Communications in embedded systems

Lecturer: Krébesz, Tamás

In general

- Point-to-point OR bus
- Wired OR wireless
- Serial OR parallel
- Synchronous OR asynchronous

- Trend is shifted towards serial communications

Main standards, protocols

- RS-232/422/485
- I2C
- CAN/DeviceNet
- Ethernet 10Mb/100Mb/1Gb
- Firewire: between PC and peripherals at 400Mbps
- J1850 SAE (OBD II)
  - Society of automotive engineers

- Profibus (Process field bus)
  - Siemens process controll
  - Smart distributed systems (SDS)
    - PLC-PC industrial comm. – Honeywell
- USB
- Felxray
- SPI
- Zigbee
- Bluetooth
- Wifi
- IrDA

RS-232/422/485

- Message structure

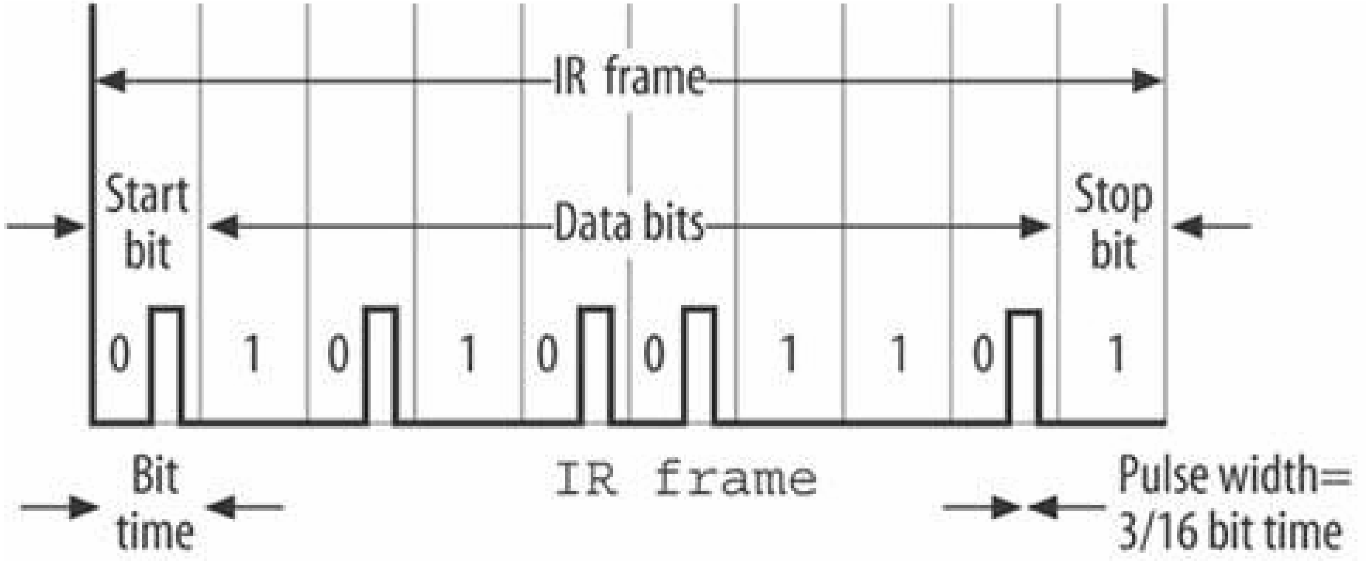
  - High = „1” = „Mark”
  - Low = „0” = „Space”
  - Start bit: 1-to-0 level change
  - Data width (word length): 5-8 bits
  - Parity: used or not before Stop bit

    - Parity offers small amount of error checking by counting „1"s in the data field (P=1->even parity)
  - Stop bit: one bit with 1 or 1.5 or 2 bit length
    - Better to say „stop period"
- Signal levels
  - RS-232: single ended
    - „0": 3…25V
    - '1': -3…-25V
  - RS-422/485: differential output

- Signal converters:
  - MAX232 (TTL->RS-232 level)
    - Charge pump is used to generate bipolar 10V from +5V of power supply
- Speed: 9.6kbps…115.2kbps

Infra red data association (IrDA)

- Physical layer different from RS-232
  - Optical data transfer in the infrared domain
- Energy saving operation:



- Half-duplex operation
  - Can receive and transmit but not simultaneously
- Error correction: 16-bit CRC
- Link distance: n*1m

- IrDA 1.0
  - 2400 baud…115.2 kbaud (default: 9600 baud)
  - RZI modulation (return-to-zero inverted)

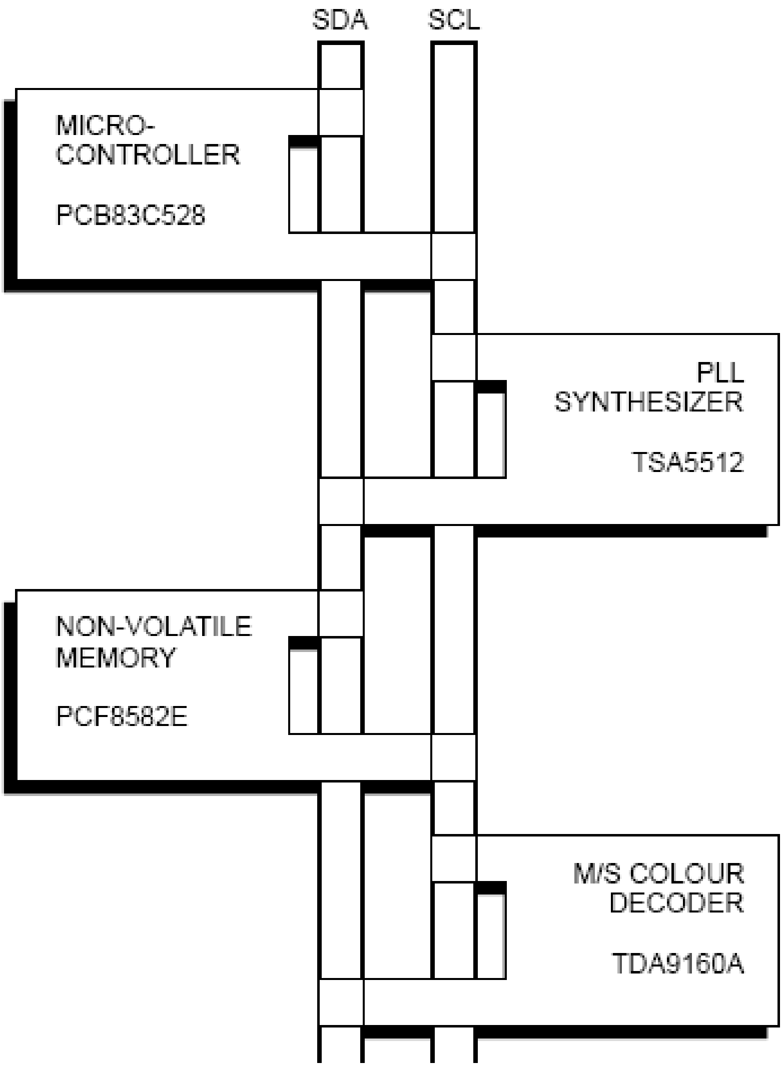
- IrDA 1.1
  - 4 Mbaud
  - Modulation
    - RZI at low data rate
    - 4PPM at high data rate

Inter IC (I2C)

- Used in
  - uC
  - Memory
  - Display
  - Keyboard
  - AD-DA converters
- Each device has its own address
- Master-slave-type comm.

- More than one master is possible
  - Collision detection and arbitration is needed
- Serial 8-bit packages
- Capacity of the bus be below 400pF
  - Number of devices in the system is limited
- Speed
  - Standard: max. 100 kbps
  - Fast: max. 400 kbps
  - High speed: max. 3400 kbps

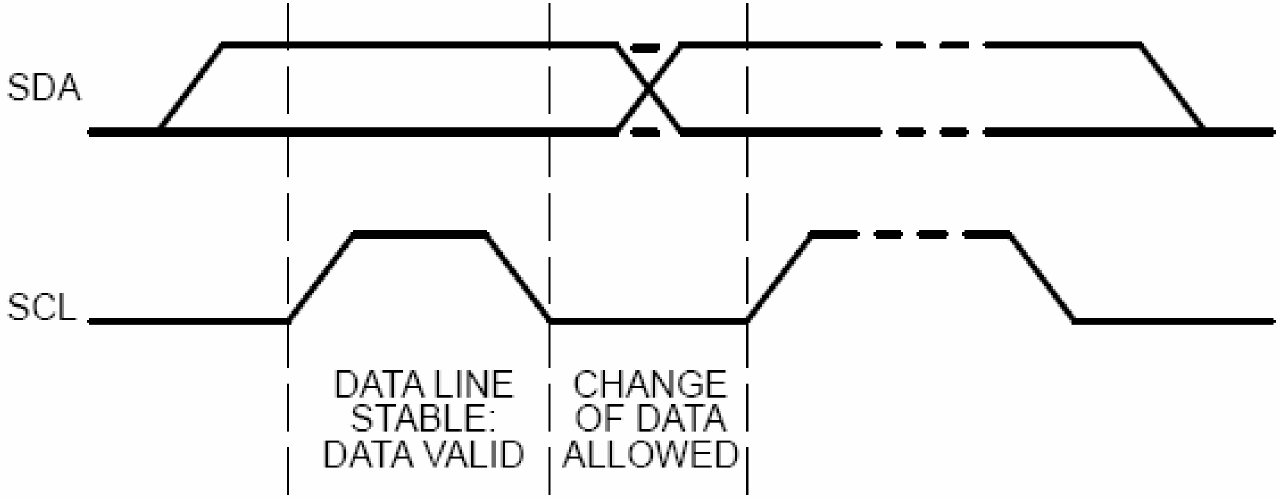
- 2 wires in the system
  - SDA: serial data line
  - SCL: serial clock line



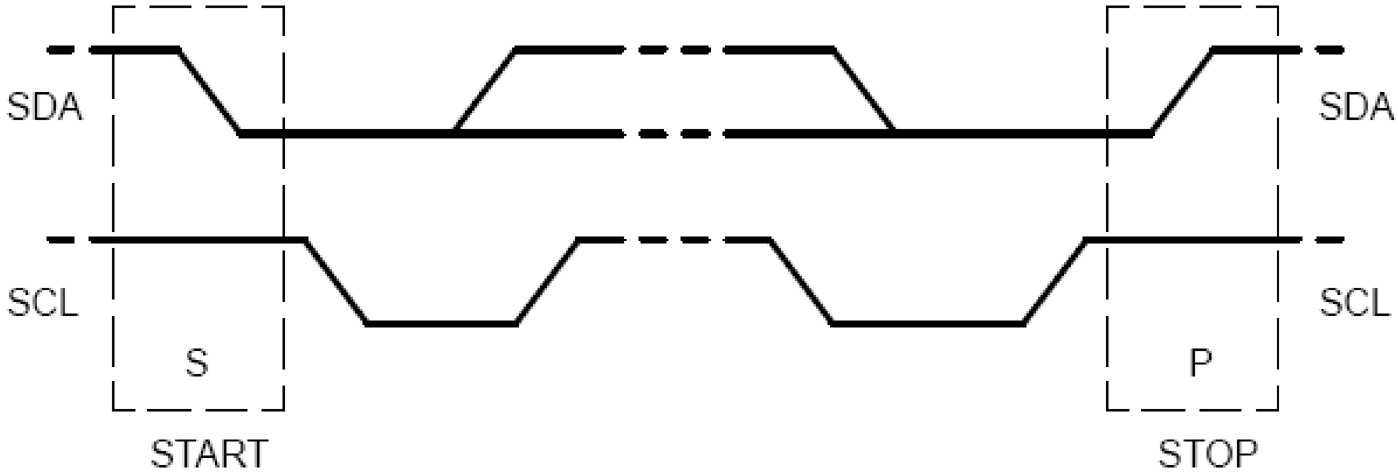
- Temrinology
  - Transmitter: send data
  - Receiver: receive data
  - Master: start/stop data transfer, generate CLK
  - Slave: addressed unit
  - Multi master: more than one master candidate
  - Arbitration: competition among master candidates for using the bus (to send data)
  - Synchronization: CLKs of multiple devices to be synchronized

- CLK is generated by the Master
- CLK can change if Slave requests for „Wait” or in case of arbitration

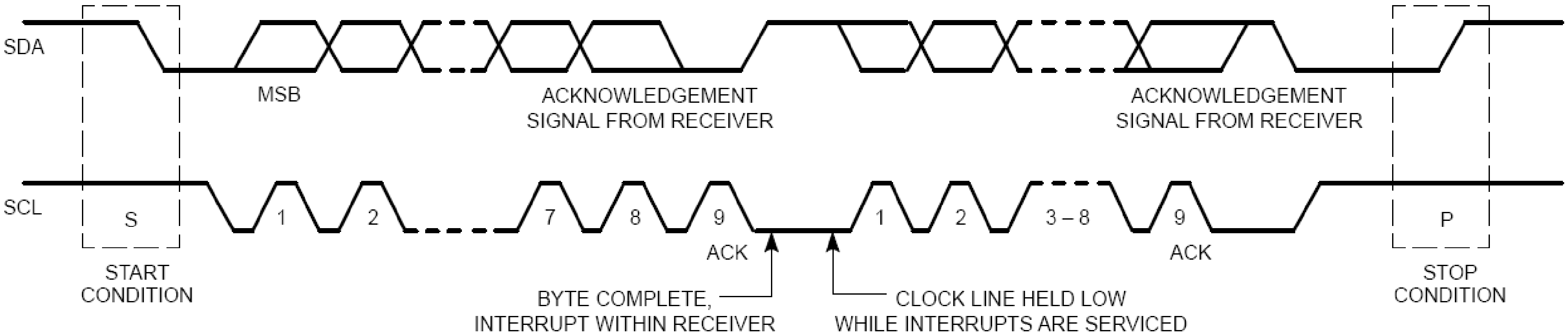
- Open collector output
- Data transfer at bit level
  - Data is valid if SCL High except for Start and Stop



– Start/Stop conditions

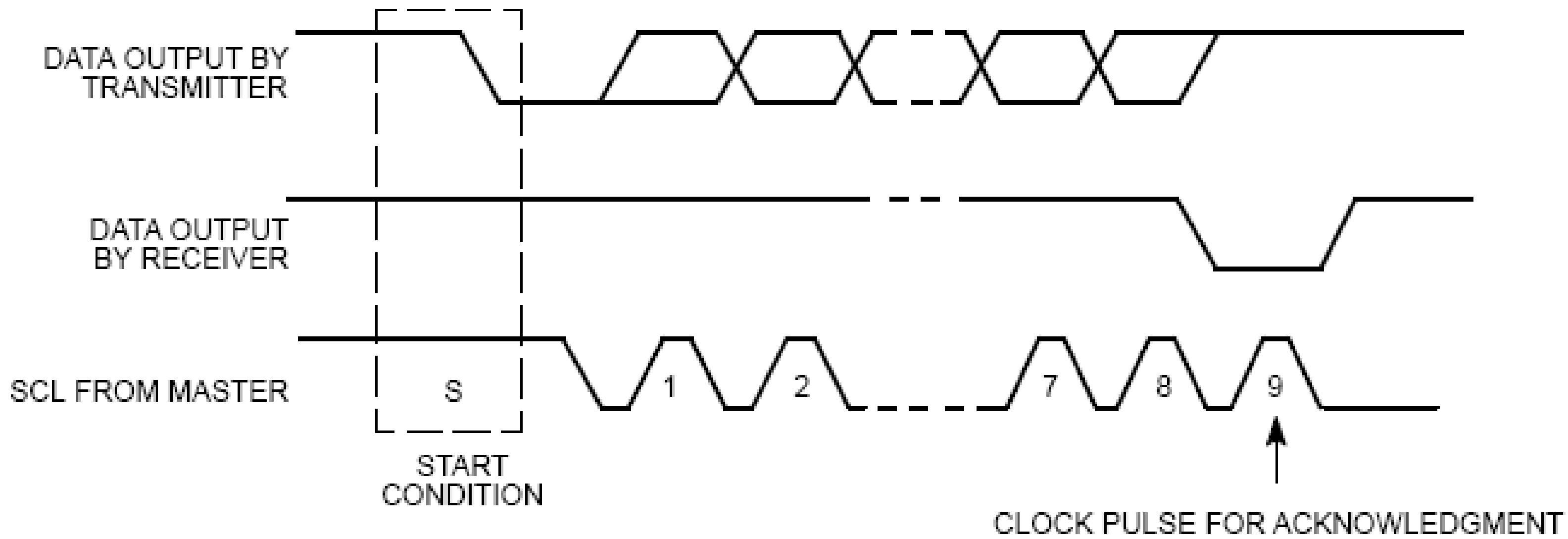


- Data transfer at byte level

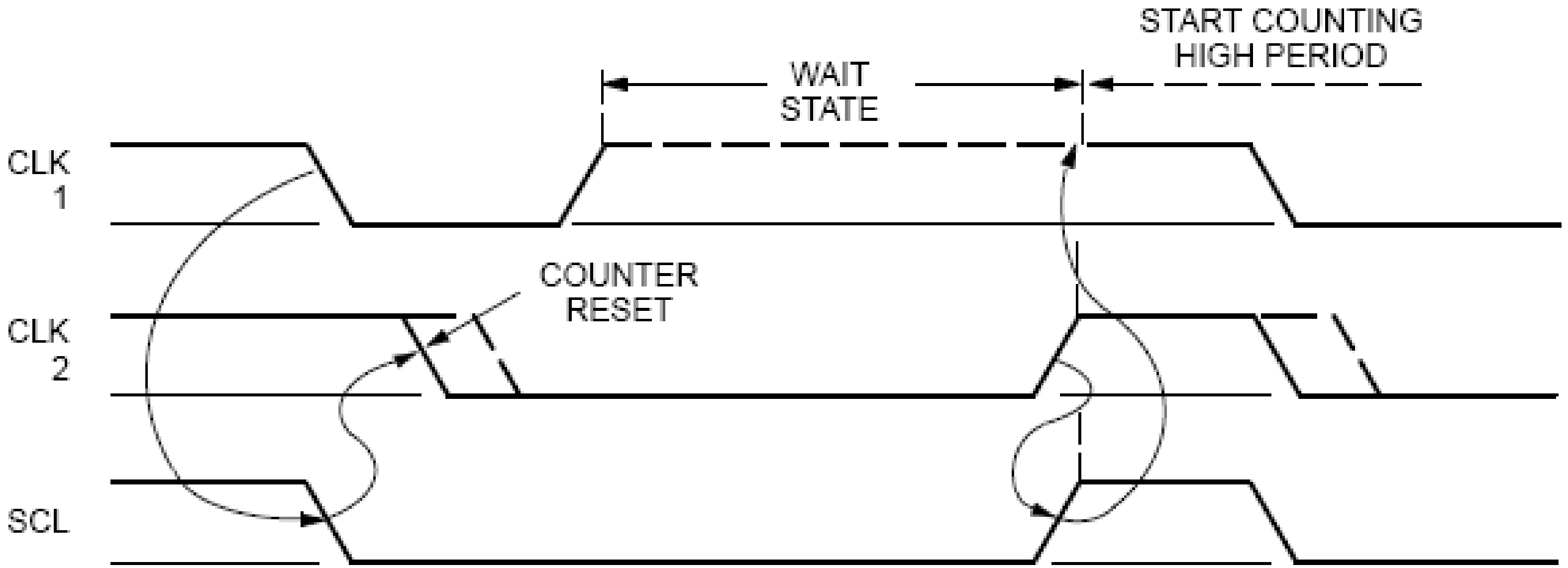


– ACK-acknowledgement
  - CLK generated by master
  - Transmitter -> ACK High
  - Receiver -> ACK Low

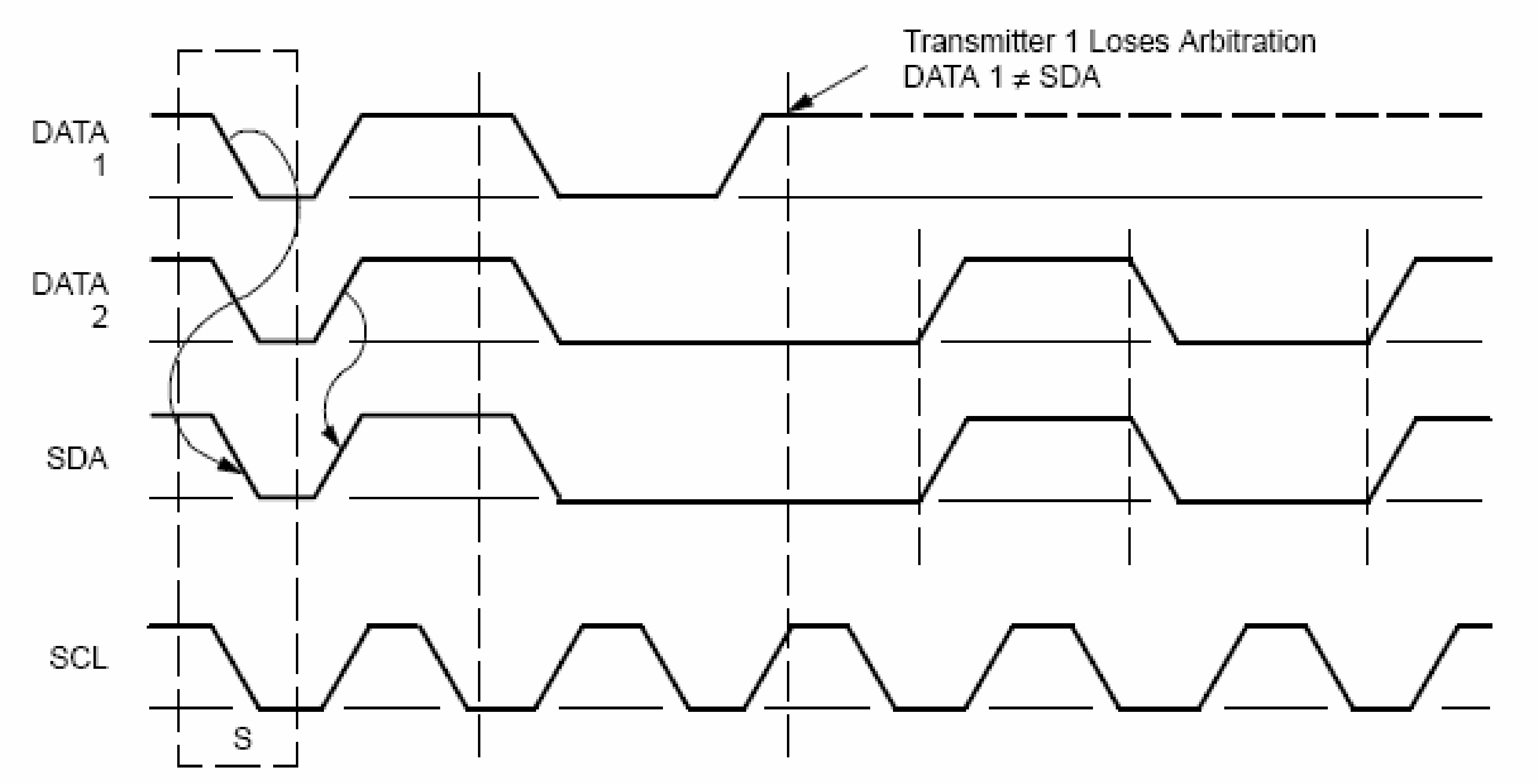
– ACK:


DATA OUTPUT BY TRANSMITTER
DATA OUTPUT BY RECEIVER
SCL FROM MASTER
S
1
2
7
8
9
START CONDITION
CLOCK PULSE FOR ACKNOWLEDGMENT

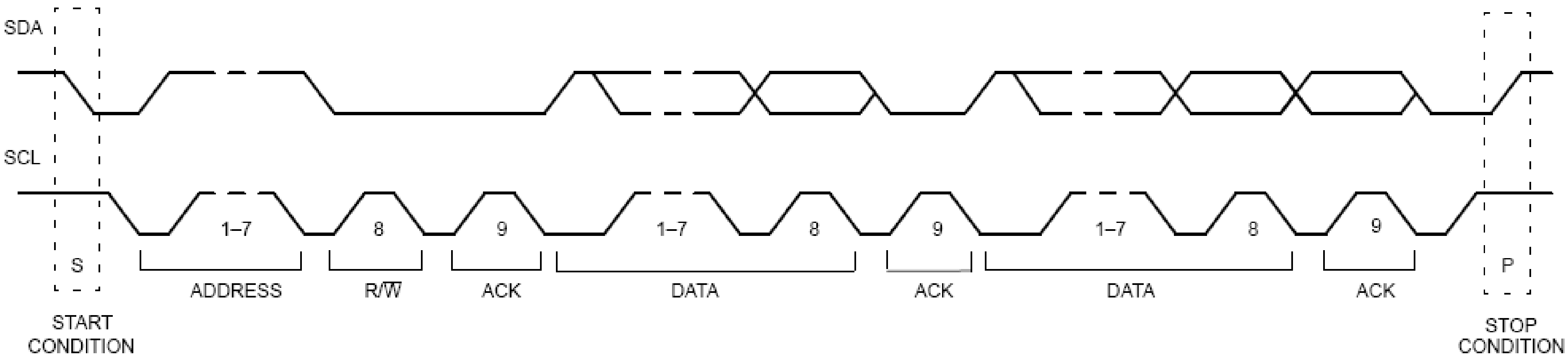
- CLK synch. during arbitration



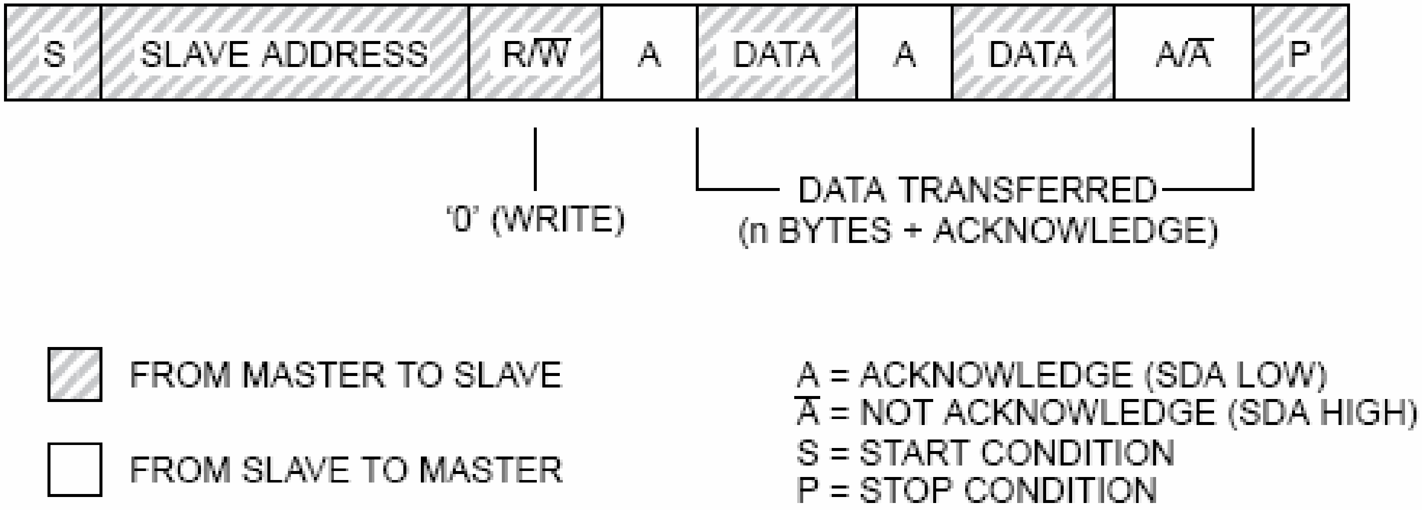
- Arbitration of Master candidates



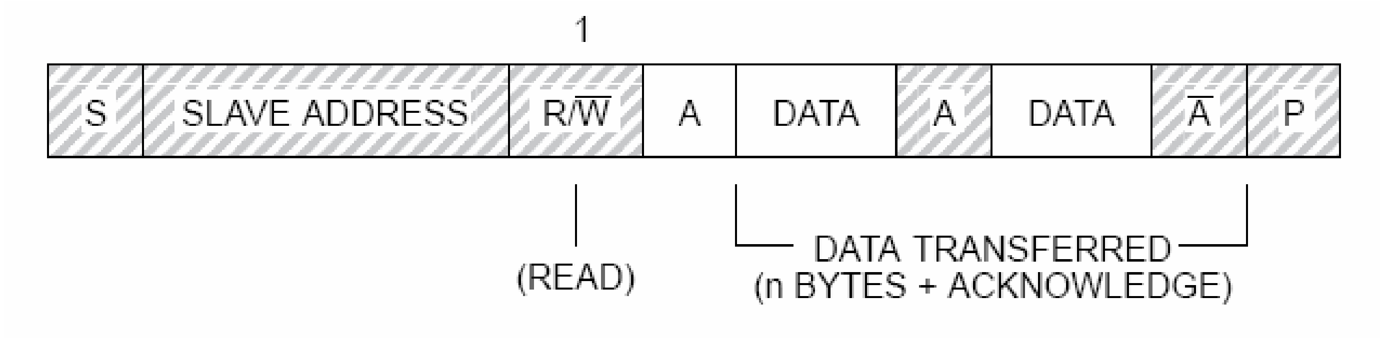
- Complete data transfer



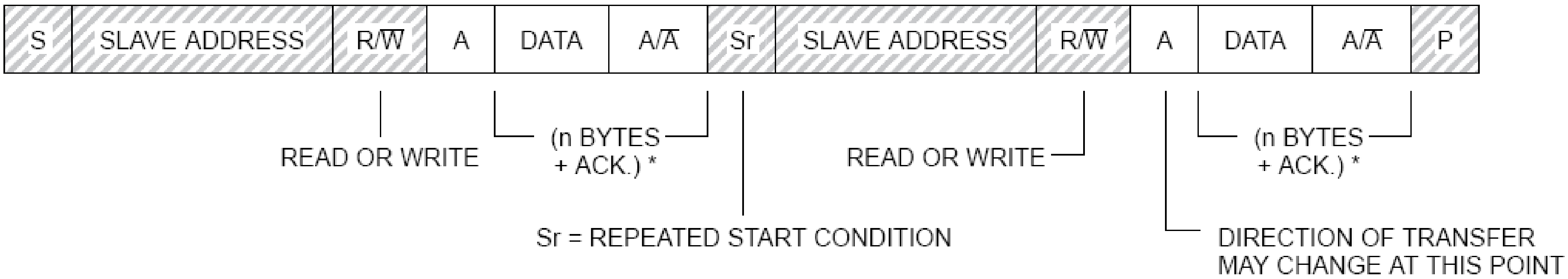
- Possible modes of data transfer
  - Transmitter is Master



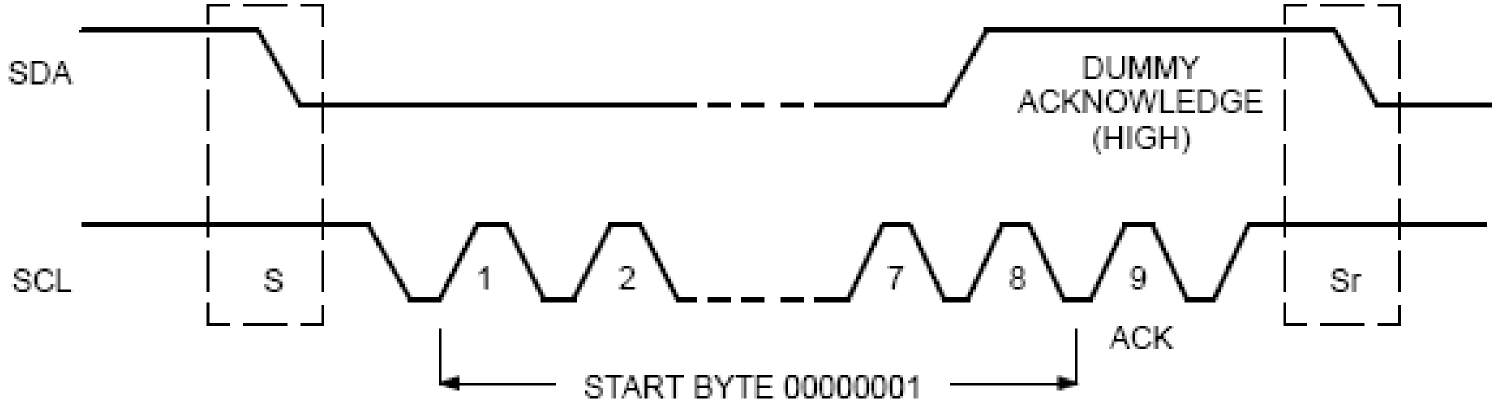
– Receiver is Master



– Combined mode



- Special addresses
  - General call address -> broadcast message
  - 10-bit addressing
  - Start byte

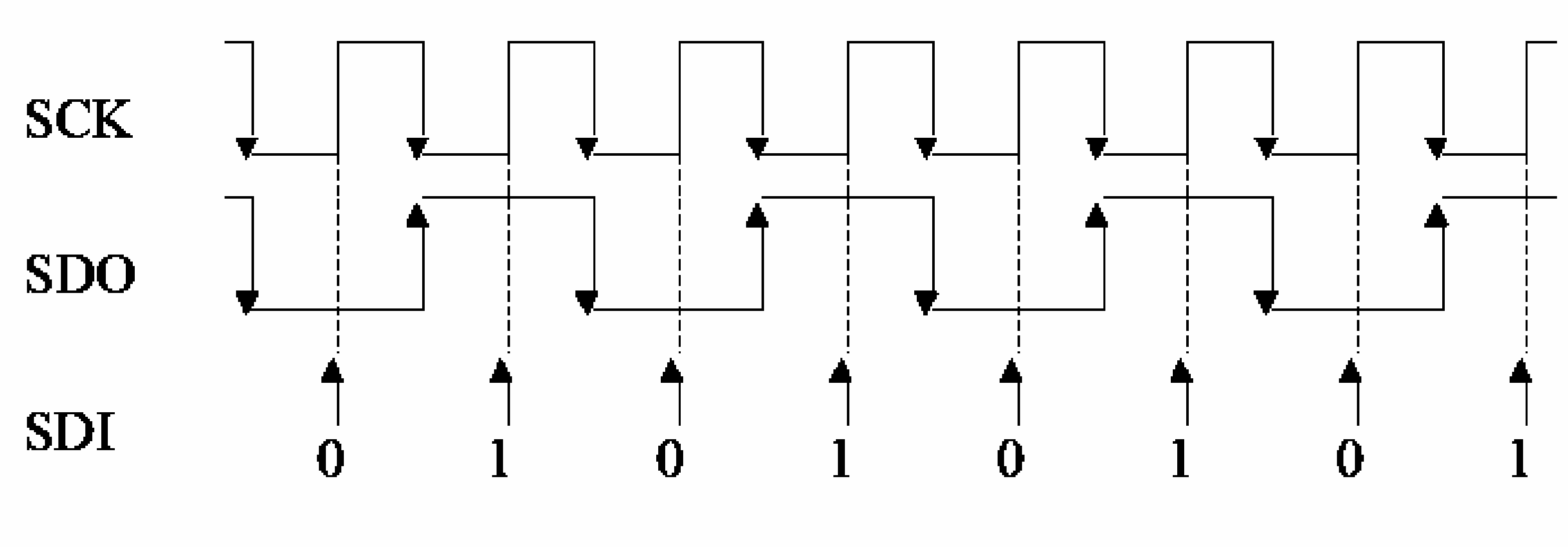


Serial peripheral interface (SPI)

- SPI mainly used when only few I/O lines are available but connection between two or more devices must be fast and easy to implement
- Sync. Master-slave comm.
  - CLK provided by Master
  - CLK rate can vary->relaxed freq. Stability
  - CLK controls when data can change or ready

- No arbitration/no unique address of a device
- SPI is data exchange protocol
  - As data is clocked out new data is clocked in
  - No „only” transmitter or receiver device
  - Must read and write or data gets lost

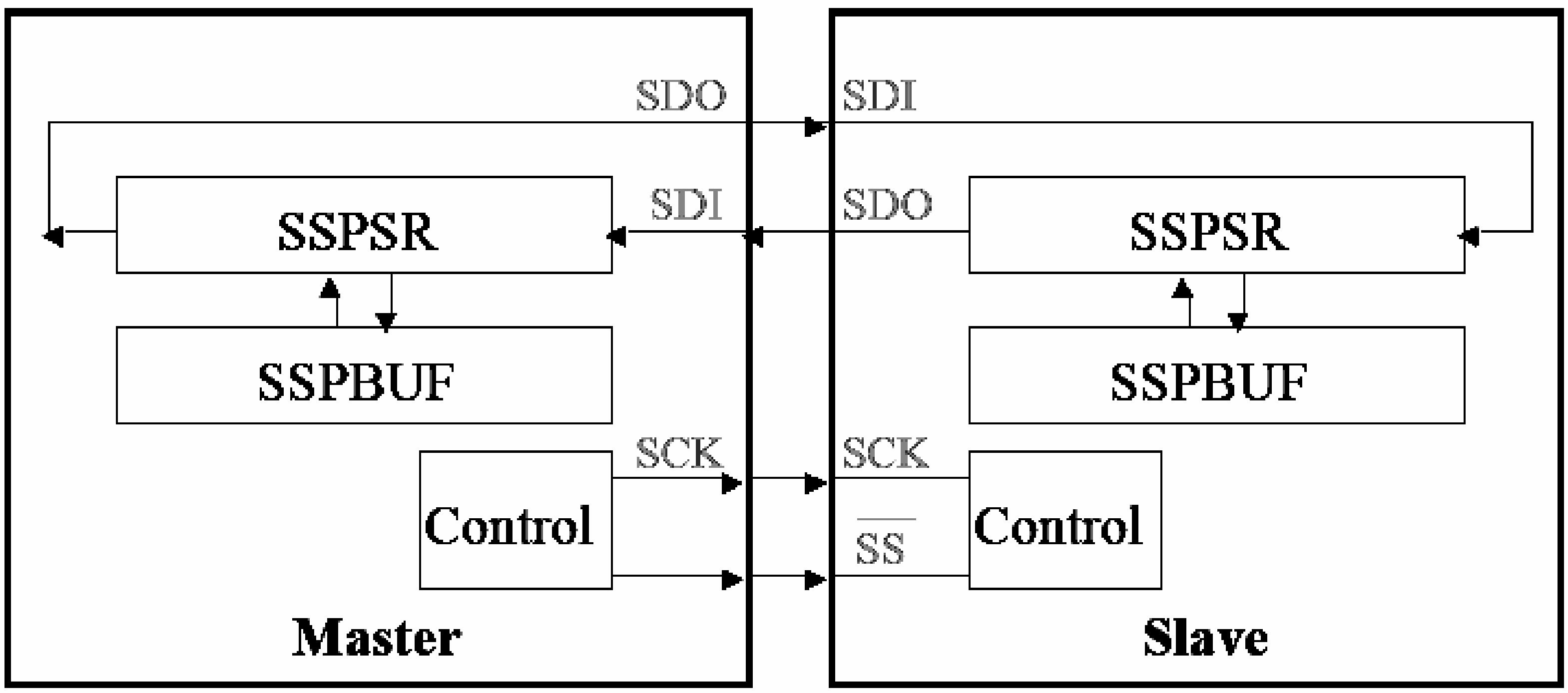
- SPI signals



  - Data changes on falling CLK (on rising also possible)
  - Reading only on rising CLK (on falling also possible)

- Lines in SPI device
  - SS not: chip select/slave select
    - SS not=High: slave listen to CLK and data
  - SCK: serial CLK
  - SDO: serial data output
  - SDI: serial data input

- SPI data transfer



  - SPI creates a data loop
  - SSPSR: shift register for the SPI module

- SSPBUF: data exchange buffer
  - Receive: data stored here and read by user SW
  - Transmit: writing into SSPBUF
    - Uploaded to SSPSR to be transmitted

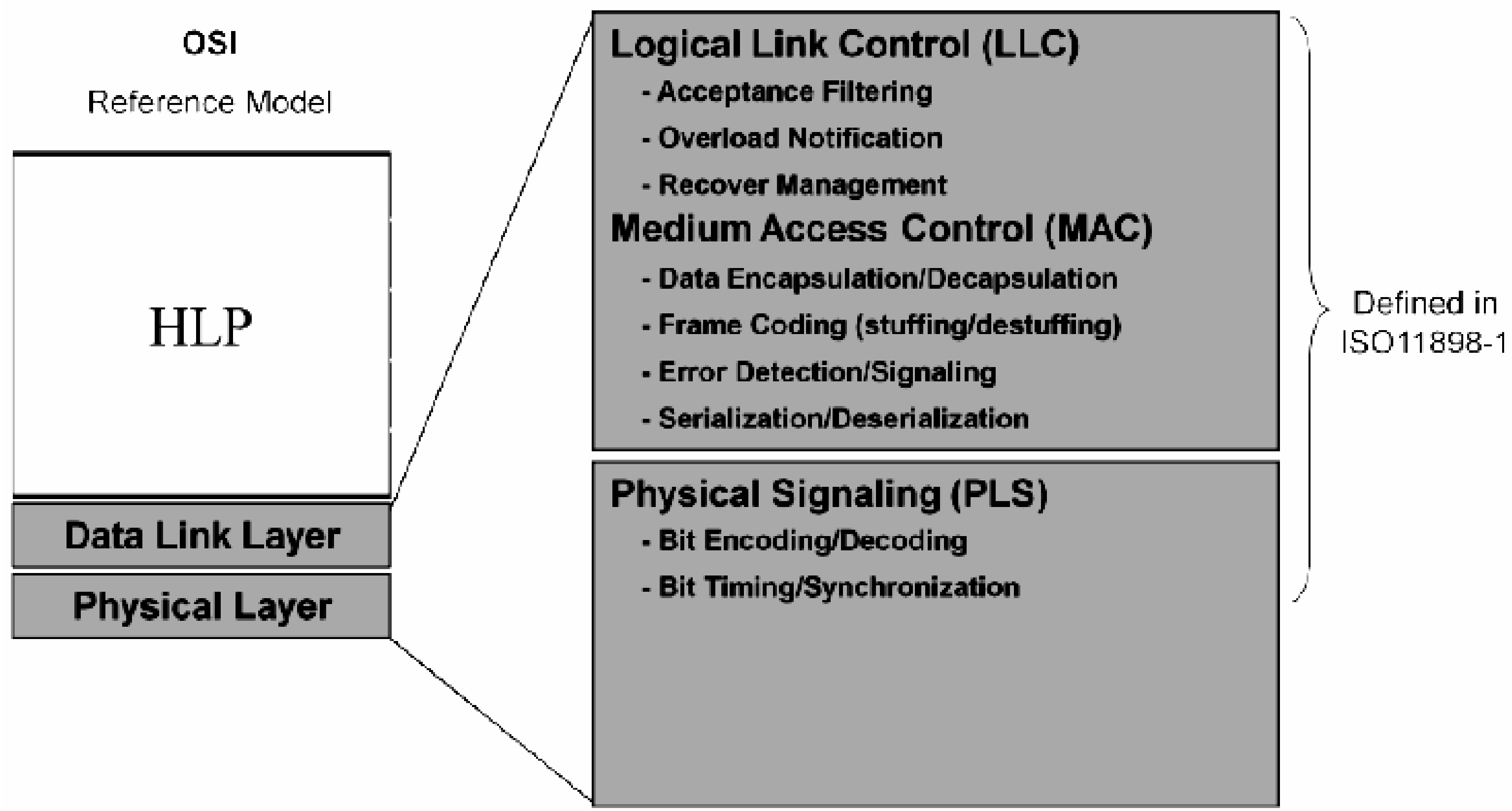
Controller area network (CAN)

- Originally developed by Bosch in 1985 to reduce cost and wiring in cars
- Defined in ISO11898 specification
- Used in distributed real-time systems
  - Vehicle industry
  - Industrial controll
  - Medical instruments

- Extremely robust serial communication protocol
  - Any device on CAN bus can detect errors
    - Message is ignored and retransmission is needed
  - Messages have to be ACK-ed by every node before processed by the addressed ones
    - 15-bit CRC used
    - If only one node reports error retransmission is needed
  - Three different error states exist
    - Different level of bus access to prevent faulty devices to take down the bus

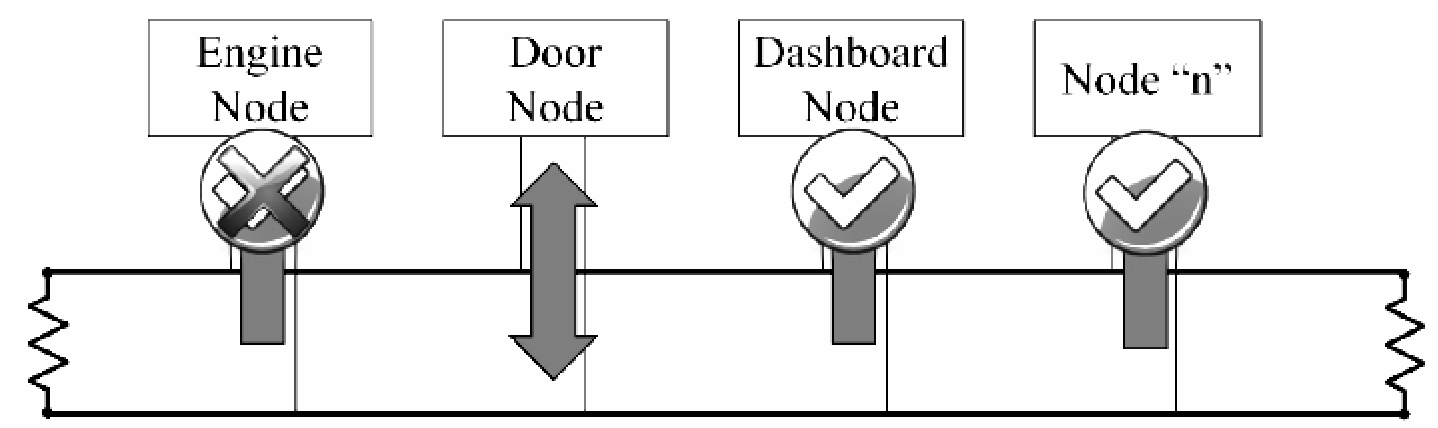
- Message-based not addressed-based
  - Messages are broadcast, devices make decision to react or not
  - New devices can be added to the system without readdressing
- High level of flexibility
  - Mester-slave OR peer-to-peer style of comm.

- Network model
  - CAN network can be mapped into OSI 7 layer
  - High layers not covered by original standard



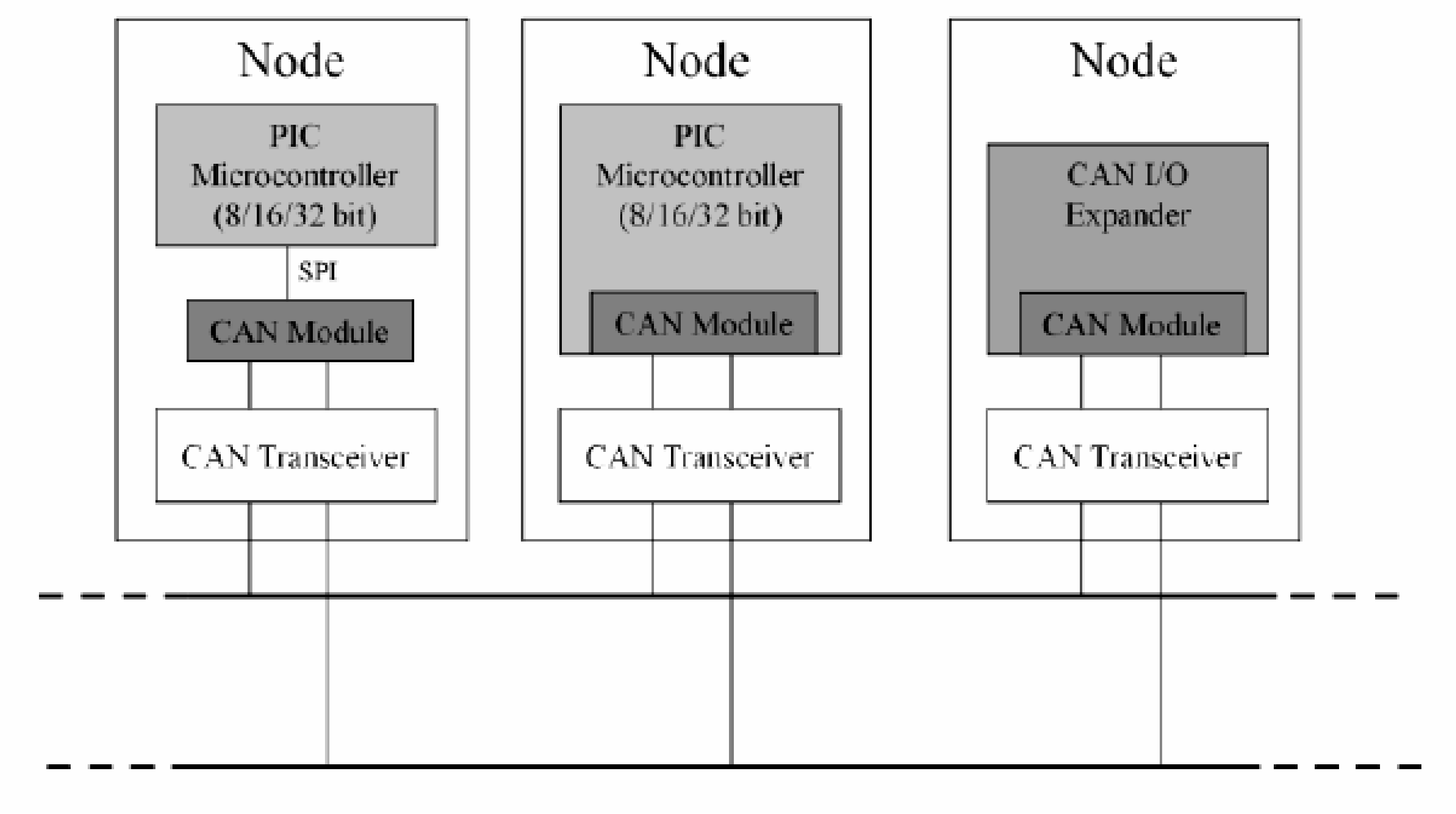
- Signal levels
  - High = Recessive
  - Low = Dominant
- Physical network
  - CAN network is made up of a group of devices called nodes
  - The smallest CAN contains at least 2 nodes
    - One transmitting
    - One receiving

– Example

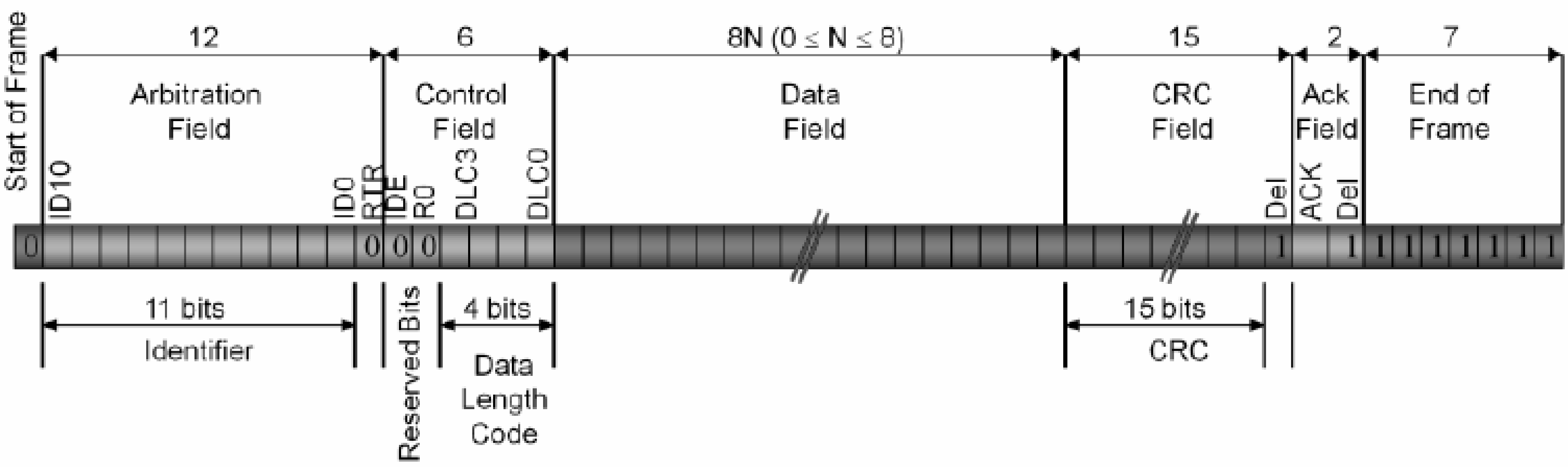


- Door node send out message of dor state
- Each node (even door node) receive and ACK it
- Nodes decide to take action or not
  - Engine node does nothing

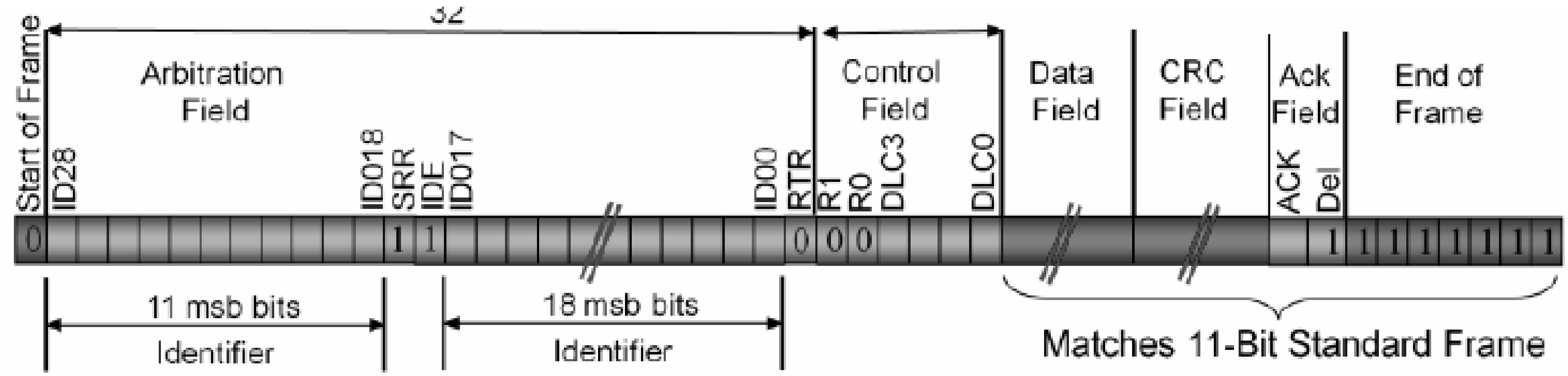
- Types of nodes



- CAN frames
  - Data frame – standard



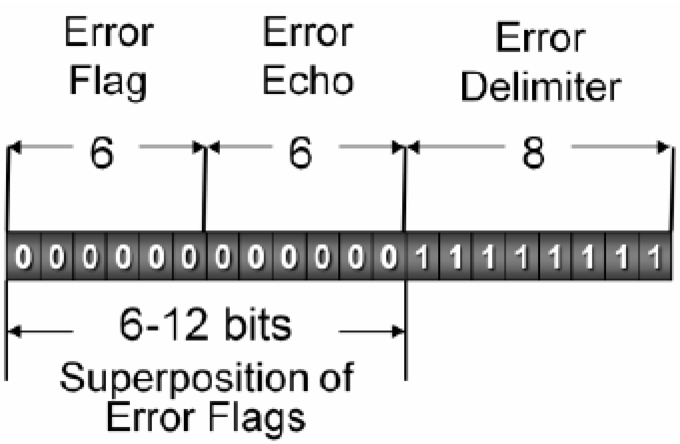
– Data frame – extended



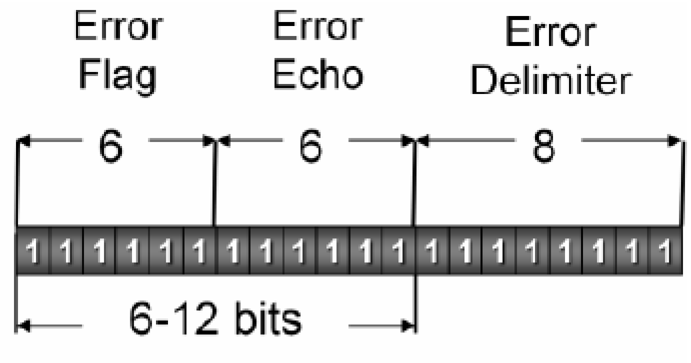
- Both standard and extended frames can coexist
- If 11-bit identifier is the same, standard has priority
- Same ID not allowed -> break arbitration process
- Lower ID number = higher priority -> ID0 always wins
- ACK: transmitter sends a recessive and listen to each receiver to transmit a dominant
- After EOF 3 bits of quiet time – inter frame space

– Remote data frame
  - Same as data frame but
    – RTR=1
    – no data
  - Used when one node needs to request data from an other
  - Identifiers are matched between transmit and receive node

– Error frame
  - Active:



- Passive:

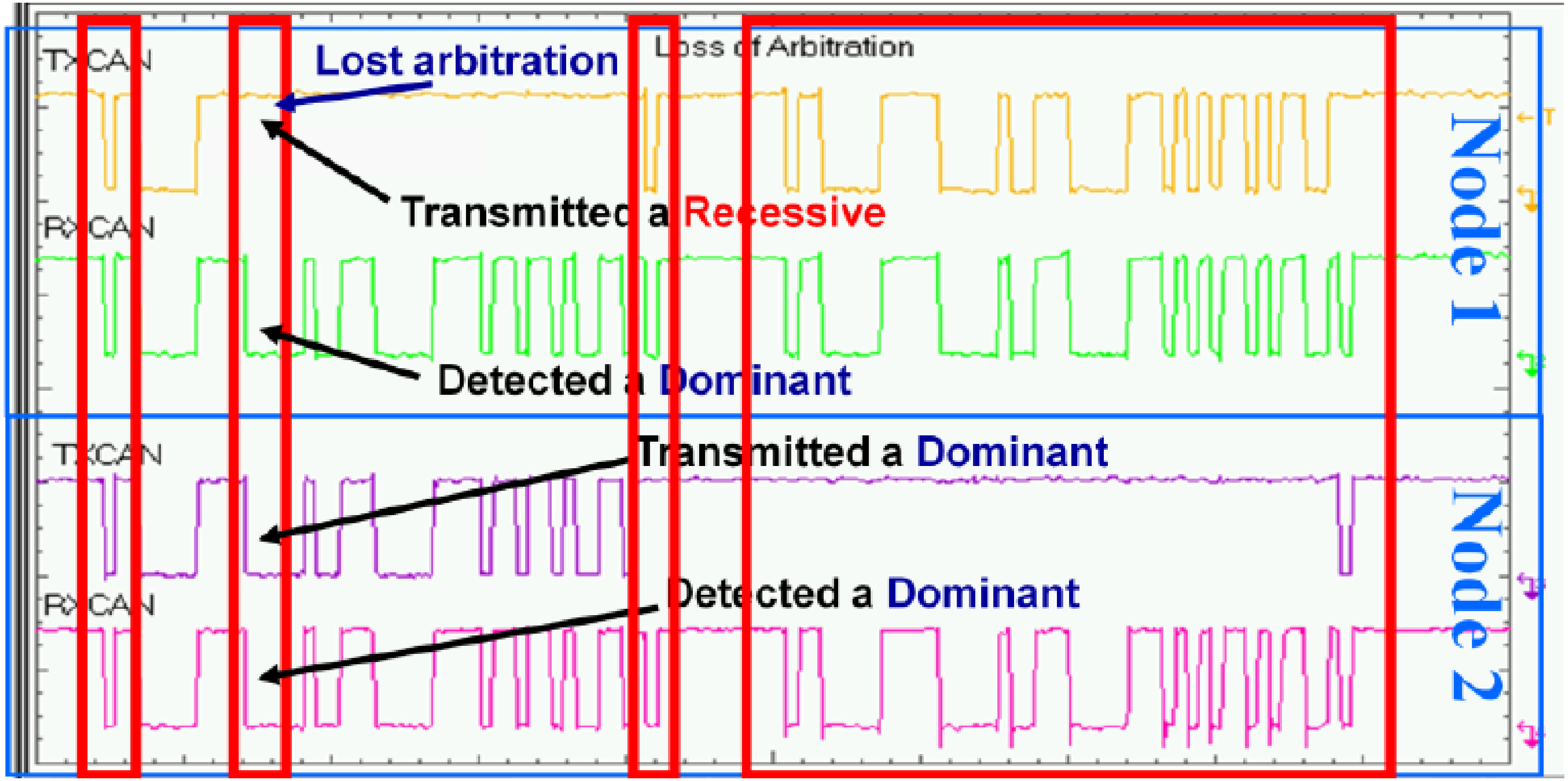


  - Error frame transmitted when any node detects CRC mismatch in Data or Remote frame
  - If a node is in error active (or passive) state -> transmit active (or passive) error frame
  - Error frame can vary in length
  - Retransmission of data is forced by Error frames

– Overload frame = max. length active error frame
  - does not cause retransmission of message
  - Only delay the message when a node needs more time for processing data

- CSMA/CD-CR
  - Carrier sense multiple access and collision detection with collision resolution
    - Carrier sense: every node listens into the bus
    - Multiple access: nodes have equal opportunity to transmit message
    - Collision detection: nodes transmit at the same time -> collision -> transmitting a recessive bit
    - Collision resolution
      - Nondestructive bitwise arbitration
      - Dominant bits win arbitration over recessive bits

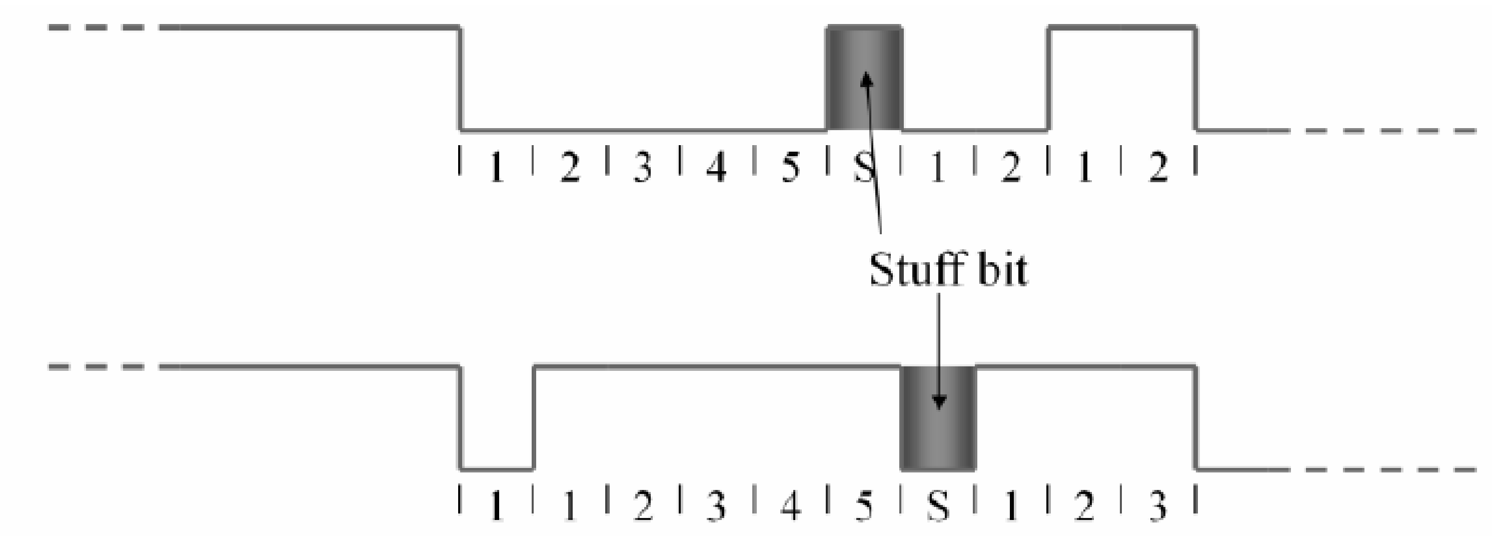
- Time diagram example of CSMA/CD-CR



- Transmit and receive lines between uC and CAN module
- Both nodes continue to transmit until there is a mismatch
  - Arbitration won by Node 2 ->continue transmission
  - Node 1 becomes receiver
- ACK: Node 1 transmit a recessive, Node 1 transmit a dominant -> no error

- Synchronization
  - No CLK lines among nodes but each node has own CLK
  - Synch. on recessive to dominant transitions
    - Hard sync. occurs at SOF and resets bit CLK
    - Re-sync. occurs at recessive-to-dominant (1-to-0) transitions -> adjust bit CLK as necessary
  - CAN implement NRZ (non return to zero) coding on the physical bus-> no edge between to like bits
    - How synchronization is maintained then?

- Solution: bit stuffing
  - Ensures edges and so sync.
  - Stuff bit occurs after 5 like bits in a row



  - Stuff bit is added at the protocol level at the transmitter and removed at the protocol level by the receiver
    - » User not aware of stuffed bits

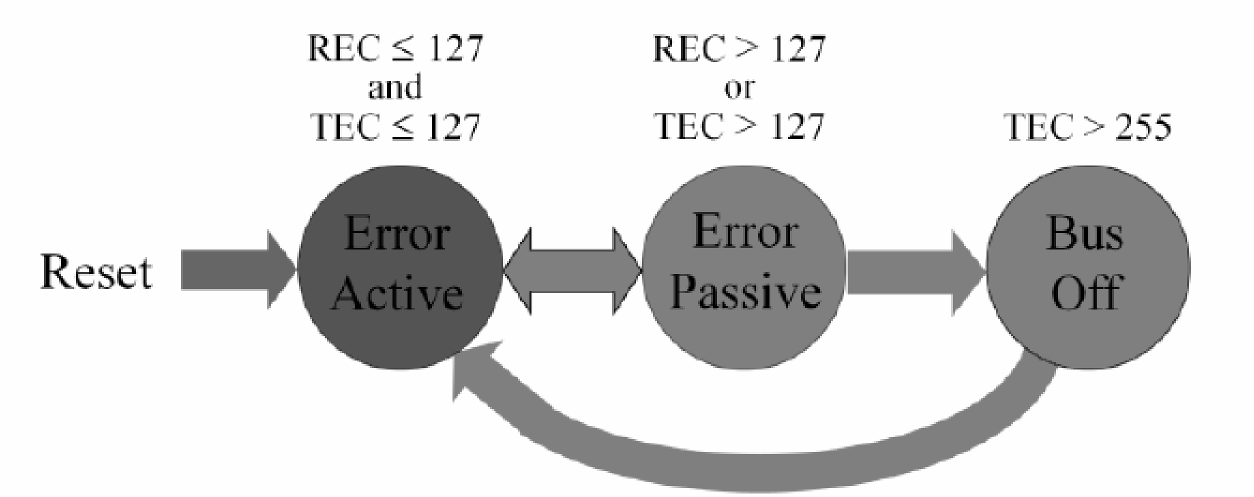
- Error handling
  - Different type of errors are defined
  - Normally working CAN nodes can become totally disconnected from the bus
    - It prevents faulty node to take down the bus
  - If error detected node transmit error frame
    - An internal receive/transmit counter incremented
    - Node can recognize if it is a bus problem and disconnects itself from the bus
    - Prevents a single node from loading the bus with error frame thus preventing valid data transmission

- 15-bit CRC error
  - CRC not matched -> error frame sent out by receiver -> original message not valid
  - Error frame receivd by transmitter -> data frame retransmission is needed
- ACK error
  - Transmit node check ACK bit
    - Recessive sent
    - Dominant is waited
  - Dominant detected: at least one node received the message correctly
  - No dominant ACK: transmitter transmit error frame to destroy original message and retransmit it

- Form error
  - Any node that detects a dominant in
    - CRC delimiter or
    - ACK delimiter or
    - EOF or
    - Interframe space -> error frame sent out -> message must be resent by transmitter
- Struff error
  - If bit stuffing violated error frame detected
  - Note: error frame intentionally violates bit stuffing

- Bit error
  - Transmitter monitors the bits sent by itself
    - In case of mismatch bit error occurred -> message resent
  - Exceptions
    - During arbitration
    - In ACK slot bit

- States of CAN nodes



  - Counters can be decremented when node detects valid message
  - Return from Bus off:
    - Config mode counters nulled
    - Long bus idle
    - 128 valid messages

References

- Microchip web seminar on CAN
- Microchip: Overview and Use of the PICmicro Serial Peripheral Interface
- Philips semiconductors: I2C bus specification-1995