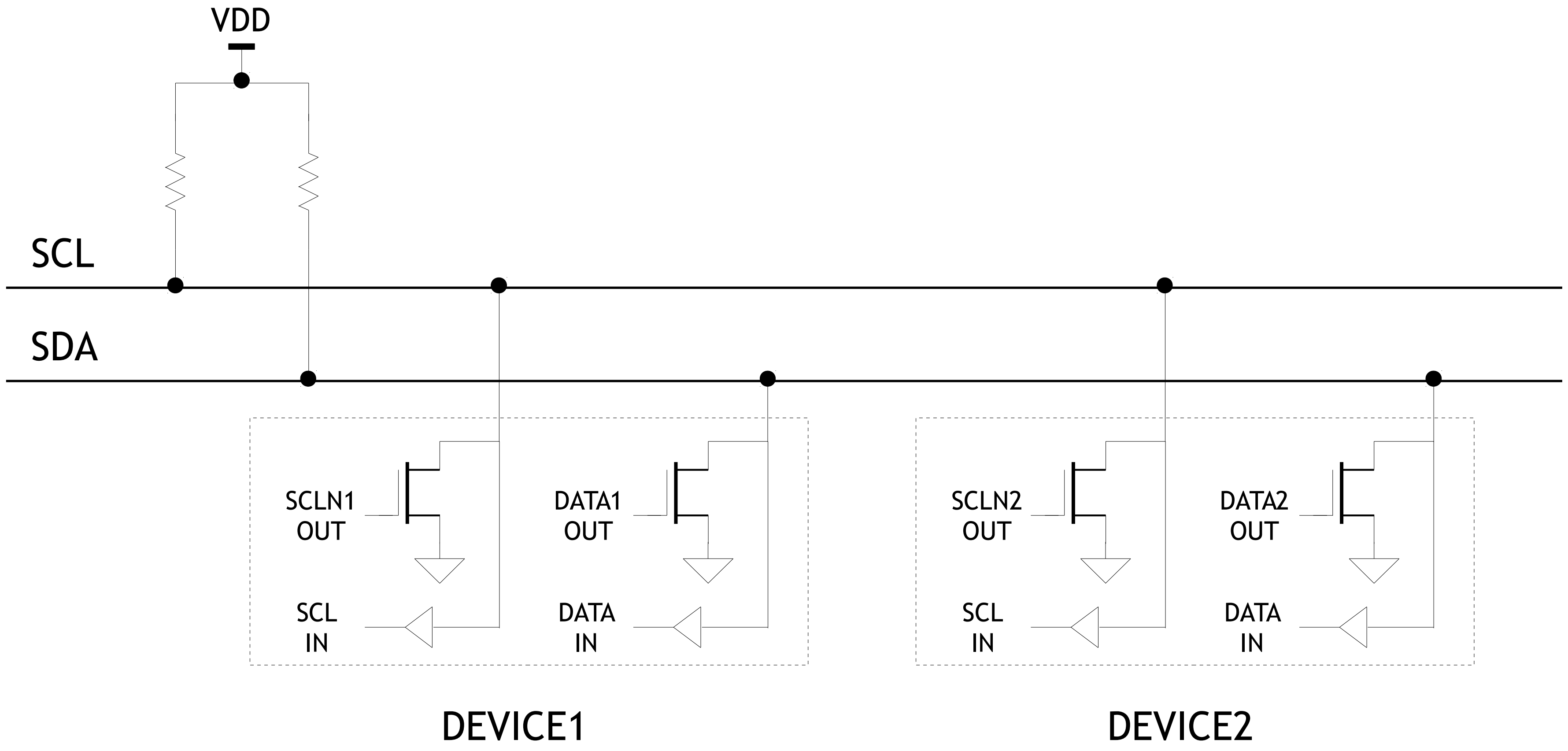
Example


Microcontroller A
LCD Driver
EEPROM
RTC
DATA
CLOCK
Temperature Sensor
ADC
Microcontroller B

I^2C
Signals

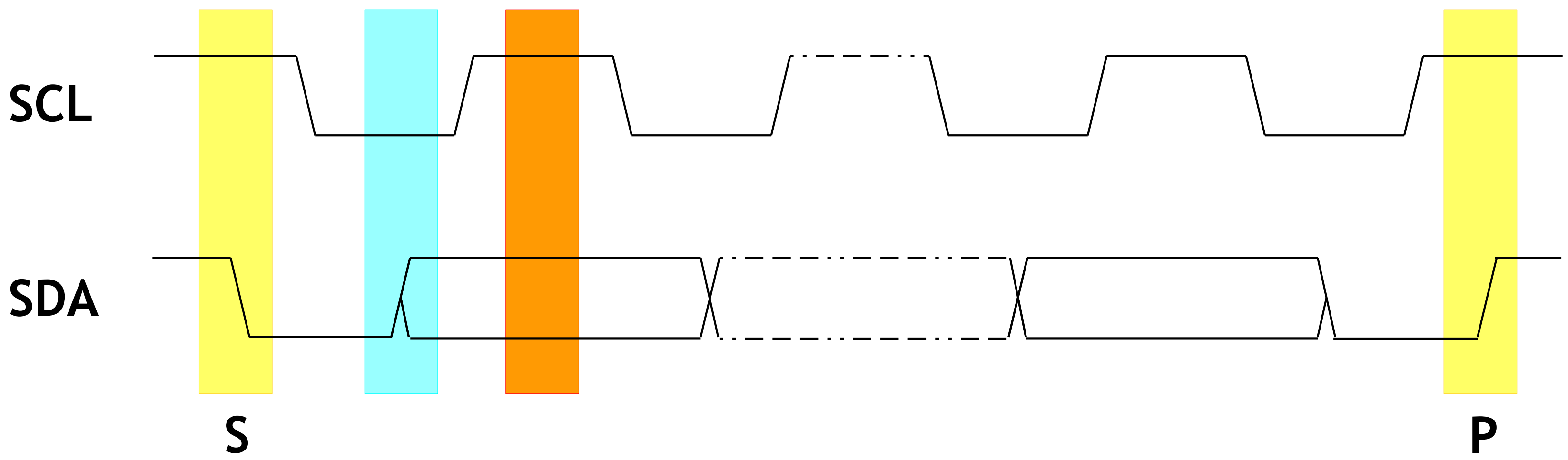
- Two-wired Interface
  - SDA
  - SCL
- Wired-AND
- Conditions and Data Validity
- Transmission

I²C
Signals – Wired-AND



I²C

Signals – Conditions and Data Validity



Data Write

Data Read

Conditions

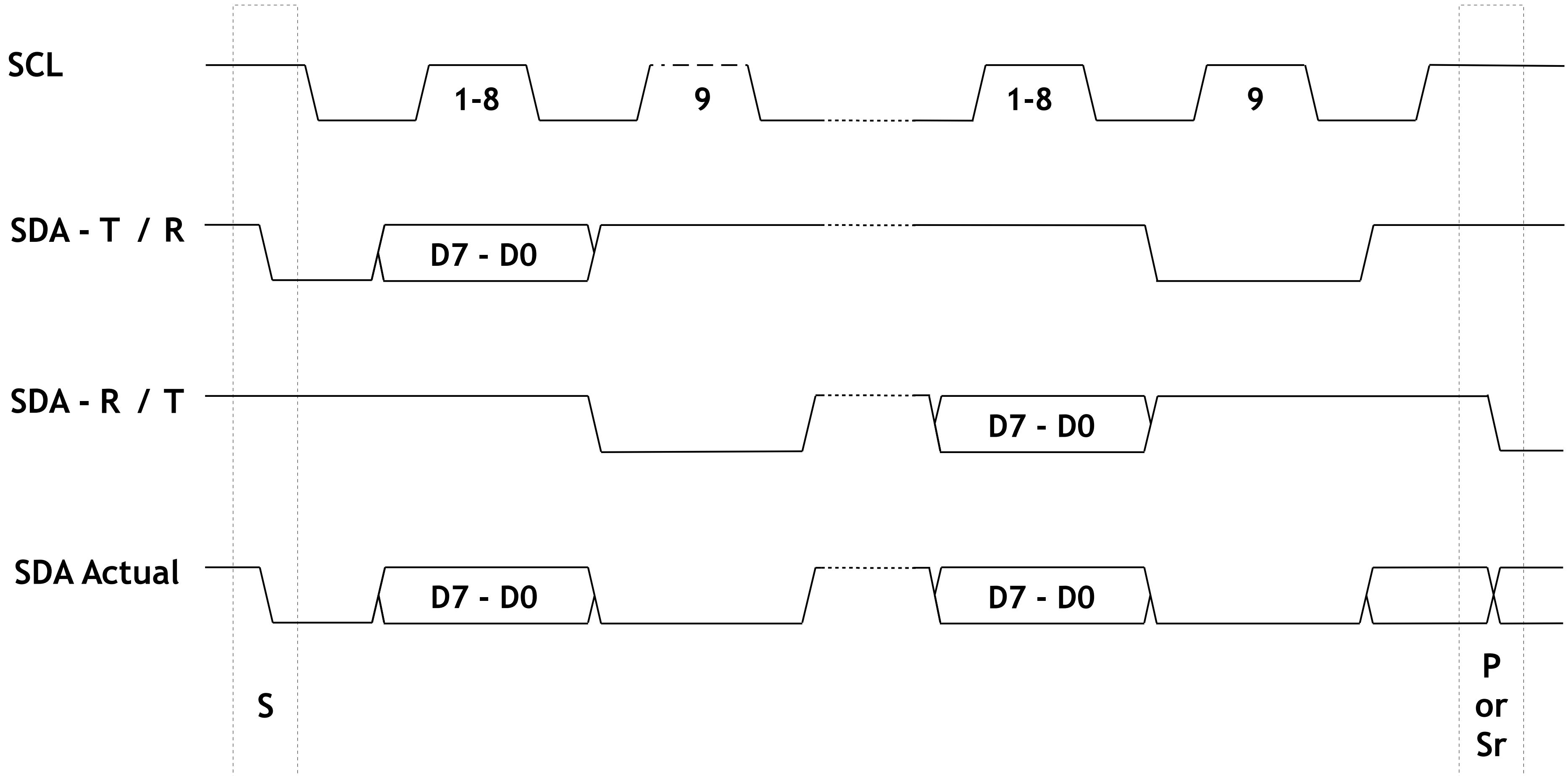
I²C

Signals – Transmission

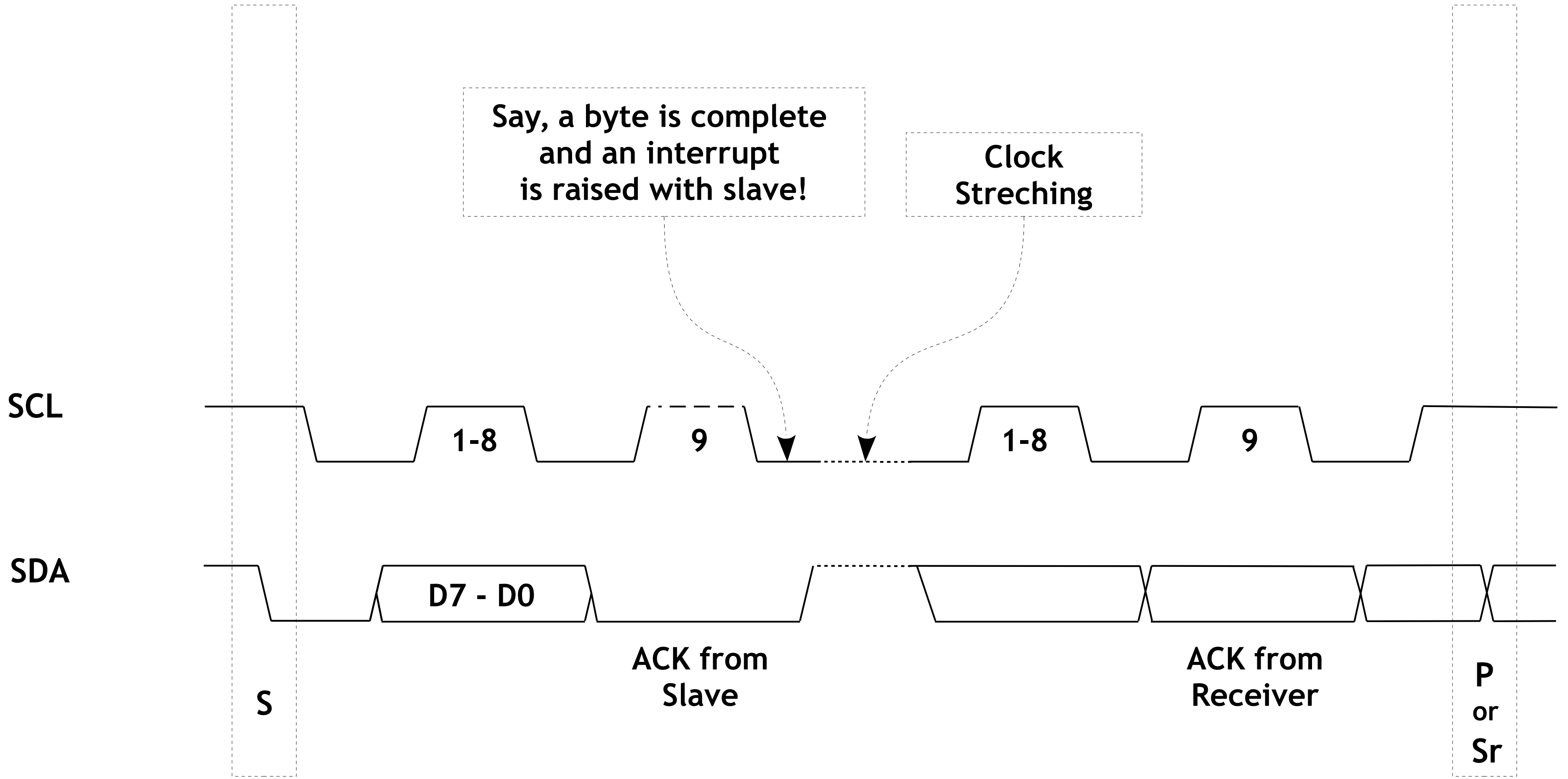
- Data on SDA
- Clocking on SCL
- Clock Synchronization
- Data Arbitration

I^2C

Signals – Data on SDA

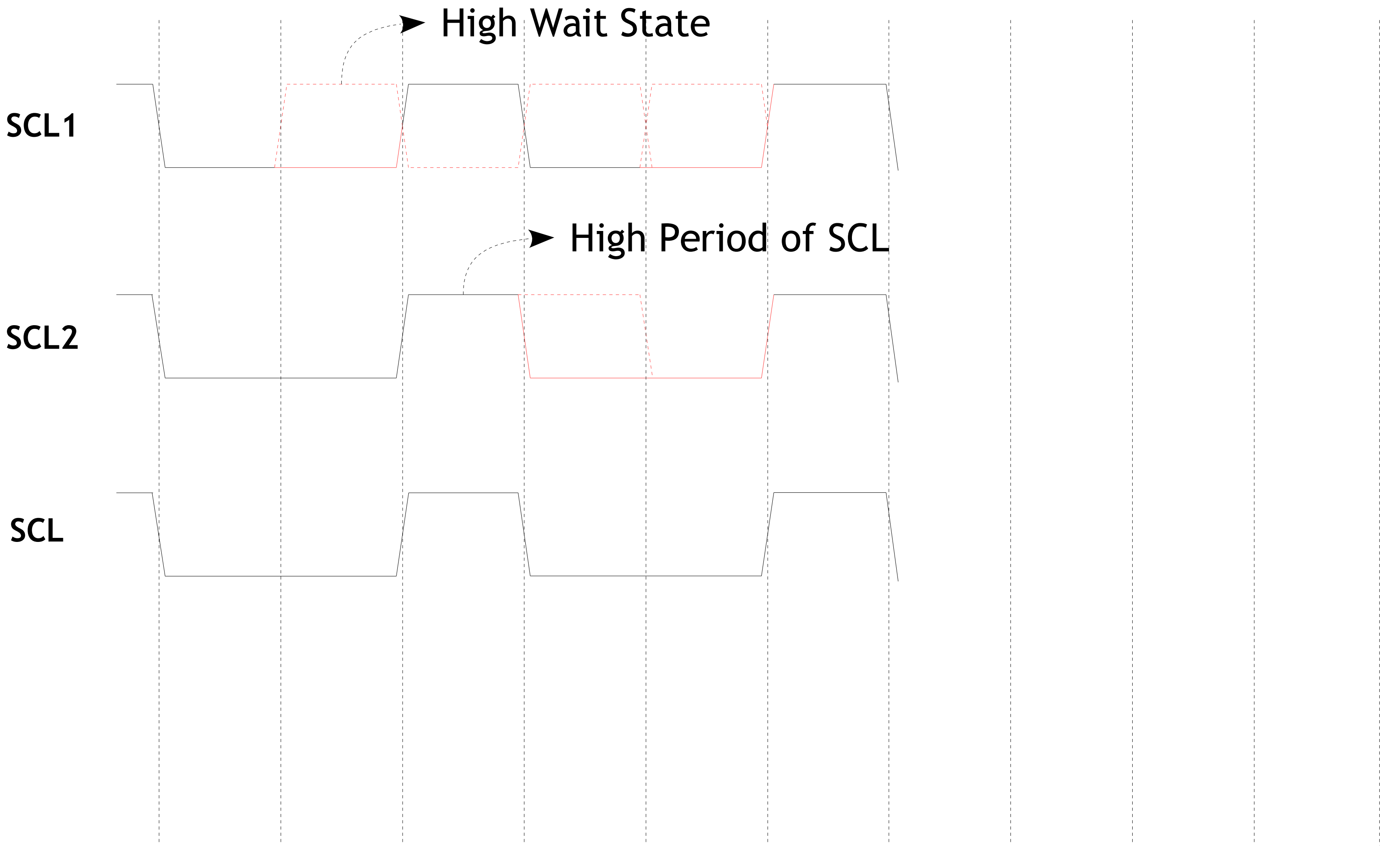


I²C
Signals – Clocking on SCL



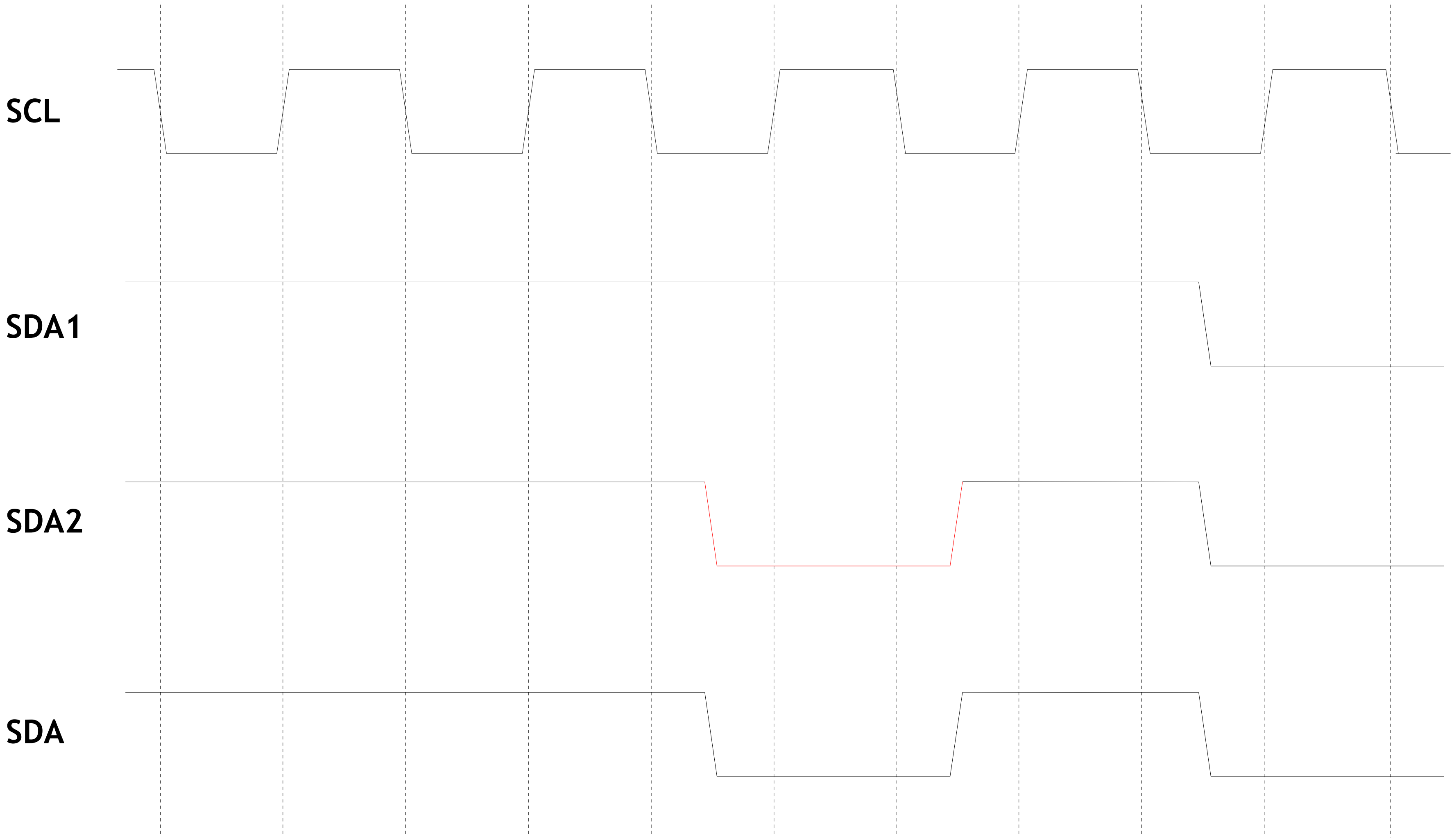
I^{2}C

Signals – Clock Synchronization



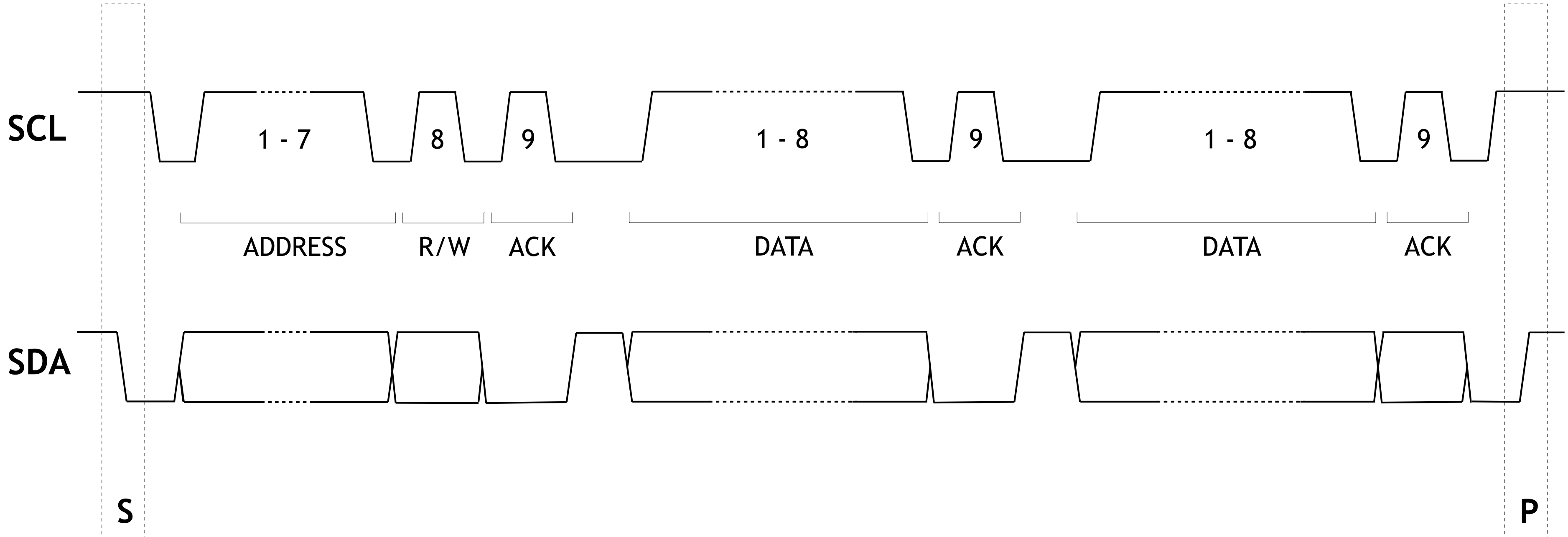
I^{2}C

Signals – Data Arbitration



I^{2}C

A Complete Data Transfer



I²C
Bus Speeds

- Bidirectional Bus
  - Standard Mode - 100 Kbit/s
  - Fast Mode - 400 Kbits/s
  - Fast Mode Plus - 1 Mbits/s
  - High Speed Mode - 3.4 Mbits/s
- Unidirectional Bus
  - Ultra Fast Mode – 5 Mbits/s
    - Uses Push-Pull Drivers (No Pullups)

Controller Area Network

Controller Area Network

- Introduction to CAN
- Basic Concepts
- Message Transfer
- Error Handling
- Fault Confinement

CAN
Introduction

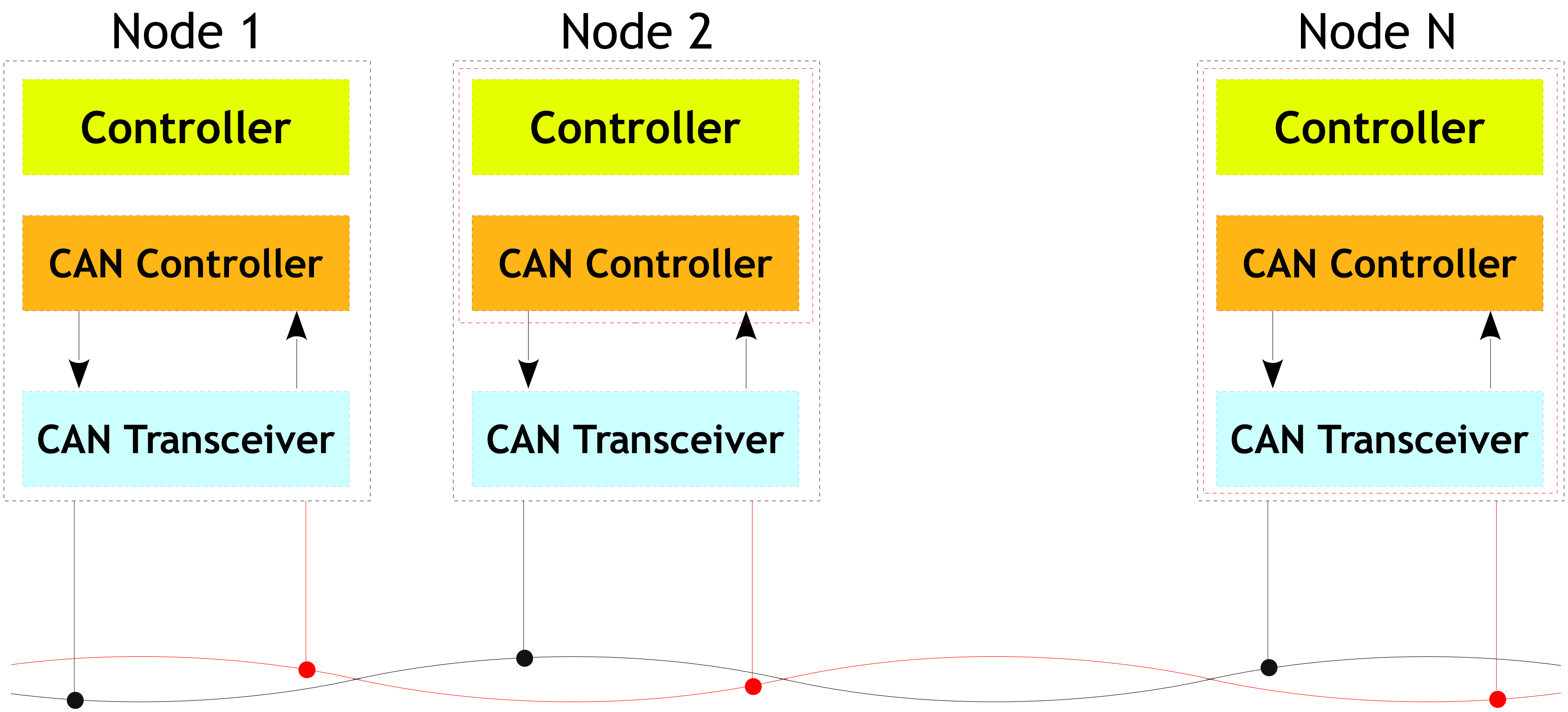
- Asynchronous
- Half Duplex
- Multi Master / Slave

CAN

Basic Concepts

- Example
- Versions
- Absence of node addressing
  - Message identifier specifies contents and priority
  - Lowest message identifier has highest priority
- Non-destructive arbitration system by CSMA with collision detection
- Simple Transmission Medium
  - Twisted pair – CAN H and CAN L
- Properties
- Layered Architecture

CAN
Basic Concepts - Example



CAN

Basic Concepts - Versions

NOMENCLATURE	STANDARD	MAX SIGNALING RATE	IDENTIFIER
Low Speed CAN	ISO 11519	125 kbps	11 bit
CAN 2.0A	ISO 11898:1993	1 Mbps	11 bit
CAN 2.0B	ISO 11898:1995	1 Mbps	29 bit

CAN

Basic Concepts - Properties

- Prioritization of Messages
- Guarantee of Latency Times
- Configuration Flexibility
- Multicast Reception with Time Synchronization
- System wide Data Consistency
- Multi master
- Error Detection and Error Signaling
- Automatic Retransmission
- Distinction between temporary errors and permanent failures of nodes and autonomous switching off of defect nodes

CAN

Basic Concepts - Layered Architecture

7	Application	Application
6	Presentation	Presentation
5	Session	Session
4	Transport	Transport
3	Network	Network
2	Data Link	Data Link
1	Physical	Physical
	OSI Model	

CAN

Basic Concepts - Layered Architecture

7 **Application**

6 **Presentation**

5 **Session**

4 **Transport**

3 **Network**

2 **Data Link**

1 **Physical**

OSI Model

LLC:
Acceptance Filtering
Overload Notifications
Recovery Managment

MAC:
Data En / Decapsulation
Frame Coding (Stuffing, Destuffing)
Medium Access Managment
Error Detection
Error Signalling
Acknowledgement
Serialization / Deserialization

Bit Encoding / Decoding
Bit Timing
Synchronization

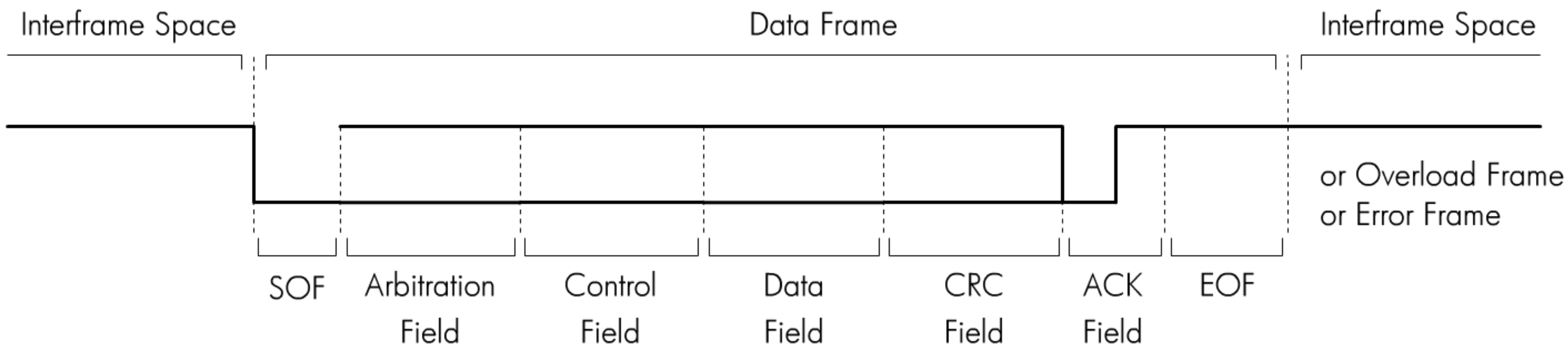
Driver / Receiver Characteristics

ΣMERTXE

CAN
Message Transfer

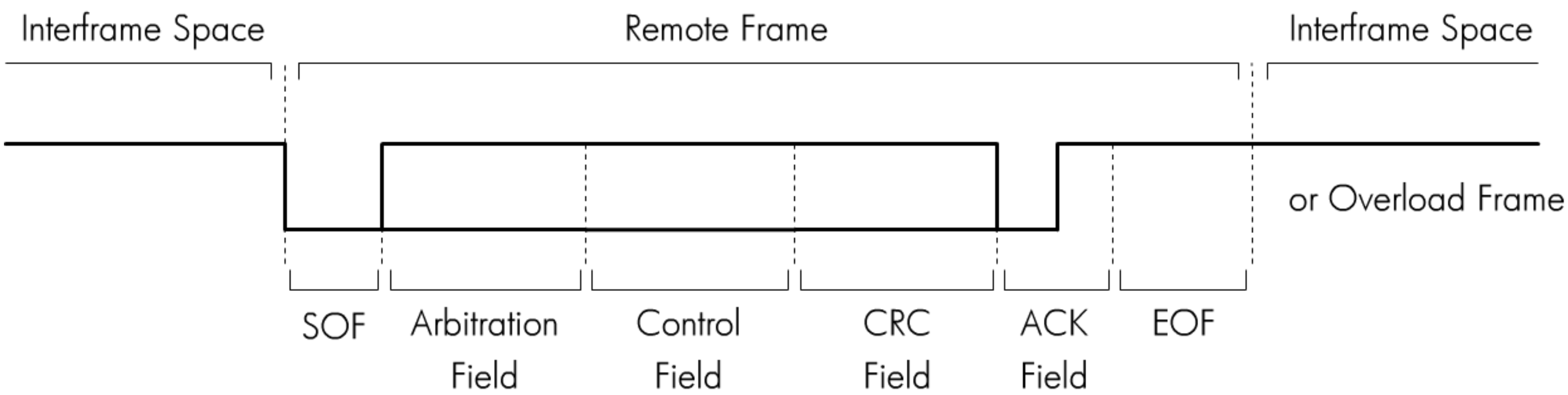
- Frame Formats
  - Standard Frame - 11 bits Identifiers
  - Extended Frame - 29 bits Identifiers
- Frame Types
  - Data Frame
  - Remote Frame
  - Error Frame
  - Overload Frame
- Frame Fields

CAN
Message Transfer – Data Frame



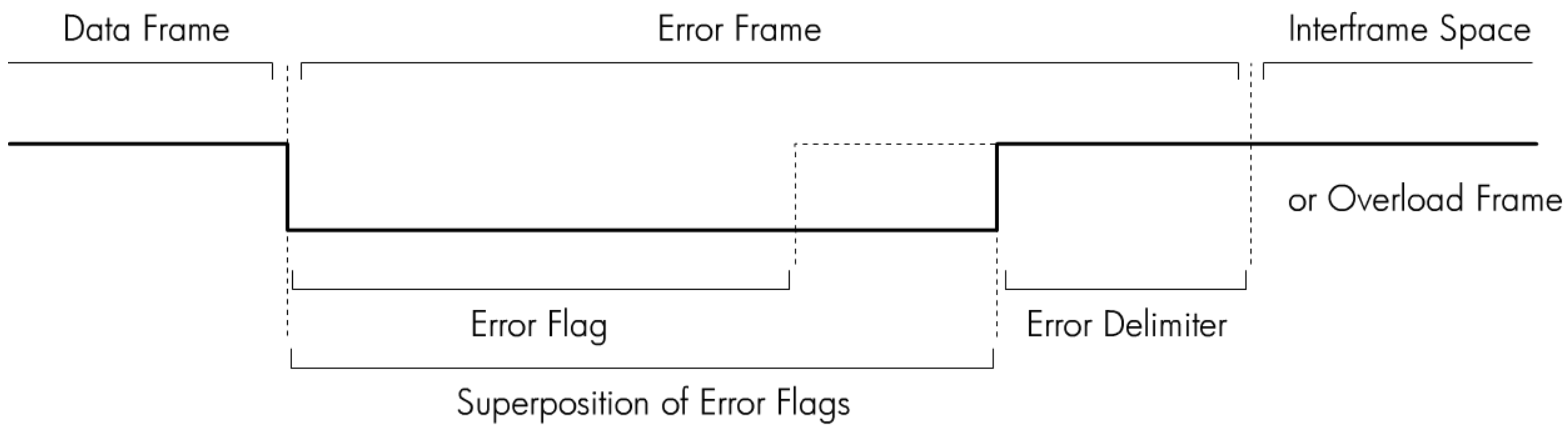
- A data frame consists of seven fields: start-of-frame, arbitration, control, data, CRC, ACK, and end-of-frame.

CAN
Message Transfer – Remote Frame



- Used by a node to request other nodes to send certain type of messages
- Has six fields as shown in above figure
  - These fields are identical to those of a data frame with the exception that the RTR bit in the arbitration field is **recessive** in the remote frame.

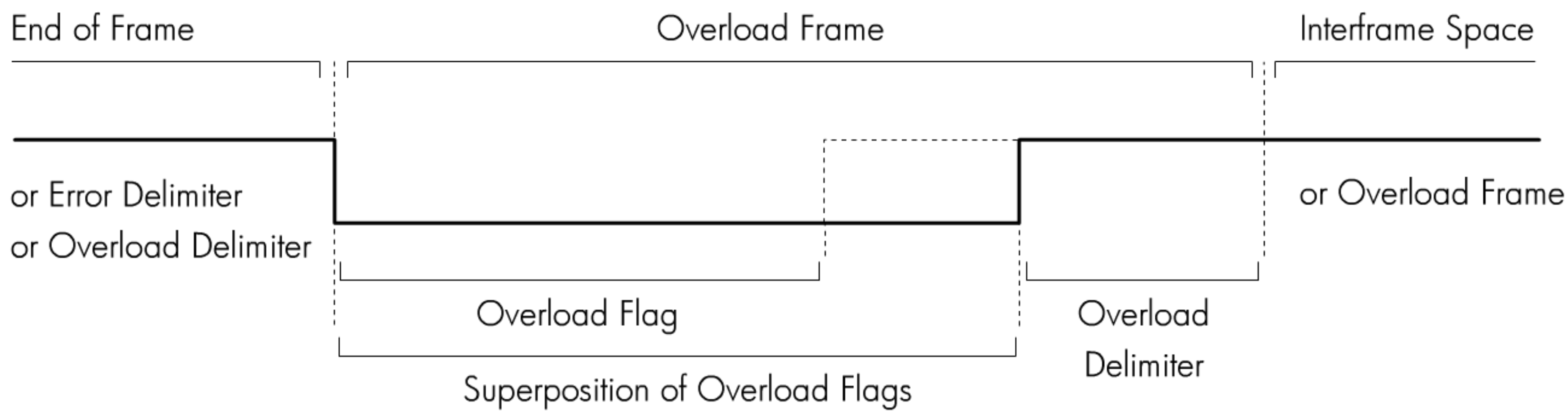
CAN
Message Transfer – Error Frame



- This frame consists of two fields.
  - The first field is given by the superposition of error flags contributed from different nodes.
  - The second field is the error delimiter.
- Error flag can be either active-error flag or passive-error flag.
  - Active error flag consists of six consecutive dominant bits.
  - Passive error flag consists of six consecutive recessive bits.
- The error delimiter consists of eight recessive bits.

CAN

Message Transfer – Overload Frame



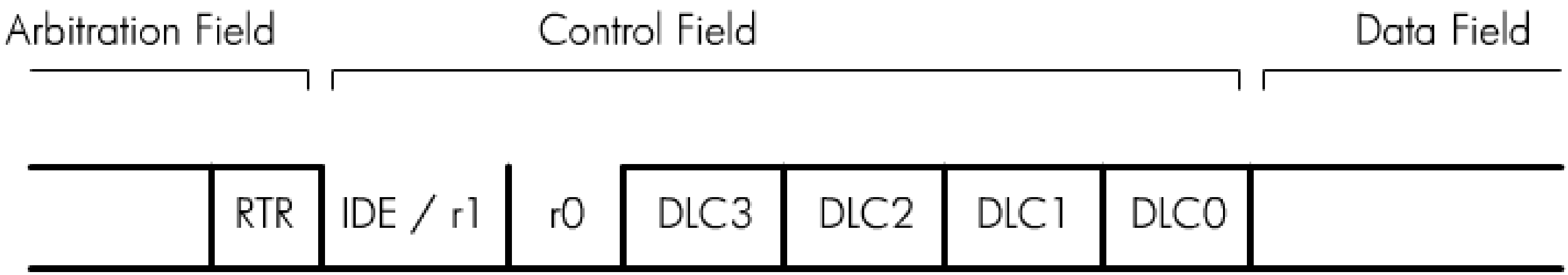
- Consists of two bit fields: overload flag and overload delimiter
- Three different overload conditions lead to the transmission of the overload frame:
  - Internal conditions of a receiver require a delay of the next data frame or remote frame.
  - At least one node detects a dominant bit during intermission.
  - A CAN node samples a dominant bit at the eighth bit (i.e., the last bit) of an error delimiter or overload delimiter.
- Format of the overload frame is shown in above fig
- The overload flag consists of six dominant bits.
- The overload delimiter consists of eight recessive bits.

CAN
Message Transfer – Frame Fields

- Control Field
- Arbitration Field
- Data Field
- CRC Field
- ACK Field

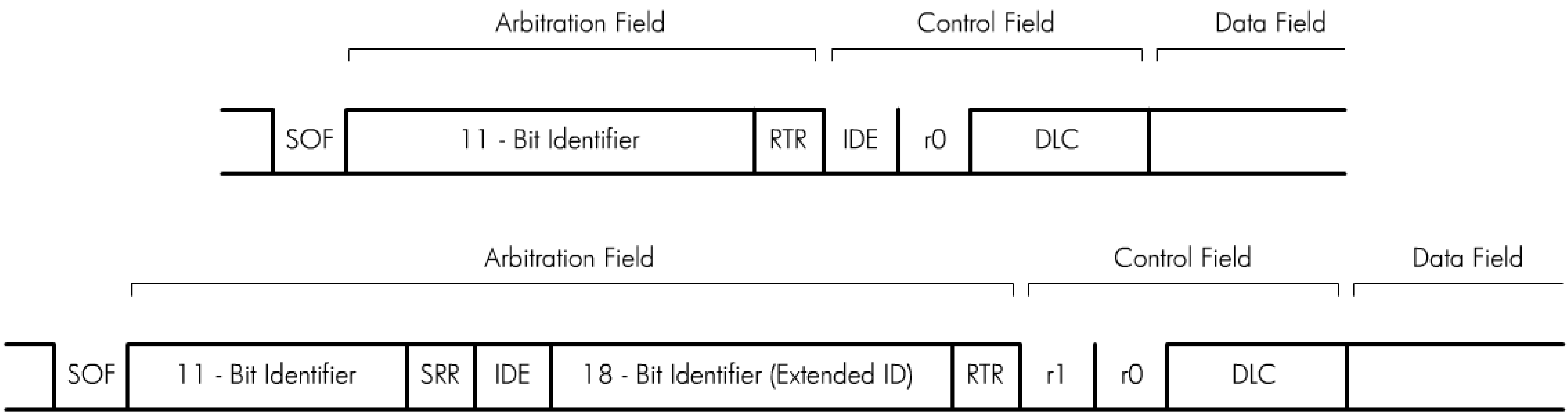
CAN

Frame Fields – Control Field



- The first bit is IDE bit for the standard format but is used as reserved bit r1 in extended format.
- r0 is reserved bit.
- DLC3…DLC0 stands for data length and can be from 0000 (0) to 1000 (8).

CAN
Frame Fields – Arbitration Field



- The identifier of the standard format corresponds to the base ID in the extended format.
- The RTR bit is the remote transmission request and must be 0 in a data frame.
- The SRR bit is the substitute remote request and is recessive.
- The IDE field indicates whether the identifier is extended and should be recessive in the extended format.
- The extended format also contains the 18-bit extended identifier.

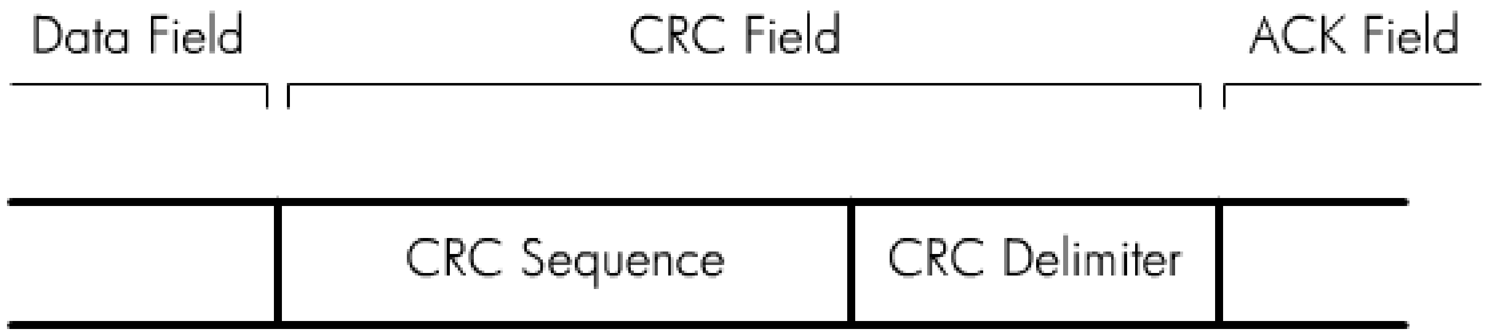
CAN

Frame Fields – Data Field

- May contain 0 to 8 bytes of data

CAN

Frame Fields – CRC Field



- It contains the 16-bit CRC sequence including CRC delimiter.
- The CRC delimiter is a single **recessive** bit.

CAN
Frame Fields – Ack Field

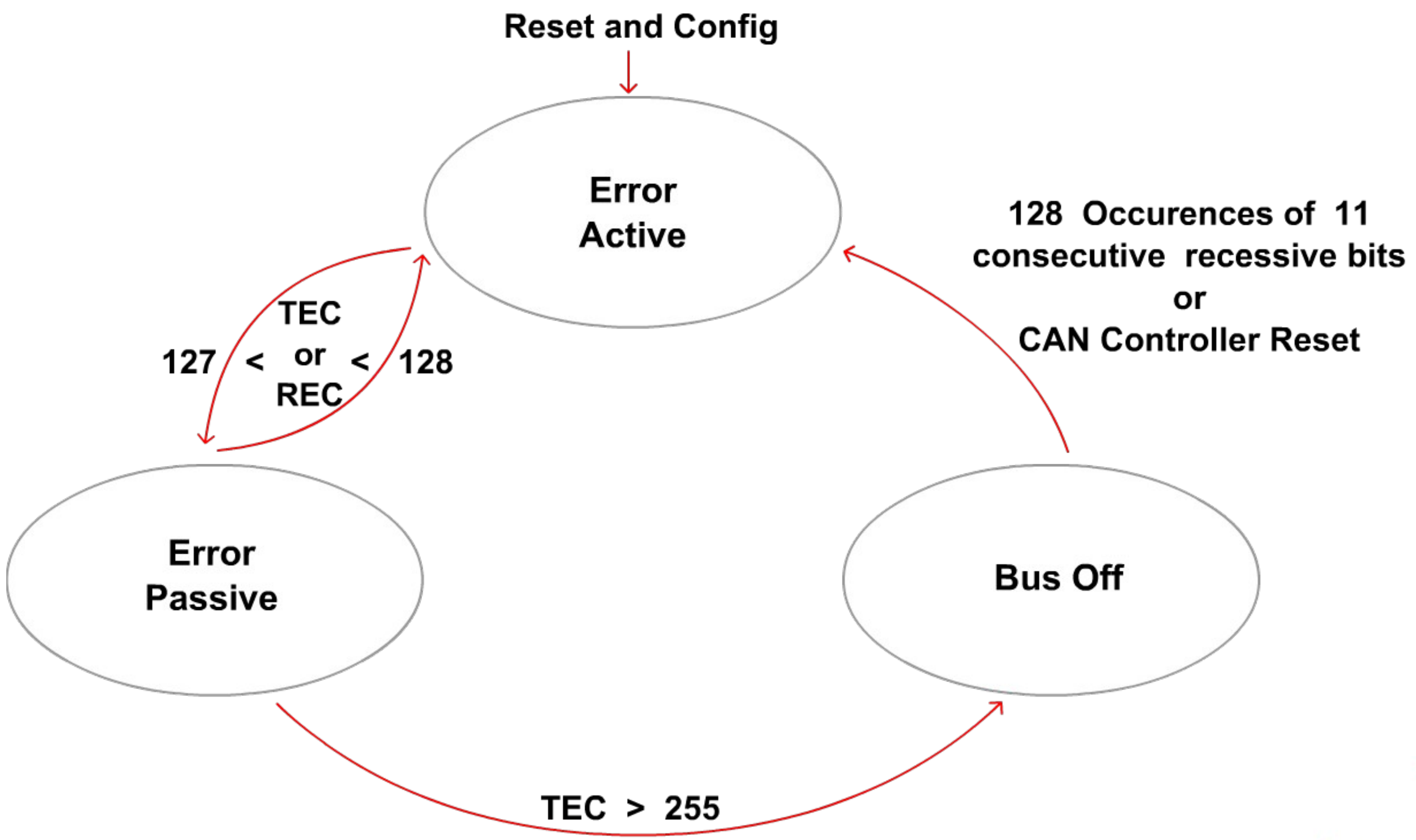
- Consists of two bits
- The first bit is the **acknowledgement bit**.
- This bit is set to recessive by the transmitter, but will be reset to dominant if a receiver acknowledges the data frame.
- The second bit is the **ACK delimiter** and is recessive.

CAN
Error Handling

- Error Detection
  - Bit Error
  - Stuff Error
- Error Signaling
  - CRC Error
  - Form Error
  - Acknowledgment Error

CAN
Fault Confinement

- Counters
  - Transmit Error Counter & Receive Error Counter



Thank You