ECE3411 – Fall 2015

Lecture 7b.

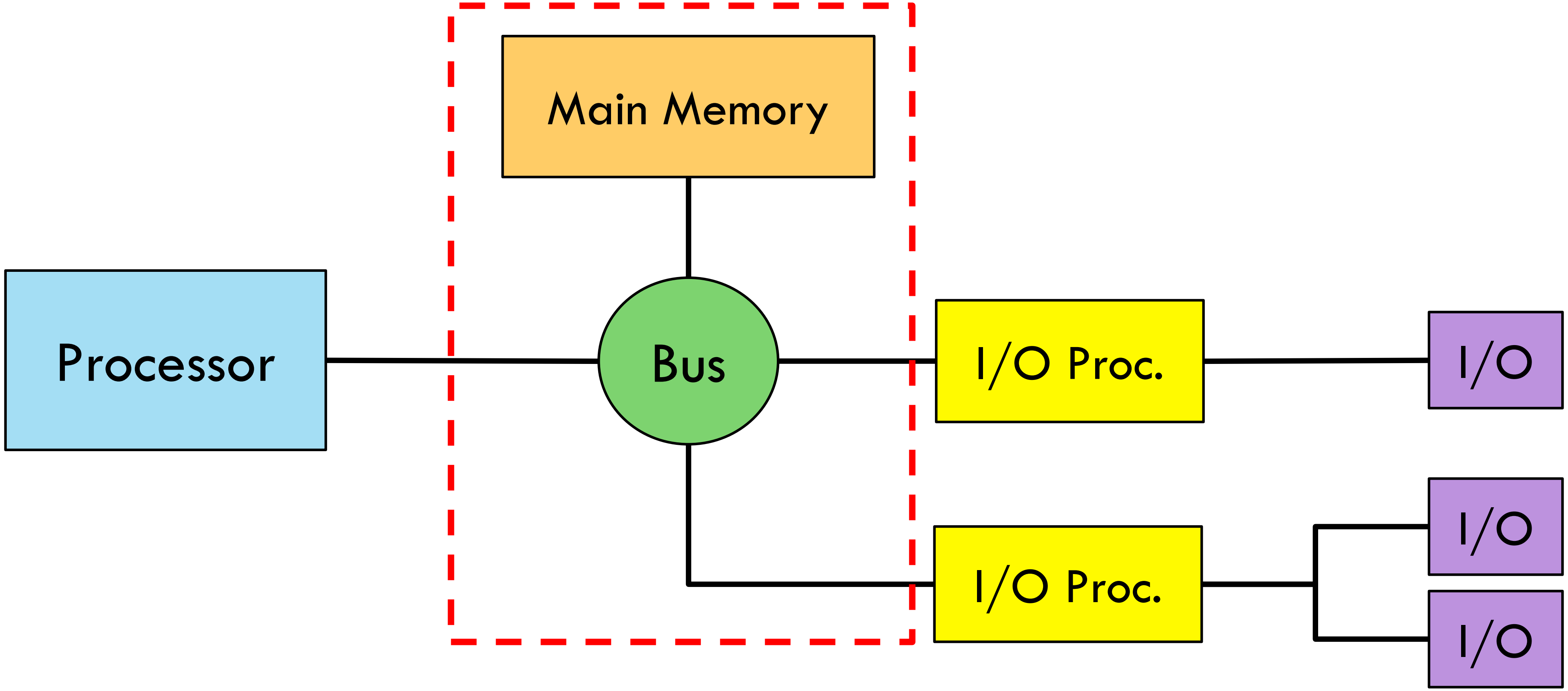
Advanced topics in Embedded Systems Design

Marten van Dijk, Syed Kamran Haider
Department of Electrical & Computer Engineering
University of Connecticut
Email: {vandijk, syed.haider}@engr.uconn.edu

UCONN

Some of these slides are extracted or copied from TTK4155:
"Industrial & Embedded System Design" offered at NTNU, Norway.

Basic Computer Components

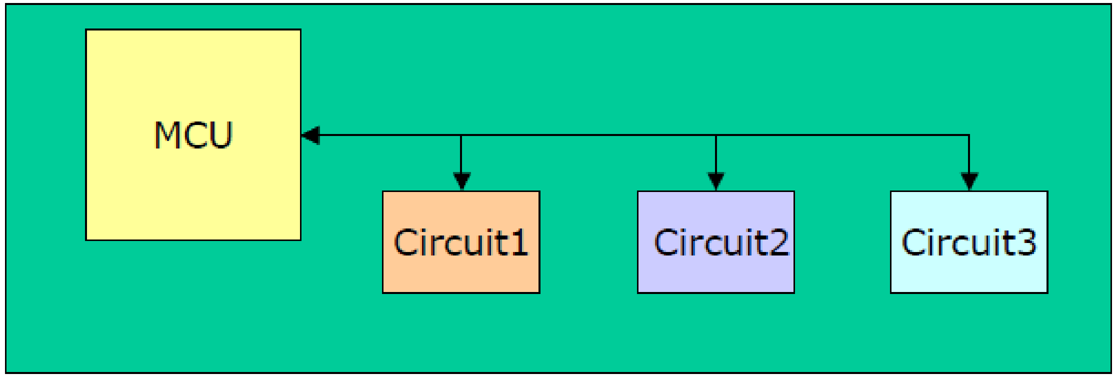


Bus and Communication Interfaces

- Parallel Bus Systems
  - Processor Buses – AVR etc.
  - Industrial Buses
    - VMEbus
    - CompactPCI
    - PC/104
    - …
- Serial Local Buses
  - SPI
  - MicroWire
  - I2C
  - 1-Wire
- Serial Lines (1 to 1, 1 to N)
  - UART
  - RS-232C
  - RS-422
  - USB
- Networks (N to M)
  - CAN
  - RS-485
  - LAN/Ethernet
- Wireless Communication
  - IR/IrDA
  - ISM
  - WiFi
  - Bluetooth
  - Zigbee

Serial synchronous interfaces

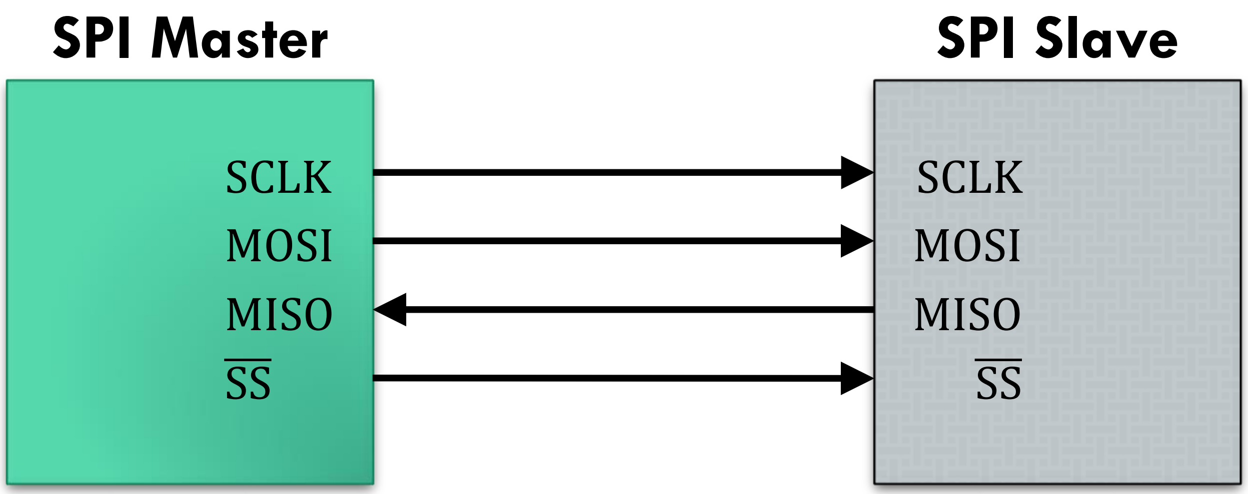
- Local serial interconnection of microcontrollers and peripheral circuits/functions
- Required features:
  - Low complexity
  - Low to medium data rate
  - Small physical footprint/few pins
  - Short distances
  - Low cost

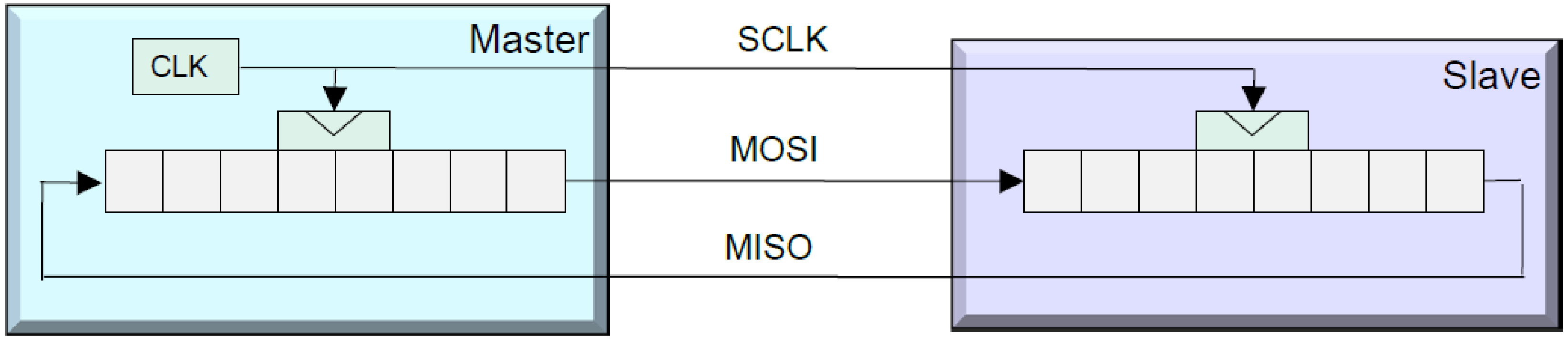


- Most MCUs have built-in peripheral units for communicating with external circuits, e.g. ATmegaAVR (SPI and TWI (I2C))
- Great abundance of different types of peripheral circuits that implements synchronous serial interfaces (Flash, EEPROM, AD, DA, RTC, Display drivers, sensors etc.)

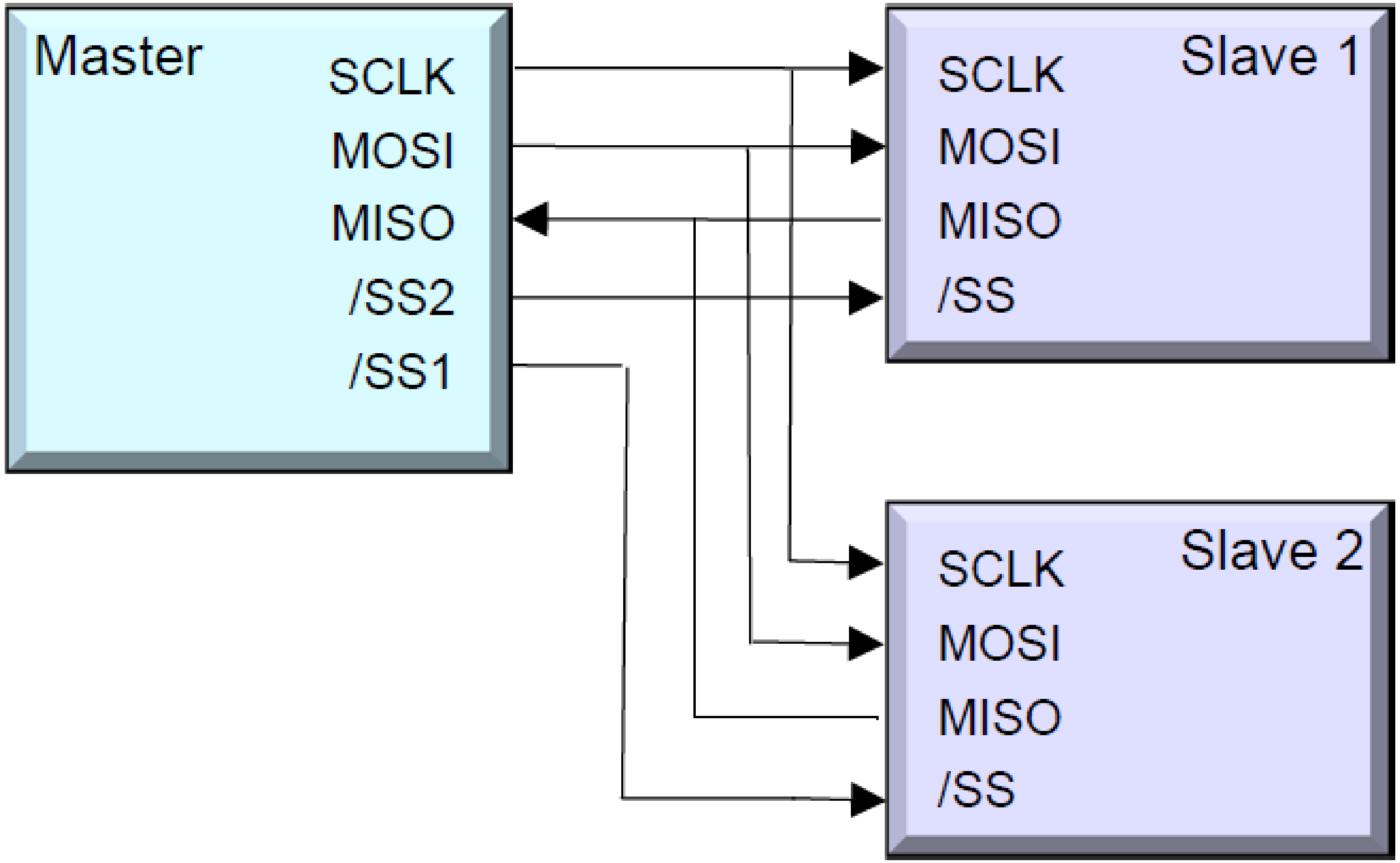
SPI: Serial Peripheral Interface

- Synchronous Data Transfer
- Master/Slave configuration
- 4-Line Bus
- Full Duplex operation



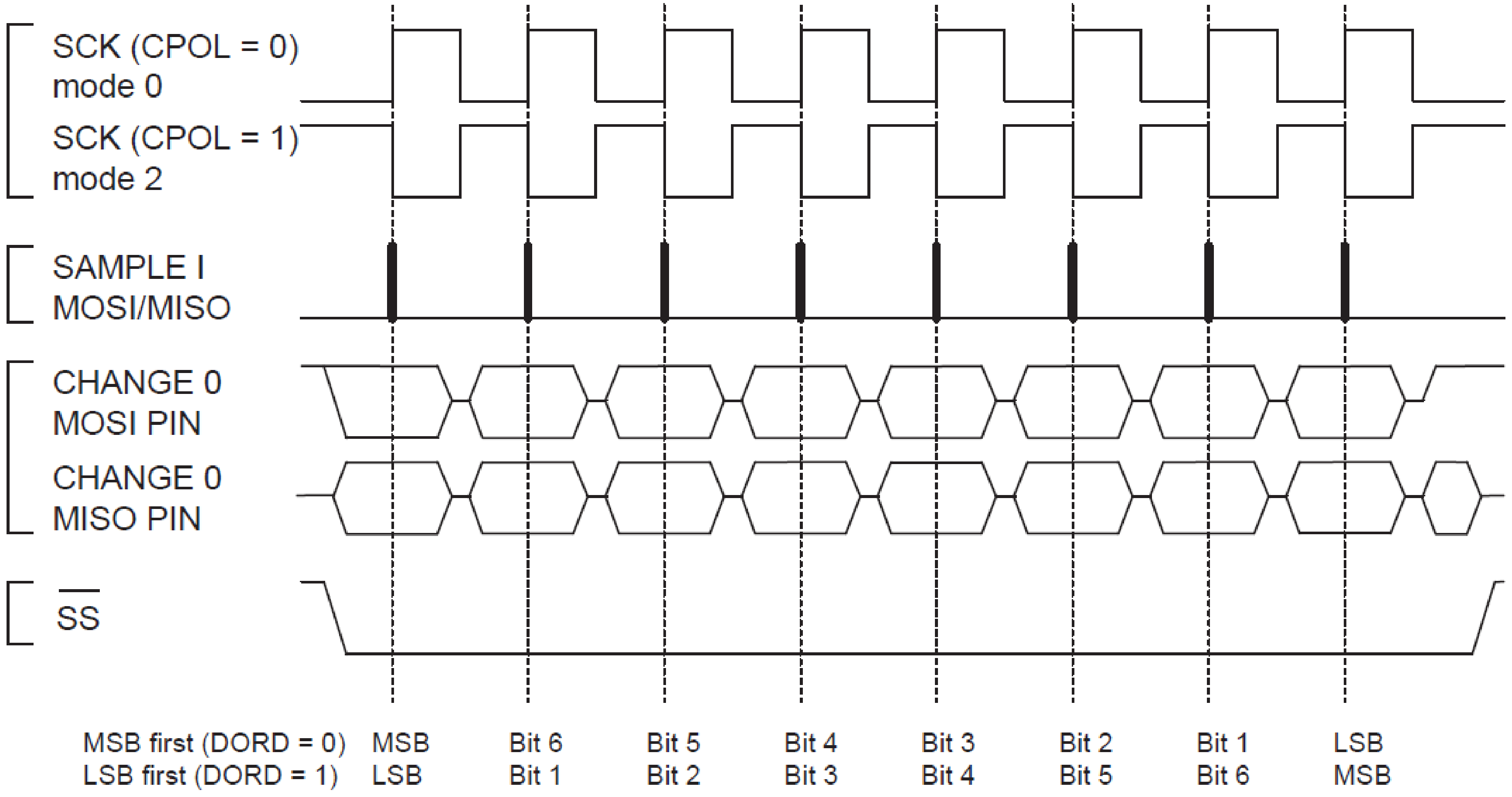


SPI Master with Multiple Slaves



SPI Frame Transfer

Figure 18-3. SPI Transfer Format with CPHA = 0

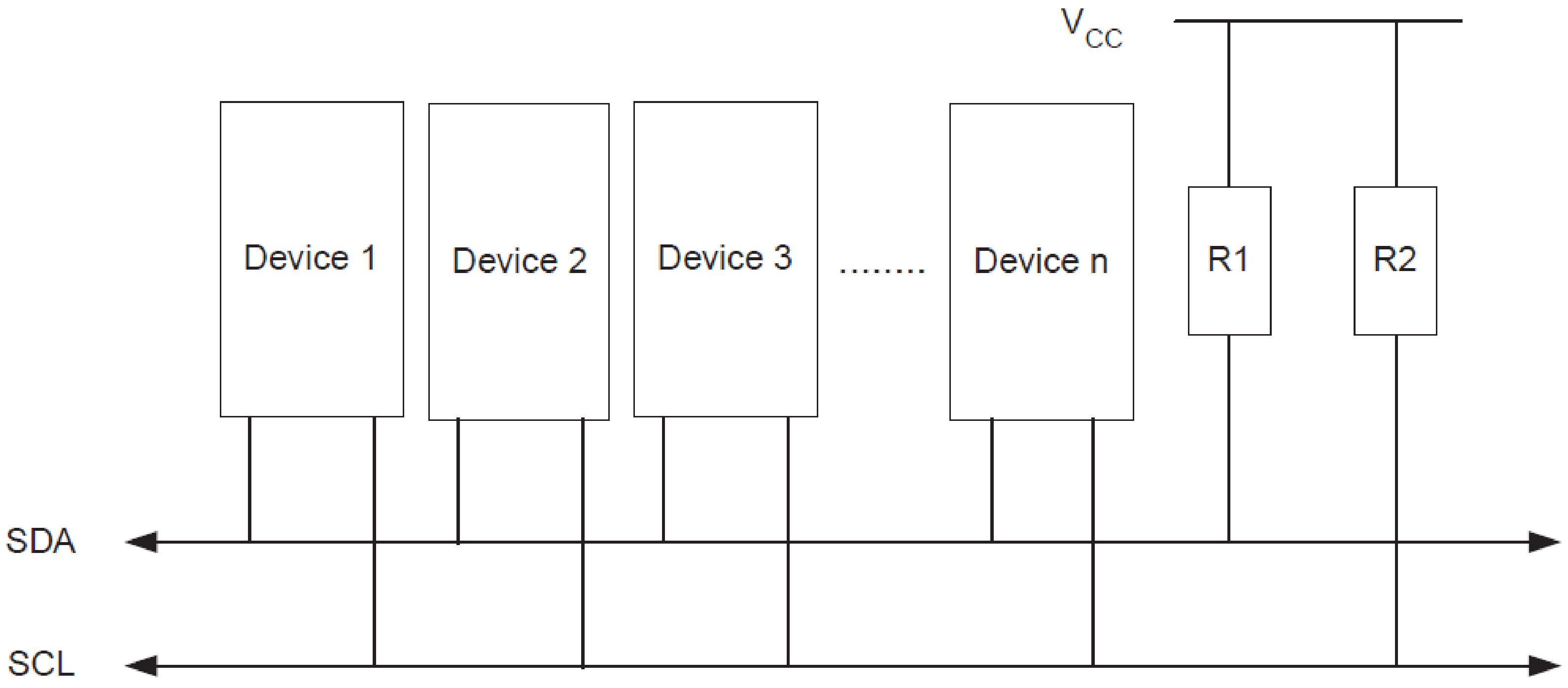


MicroWire (μWire)

- Essentially a subset of SPI
- SPI mode 0 → (CPOL, CPHA) = (0, 0)
- Often found in half duplex “three-wire mode”
- Common bi-directional serial data line → only three wires needed (SIO, SCLK, CS)
- Used in e.g. RTCs and serial EEPROMs

I^2C: Inter Integrated Circuit bus

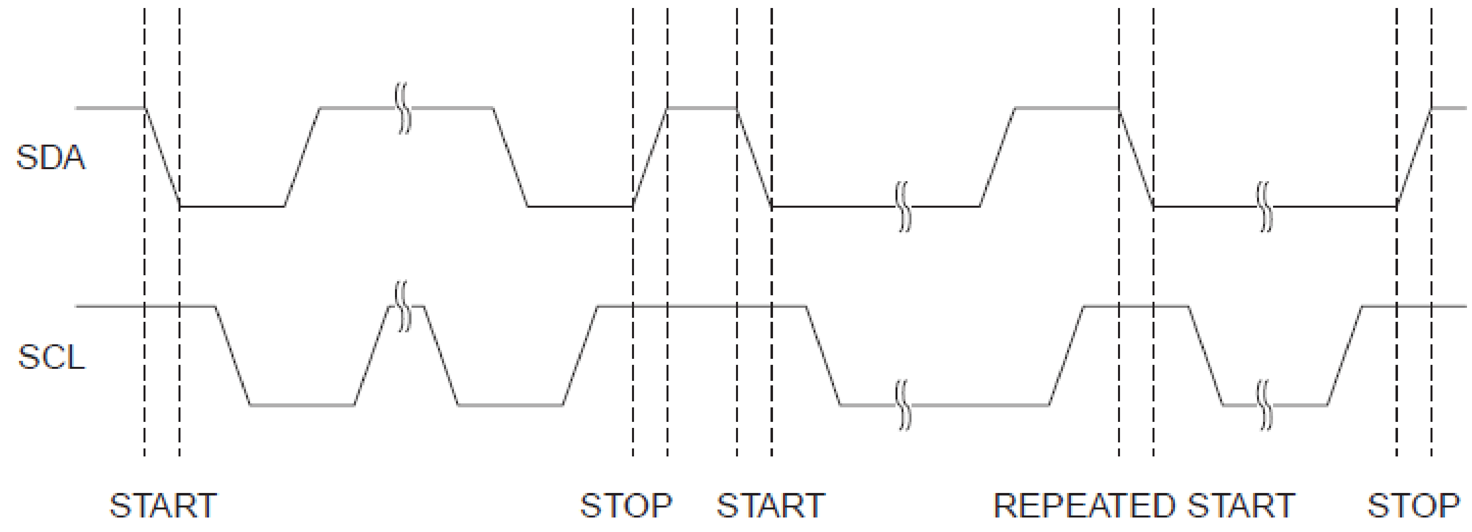
- Also known as Two Wire Interface (TWI)
- Allows up to 128 different devices to be connected using only two bi-directional bus lines, one for clock (SCL) and one for data (SDA).
- All devices connected to the bus have individual addresses.



I^2C START and STOP Conditions

- START and STOP conditions are signaled by changing the level of the SDA line when the SCL line is high.
- When a new START condition is issued between a START and STOP condition, this is referred to as a REPEATED START condition

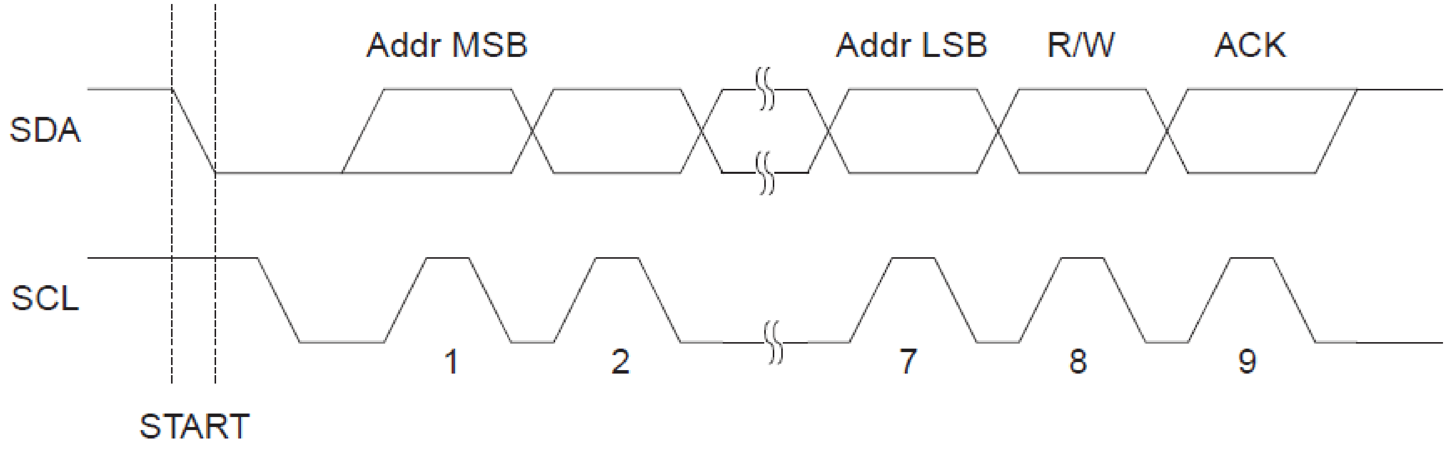
Figure 21-3. START, REPEATED START and STOP conditions



I^2C Address Packet Format

- All address packets transmitted on the TWI bus are 9 bits long:
  - 7 address bits, one READ/WRITE control bit and an acknowledge bit.
- When a Slave recognizes that it is being addressed, it should acknowledge by pulling SDA low in the ninth SCL (ACK) cycle.
- The Master can then transmit a STOP condition, or a REPEATED START condition to initiate a new transmission.

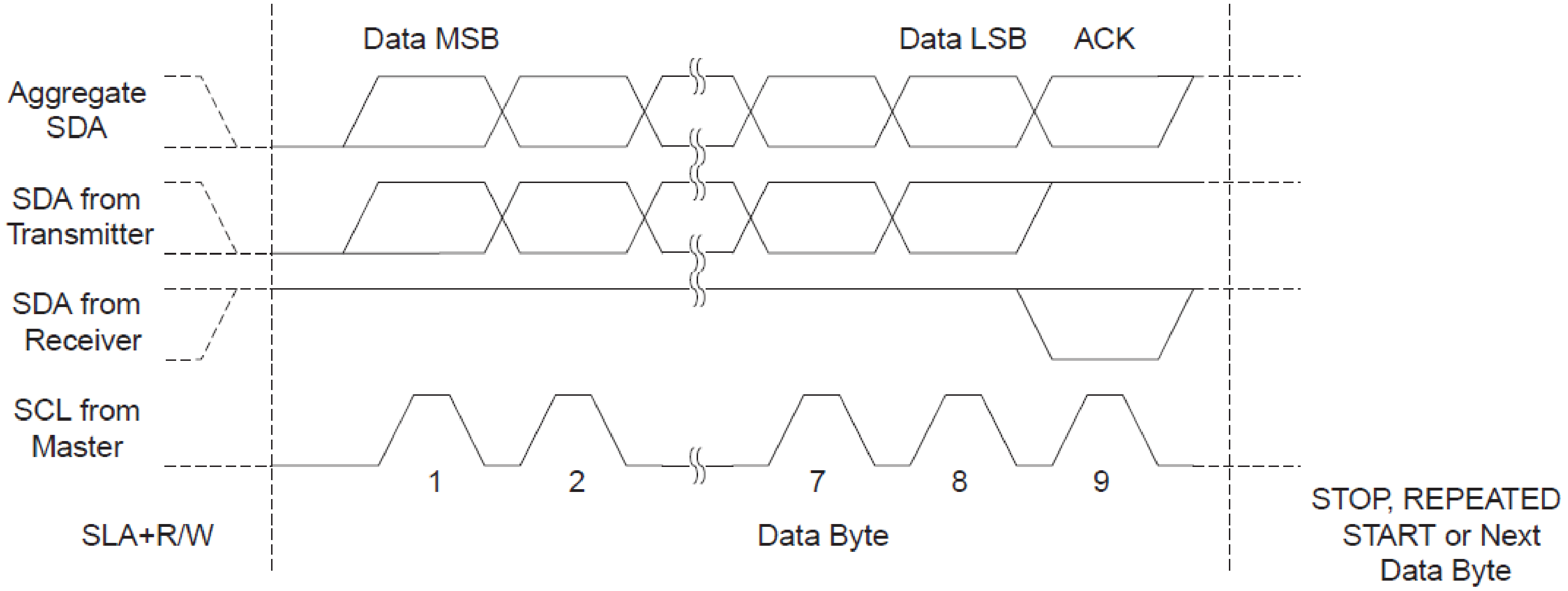
Figure 21-4. Address Packet Format



I^2C Data Packet Format

- All data packets transmitted on the TWI bus are 9 bits long:
  - One data byte and one acknowledge bit.
- An Acknowledge (ACK) is signaled by the Receiver pulling the SDA line low during the ninth SCL cycle. If the Receiver leaves the SDA line high, a NACK is signaled.

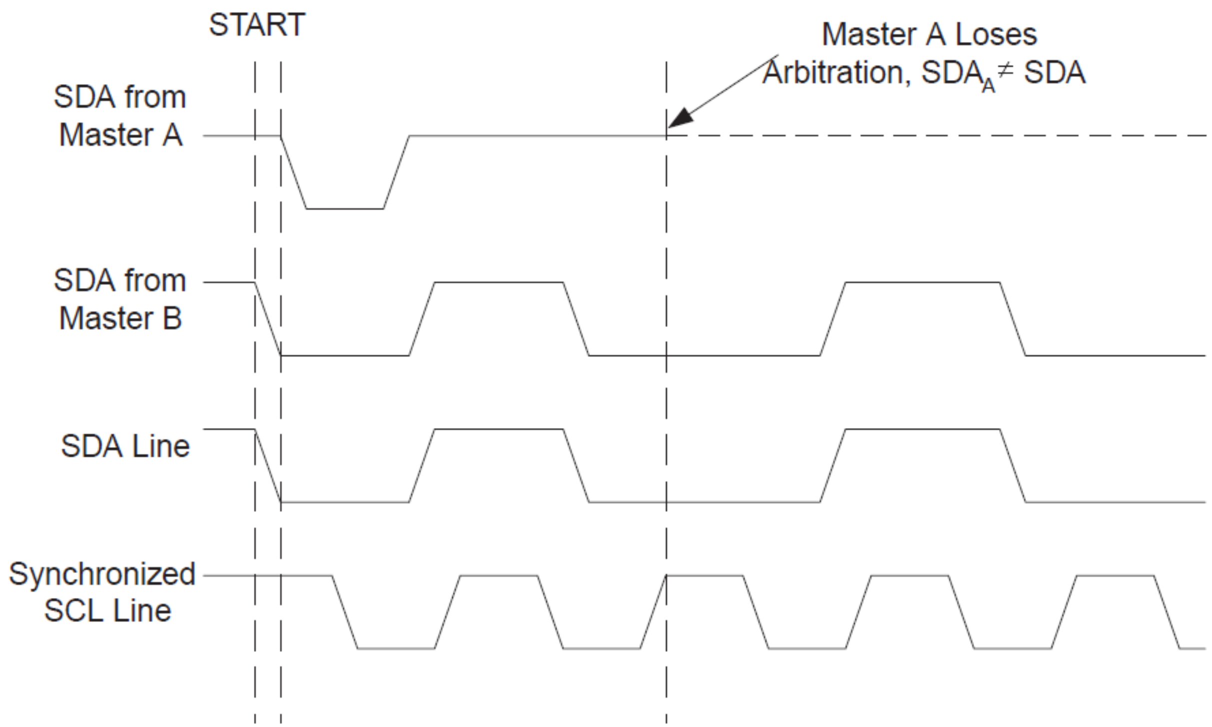
Figure 21-5. Data Packet Format



I²C Bus Arbitration

- Arbitration is carried out by all masters continuously monitoring the SDA line after outputting data.
- If the value read from the SDA line does not match the value the Master had output, it has lost the arbitration.

Figure 21-8. Arbitration Between Two Masters



Bus and Communication Interfaces

- Parallel Bus Systems
  - Processor Buses – AVR etc.
  - Industrial Buses
    - VMEbus
    - CompactPCI
    - PC/104
    - …
- Serial Local Buses
  - SPI
  - MicroWire
  - I2C
  - 1-Wire
- Serial Lines (1 to 1, 1 to N)
  - UART
  - RS-232C
  - RS-422
  - USB
- Networks (N to M)
  - CAN
  - RS-485
  - LAN/Ethernet
- Wireless Communication
  - IR/IrDA
  - ISM
  - WiFi
  - Bluetooth
  - Zigbee

CAN – Controller Area Network

Industrial network protocol (automotive fieldbus (Bosch))

- CAN is based on a message oriented broadcast communication mechanism.
- Well suited for interconnection of distributed embedded computers in industrial environments.
- Robust against sporadic and persistent disturbances.
- CAN messages are identified by using a message identifier.
  - Message identifier has to be unique within the whole network
- Message identifier defines not only the content but also the priority of the message.
  - The identifier with the lowest binary number has the highest priority.
- Acceptable performance for monitoring and control applications:
  - 1 Mbps@40 meters
  - 40 kbps@1000 meters

CAN Frame Format & Types

<u>Information transmitted in frames of different formats and types</u>

Two frame formats:

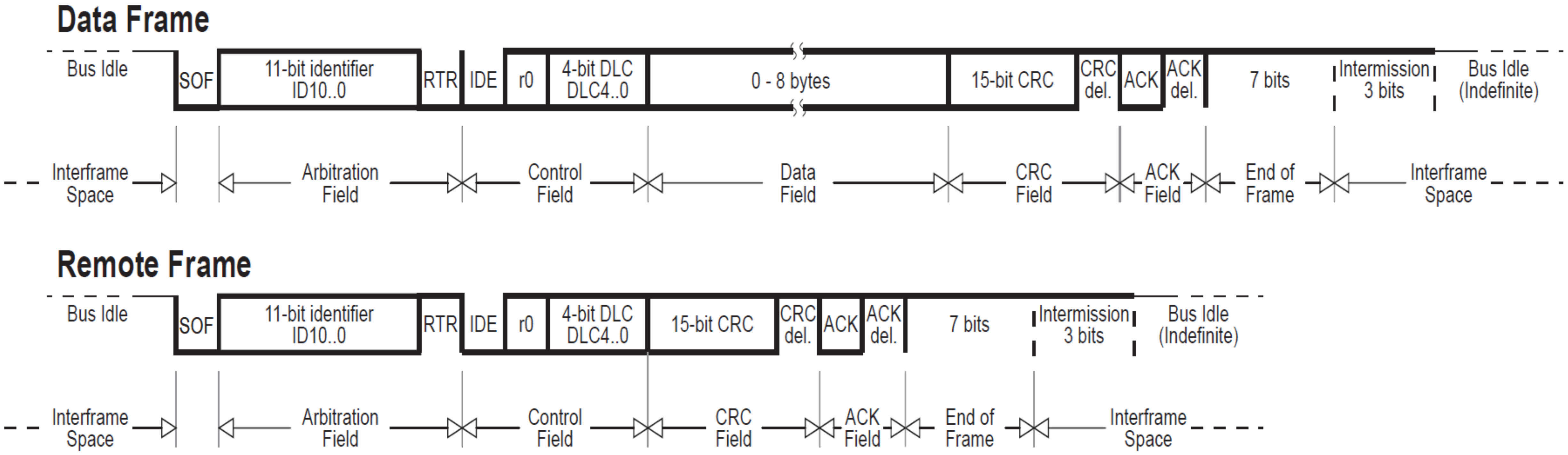
- Standard – 11 bits message identifier (CAN2.0A)
- Extended – 29 bits message identifier (CAN2.0B)

Four frame types:

- DATA FRAME
- REMOTE FRAME
- ERROR FRAME
- OVERLOAD FRAME

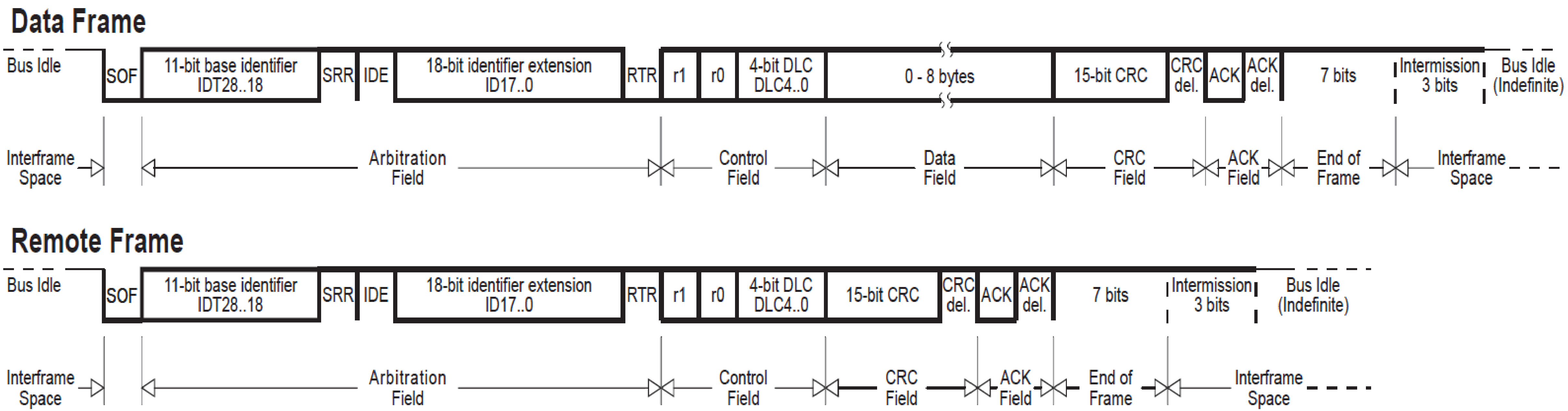
Messages are identified by a unique number (message identifier)

CAN Standard Frame Format



SOF – Start of Frame,
RTR – Remote Transmission Request, IDE – Identifier Extension Bit
SRR – Substitute Remote Request bit, DLC – Data Length Code

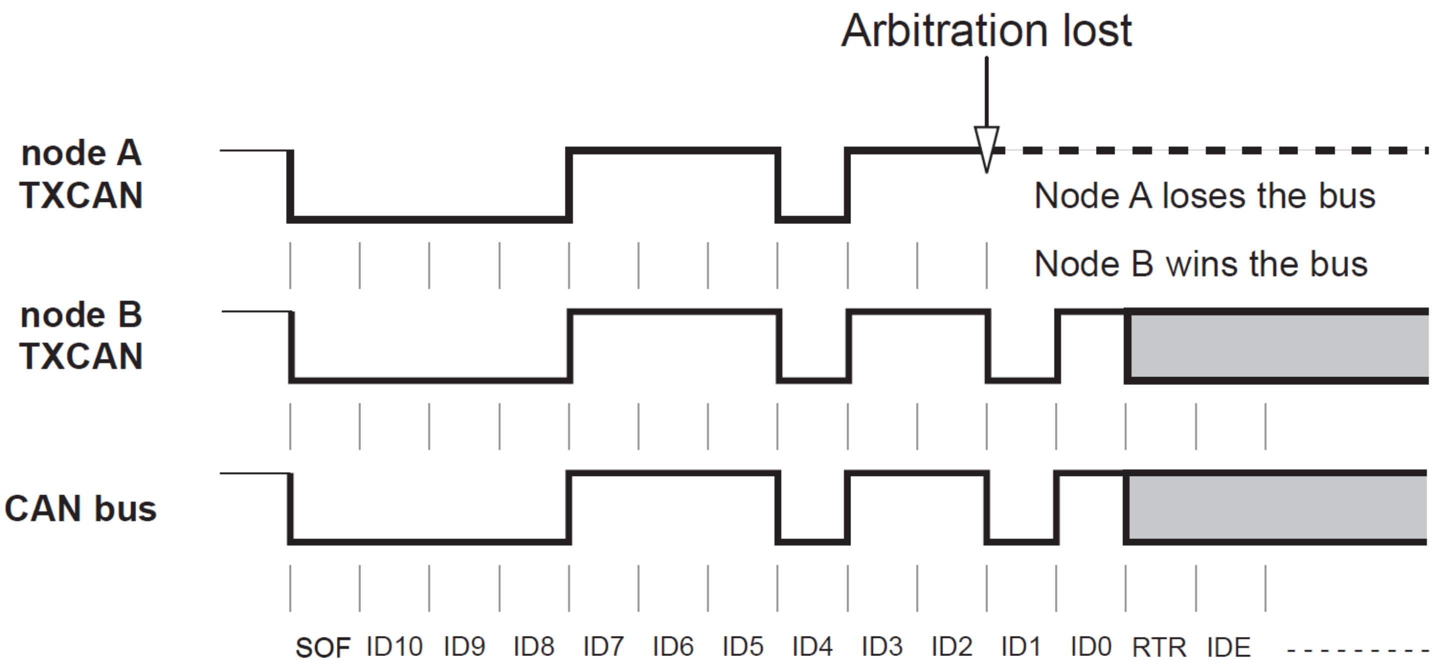
CAN Extended Frame Format



SOF – Start of Frame,
RTR – Remote Transmission Request, IDE – Identifier Extension Bit
SRR – Substitute Remote Request bit, DLC – Data Length Code

CAN Bus Arbitration

- The CAN protocol handles bus accesses according to the concept called **"Carrier Sense Multiple Access with Arbitration on Message Priority"**.
- During transmission, arbitration on the CAN bus can be lost to a competing device with a higher priority CAN Identifier, i.e. identifier with lower value.



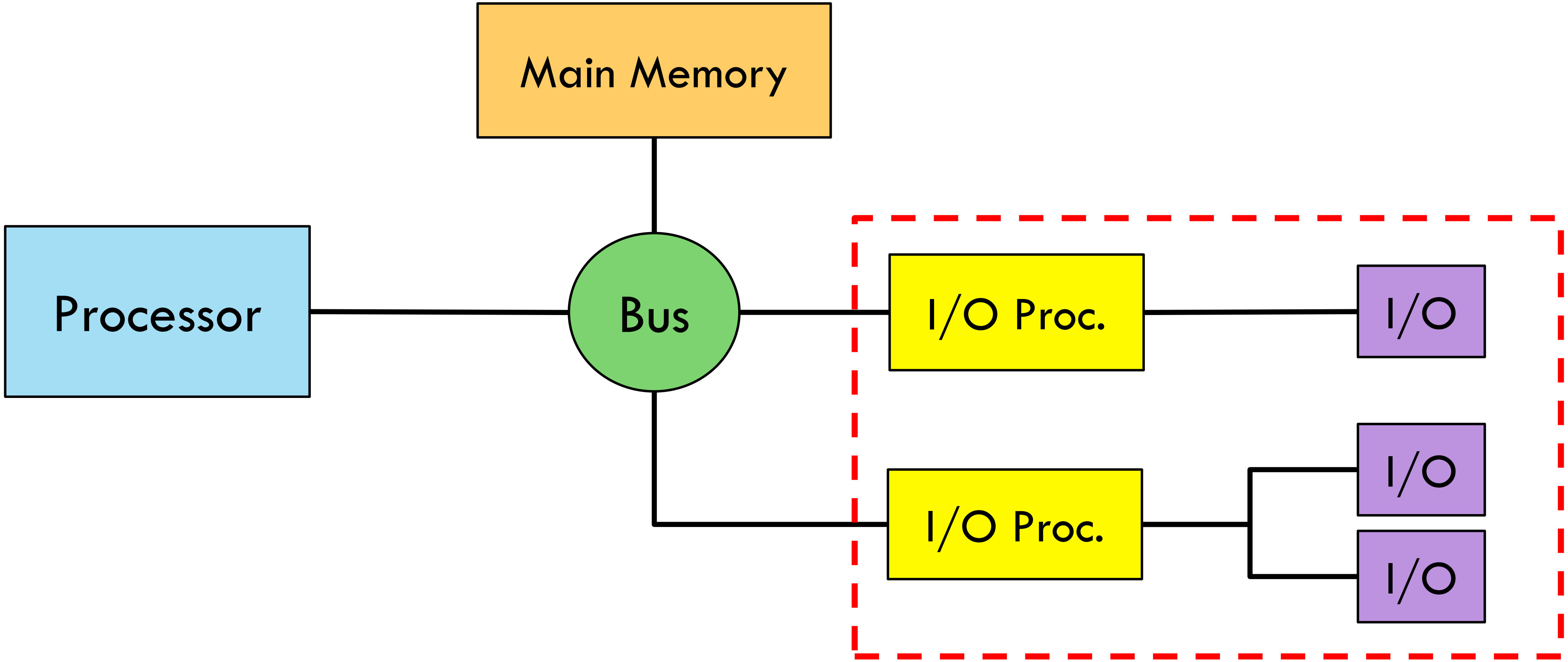
Error Detection in CAN

The CAN protocol signals any errors immediately as they occur.

Following are some error detection mechanisms implemented at the CAN message level:

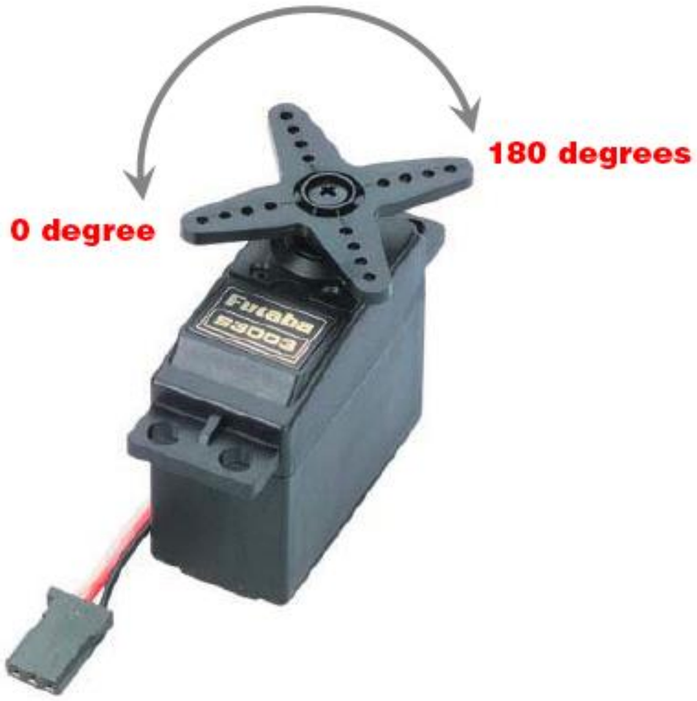
- Cyclic Redundancy Check (CRC)
- Frame Check:
  This mechanism verifies the structure of the transmitted frame by checking the bit fields against the fixed format and the frame size.
- ACK Errors:
  Frames received are acknowledged by all receivers through positive acknowledgement. If no acknowledgement is received by the transmitter of the message an ACK error is indicated.

Basic Computer Components



Servo Motor

- A Servo is a small device that has an output shaft that can be positioned to specific angular positions based on input PWM signal.
- The servo motor has a potentiometer that is connected to the output shaft and allows the control circuitry to monitor the current angle of the servo motor.
- A normal servo is used to control an angular motion of between 0 and 180 degrees.



Ref: http://lizarum.com/assignments/physical_computing/2008/servo.html

Servo Motor Applications

- Servos are typically used to control elevators, rudders and ailerons.

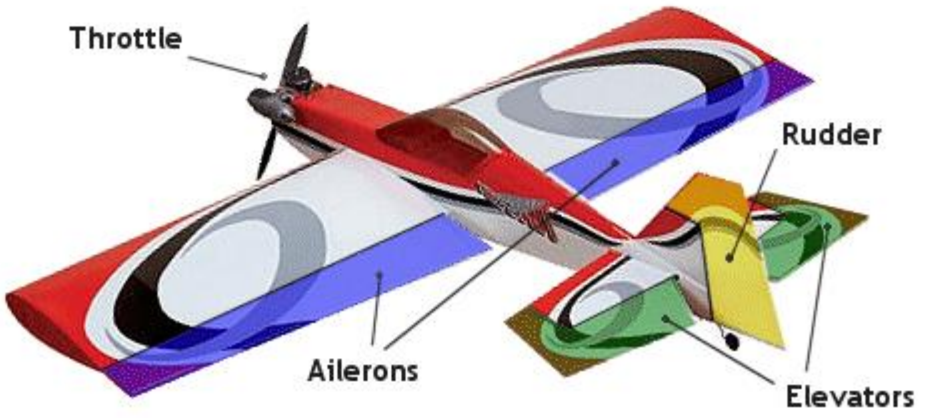
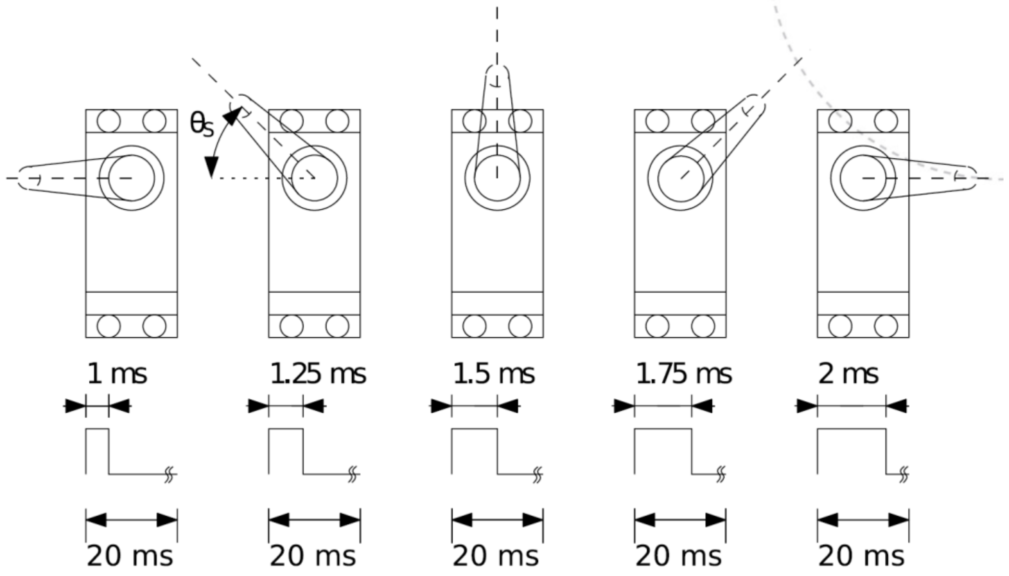


Image Refs: http://www.greatplanes.com/discontinued/gpma1414.html
http://www.rc-airplane-world.com/rc-airplane-controls.html

Controlling the servo

- The servo is controlled using a 50 Hz PWM signal (i.e. signal period = 20 ms)
- The angle of the servo is determined by the pulse width (i.e. the duty cycle)
  - 1.5ms corresponds to the center position.
- By varying the pulse width, we can control the angle
- The pulse width must never be outside the range 0.9 to 2.1ms



Connecting the servo

- Typically the servo connectors have 3 wires which should be connected as follows:
  - Red → VCC (+5V)
  - Black → GND (0V)
  - Yellow → PWM signal