Chapter 3

Communication Protocols

Introduction:
Parallel/Serial Communication

In a system with binary codification

1 Physic Line → 1 bit

If the indivisible chunk > 1 bit

MULTIPLEXING

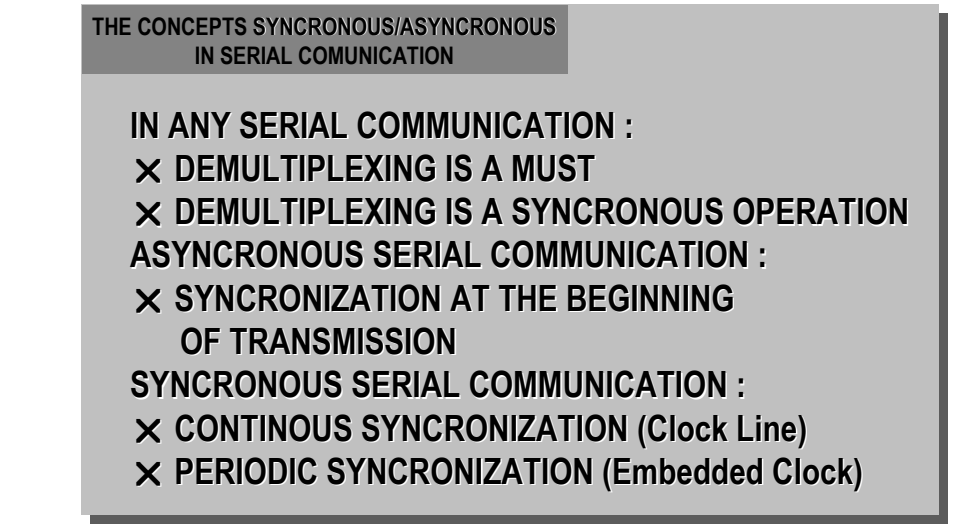
× SPACE (Paralel)

× TIME (Serial)

With a binary code, only a bit may be transmitted through a single line. For units of information of more than one bit, wether we use more lines (Space multiplexing or parallel) or we assign a time slot to every bit (Time multiplexing or serial)

EXAMPLES

- μP Bus
  - Paralel ➡ AB, DB, CB
  - 1 Transfert ≥ 1 Clock cycle
- RS232
  - Serial ➡ 1 Physic Line
  - 1 Bit = 1 Clock cycle



To recover time multiplexed data we need the clock that has been used to multiplex them, so we need to be syncronized. Strictly speaking, serial transmission is syncronous.

Disrinction between syncronous and asyncronous serial transmission is based on how we do recover syncronization.

In syncronous serial transmission without dedicated clock line, clock is recovered from data.

CONSEQUENCES OF SYNCRONIZATION METHOD

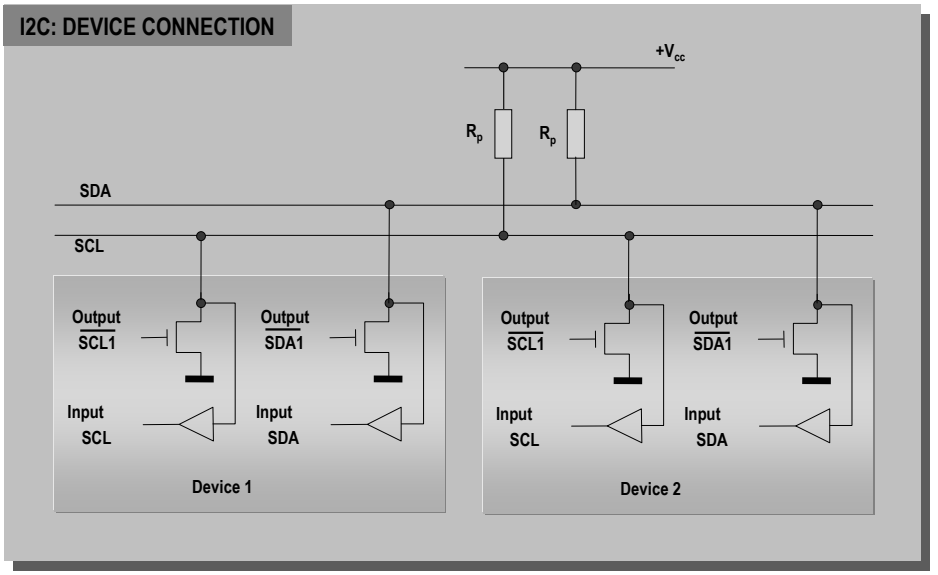
ASYNCRONOUS SERIAL COMMUNICATION
- ✕ CLOCK SHIFT ACUMMULATION
- ✕ LIMITED MESSAGE LENGTH
- ✕ LIMITED TRANSMISSION SPEED

SYNCRONOUS SERIAL COMMUNICATION

✕ CONSTANT SYNCRONIZATION
- ✕ STRICTLY SYNCRONOUS
- ✕ EXTRA CLOCK LINE

✕ PERIODIC SYNCRONIZATION
- ✕ SYNCRONIZATION RATE DEPENDS ON SPEED
- ✕ SYNCRONIZATION DEPENDS ON MESSAGE

I2C Bus

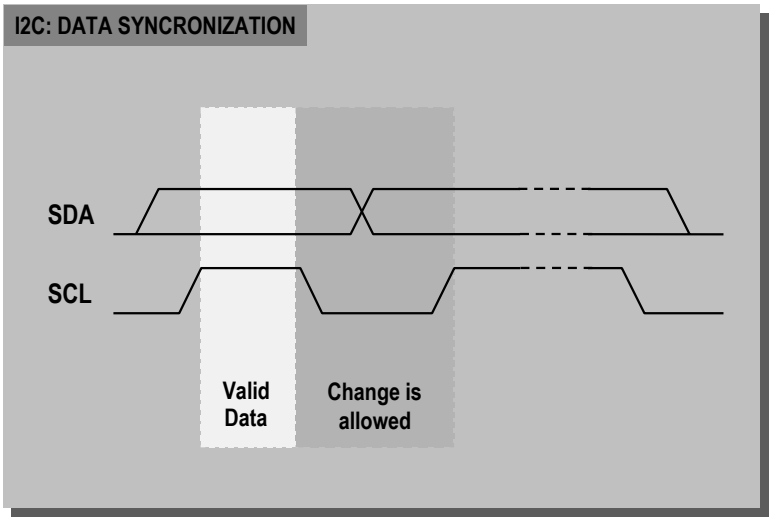


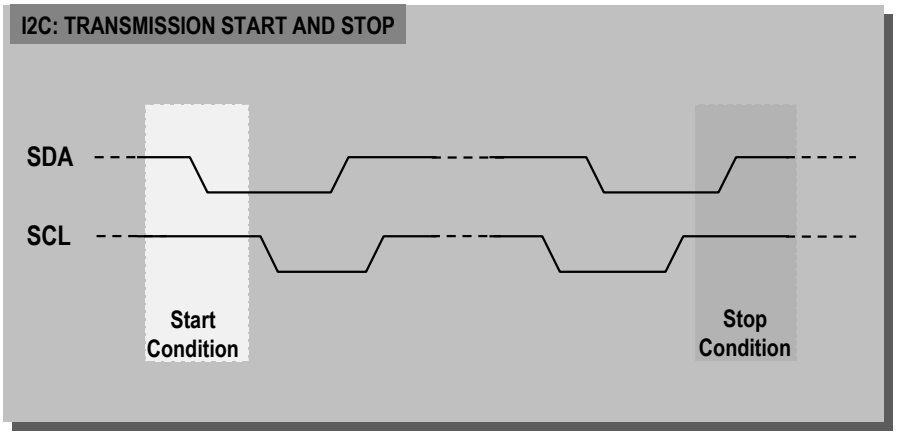
Separate CLOCK (SCL) and DATA (SDA) lines

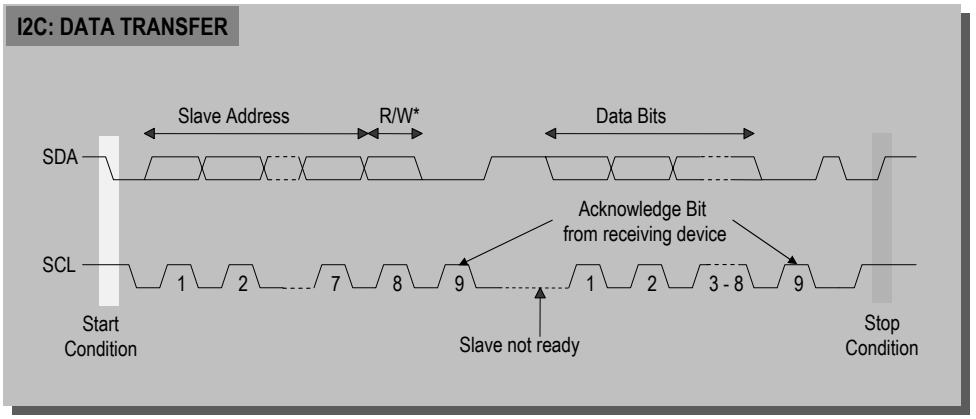
Uses DOMINANT STATE SIGNALING (Output devices are open collector or open drain)

DOMINANT state is line at low level, when ANY device connected to the line is transmitting a low level.

RECESSIVE state is line at high level, when ALL devices connected to the line are transmitting a high level.

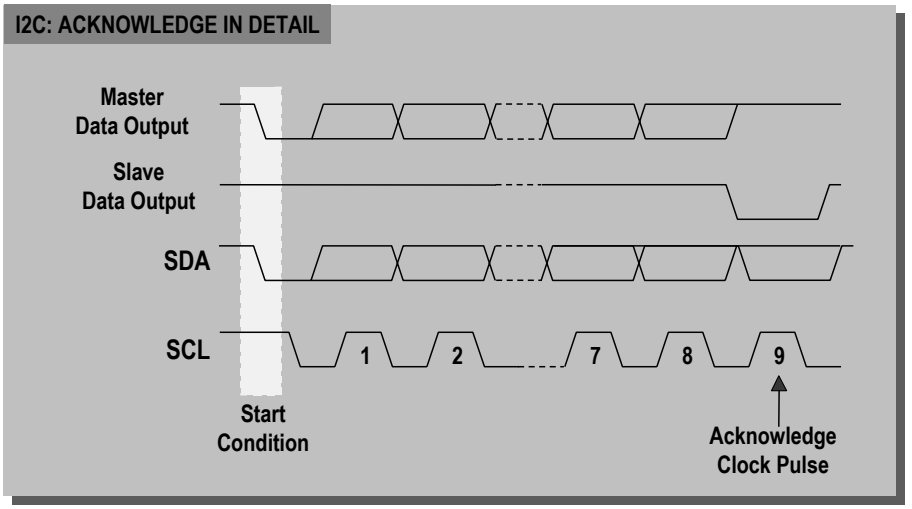

I2C: DATA SYNCRONIZATION
SDA
SCL
Valid Data
Change is allowed


I2C: TRANSMISSION START AND STOP
SDA
SCL
Start Condition
Stop Condition



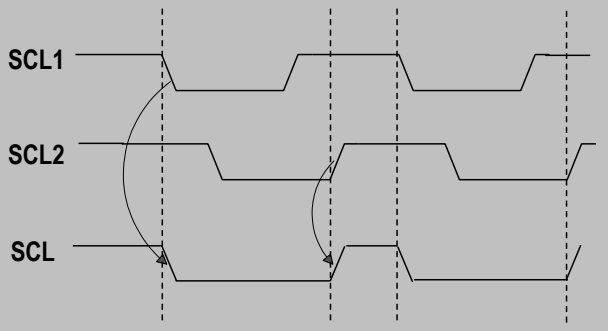
Basic transfer is:

1) Start condition
2) 7 address bits
3) 1 R/W* bit
4) 1 address acknowledge bit (Slave forcing a dominant level)
5) Undefined time (optional) of "Slave not ready", where slave forces a dominant level in SCL line, thus stopping the transmission until ready.
6) 8 data bits or 8 bit groups in case of multiple data transmission.
7) 1 data acknowledge bit (for every data byte)
8) Stop condition

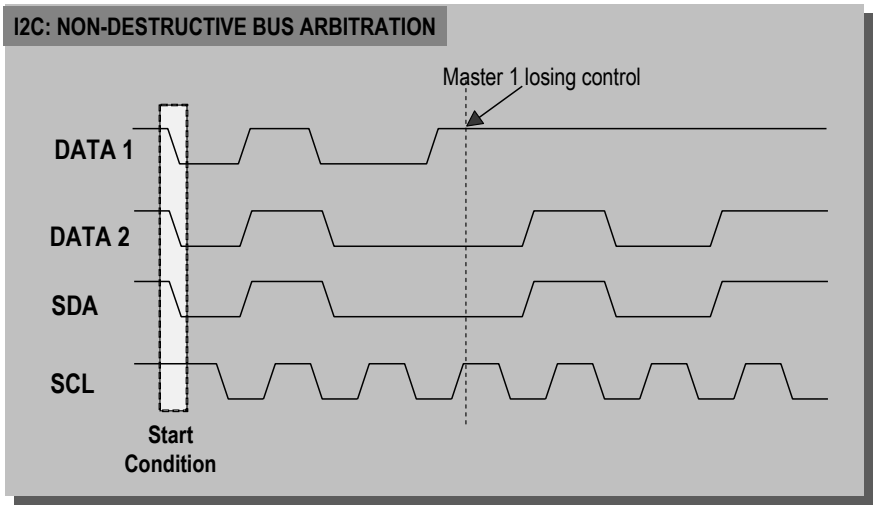


During acknowledge bit time, *master* leaves SDA line floating (Recessive) and *slave* forces line dominant.

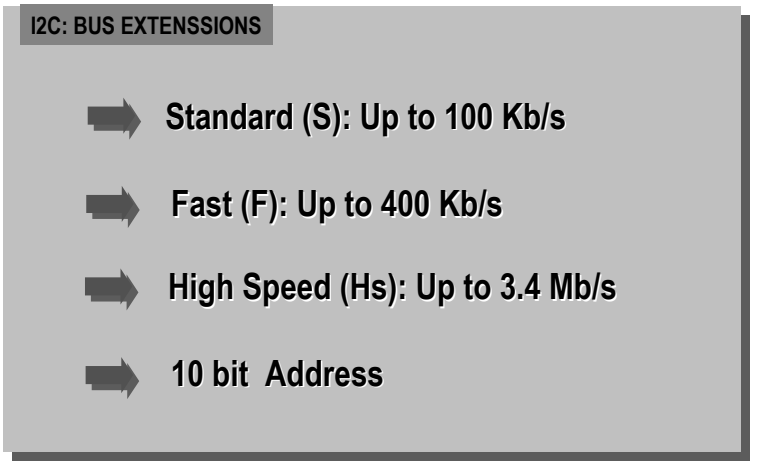
I2C: SCL SYNCRONIZATION



SCL = SCL1 *and* SCL2



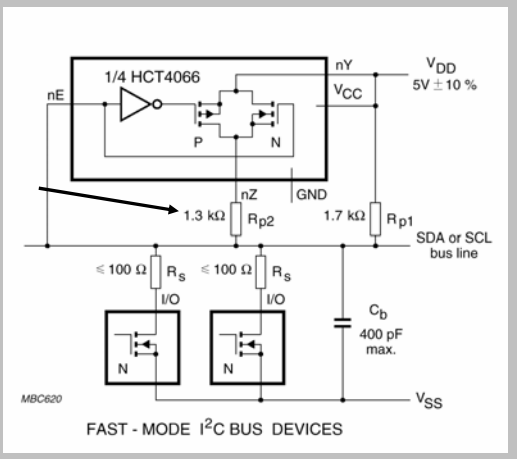
Address containing more dominant bits gets the bus



See I2C specs (Annexed documents)

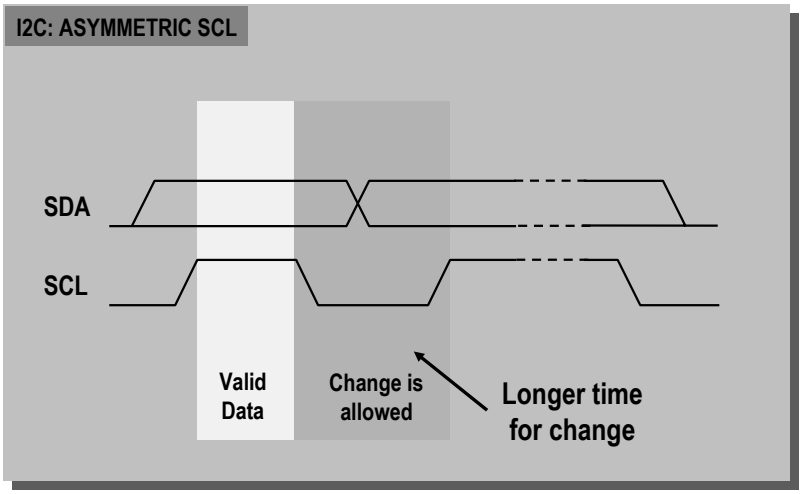
I2C: FAST MODE: P.U. RESISTANCE

ON=0.8V
OFF= 2V



Needed from C_L = 200pF

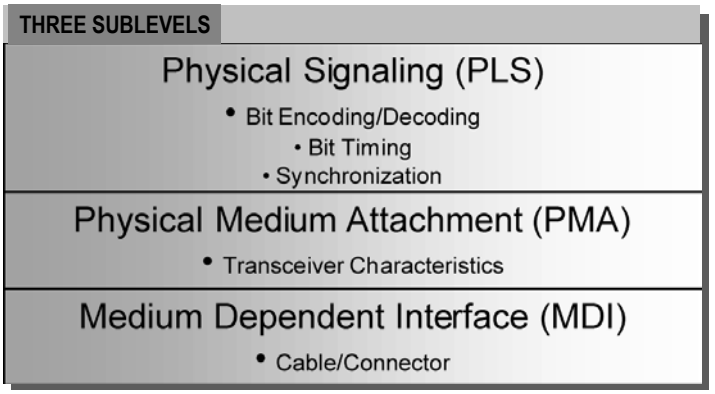
See I2C specs (Annexed documents)



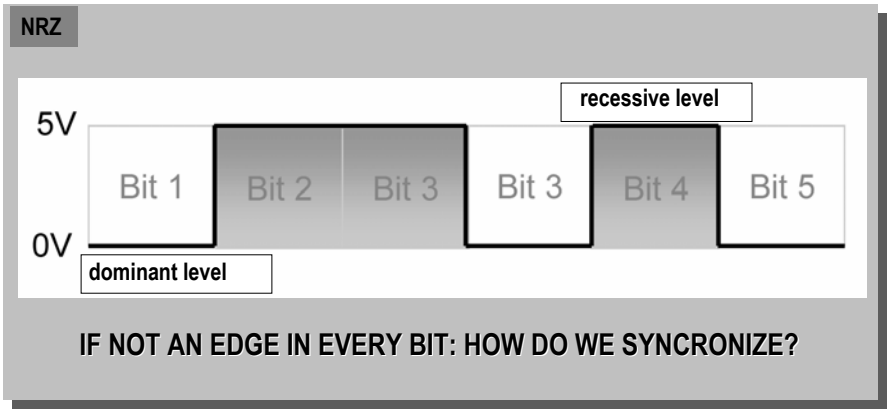
SCL may be stretched up to 30-70

CAN Bus

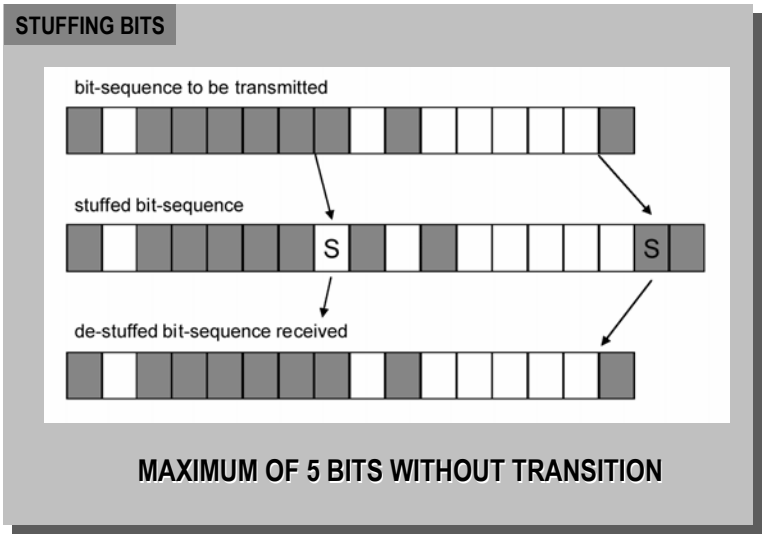
Physical Layer



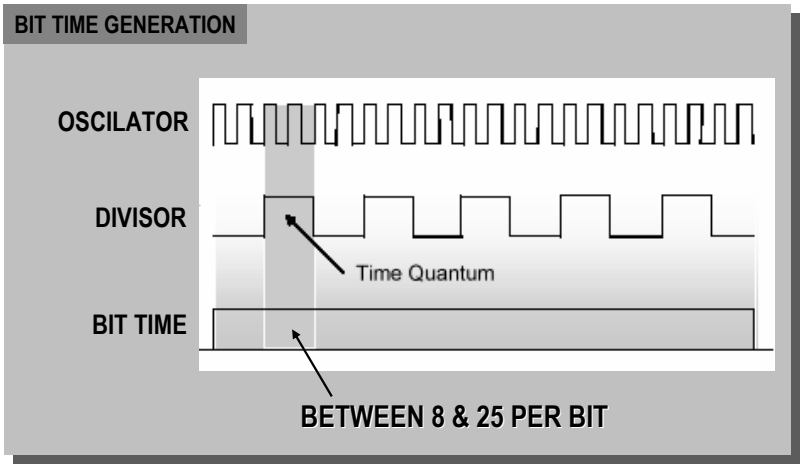
1) Sublevel MDI: Standard = 120 ohm twisted pair
2) Signaling: Current mode differential. DOMINANT state signaling
3) Controller: Syncronization, bit timing and coding-decoding.



No Return to Zero (NRZ)

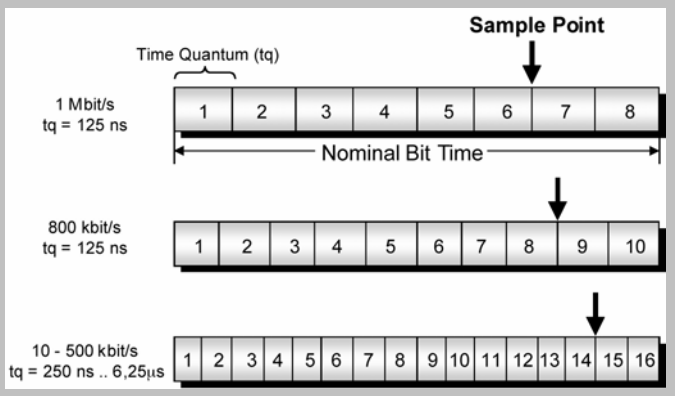


If data do not have a transition at least every 5 bit, a stuffing bit is added for resyncronization



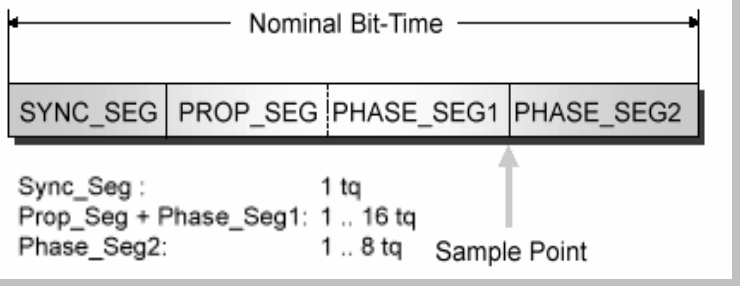
Two frequency divission stages to adjust bit time

TRANSMISSION SPEED ADJUST



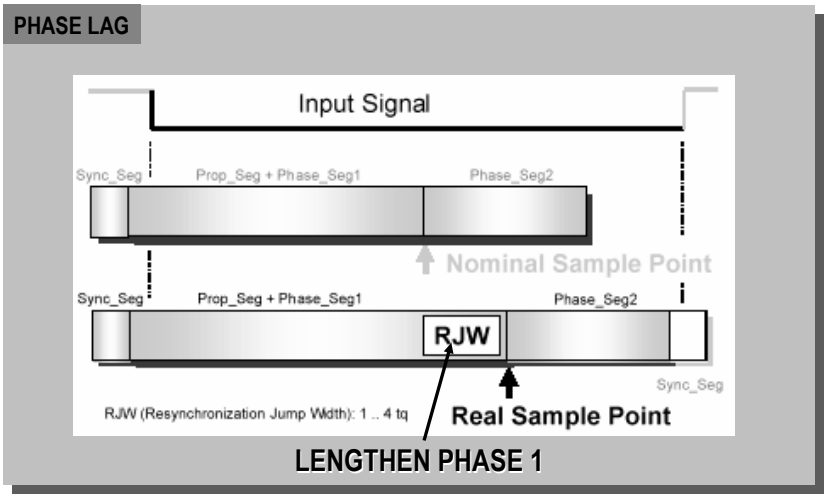
TWO OPTIONS: DIVIDING FACTOR & QUANTUM NUMBER

1 BIT = 4 SEGMENTS

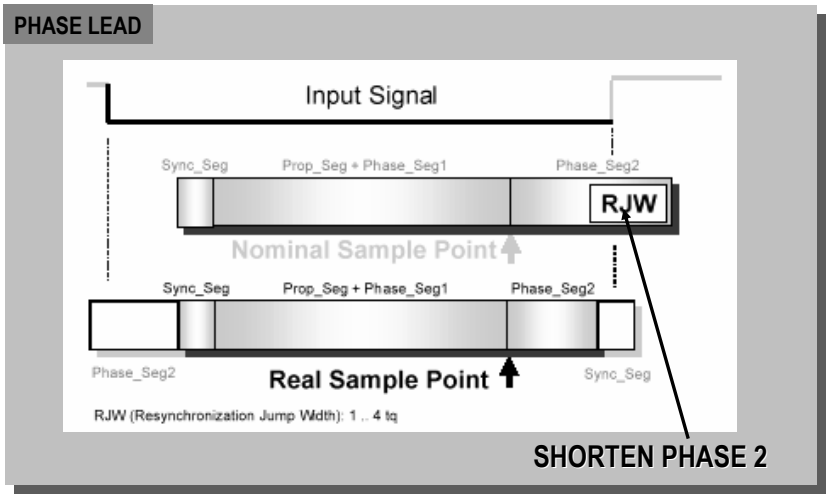


SEGMENT FUNCTIONALITY

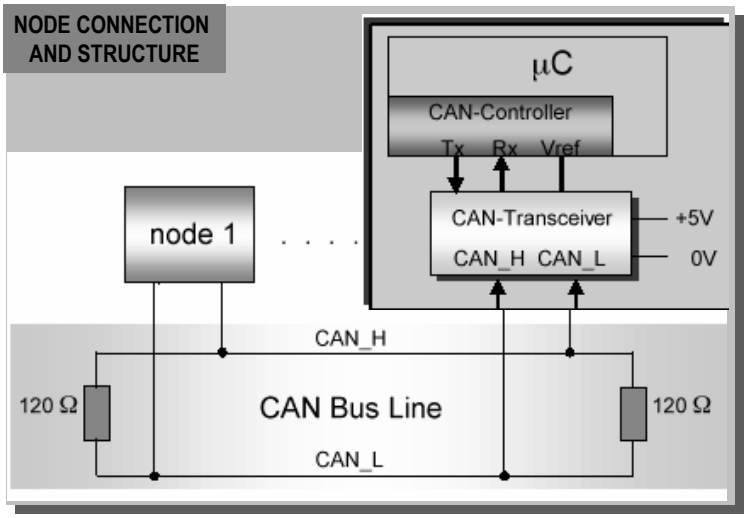
- SYNC: INDICATES THE BIT START
- PROP: COMPENSATES INTER-NODE DELAY
- PHASE 1: COMPENSATES PHASE LAG
- PHASE 2: COMPENSATES PHASE LEAD



RJW (*Resincronization Jump Width*). Lengthens phase_1 to recover syncronization.



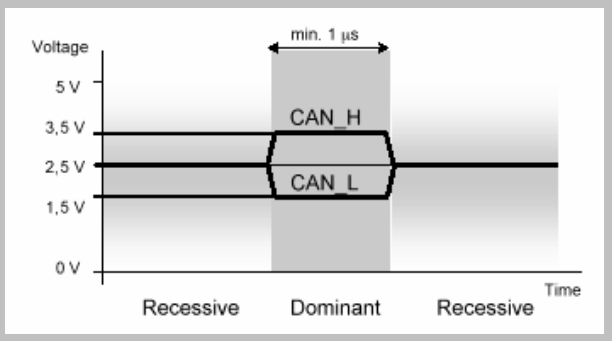
RJW (*Resincronization Jump Width*). Shortens phase_2 to recover syncronization



Only end nodes have 120 ohm resistor.

All nodes have to be able to act as a *master*.

BUS LEVELS



RECESSVE: CAN_H-CAN_L < 0.5V
DOMINANT: CAN_H-CAN_L > 0.9V

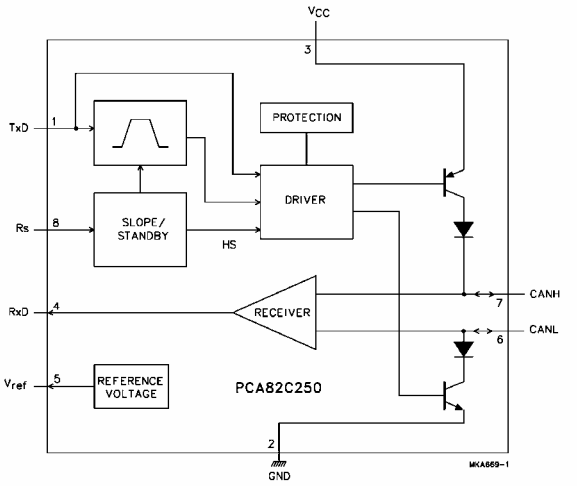
Transmitter:

RECESSIVE: CAN_H-CAN_L < 0.1V
DOMINANT: CAN_H-CAN_L > 2V

Receiver:

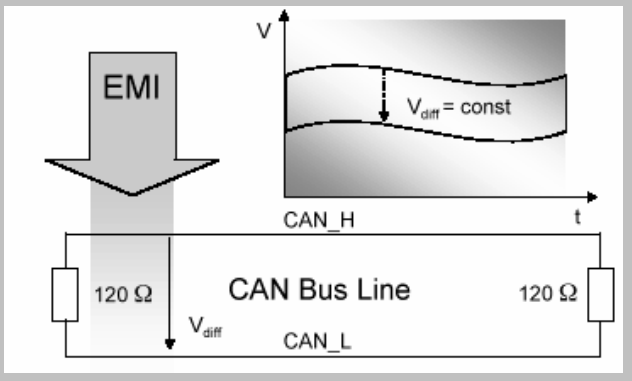
RECESSIVE: CAN_H-CAN_L < 0.5V
DOMINANT: CAN_H-CAN_L > 0.9V

BUS DRIVERS



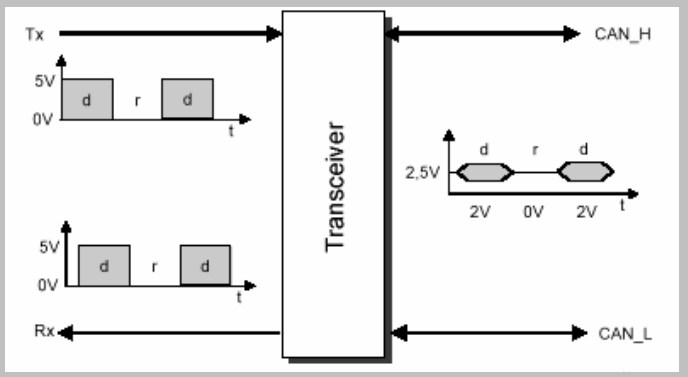
Also week Pull-up resistors of 15Kohm approx.

DIFERENTIAL TRANSMISSION



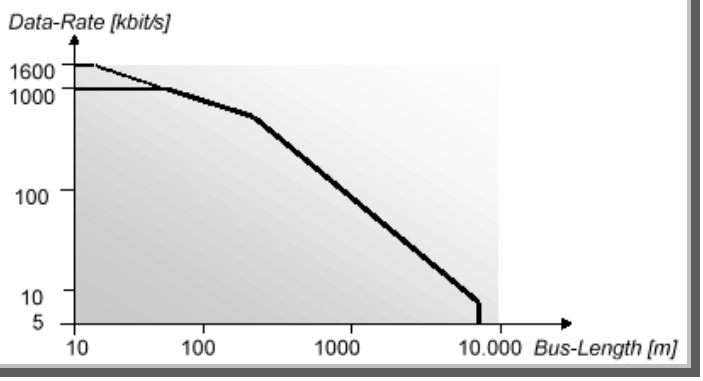
GOOD IMMUNITY TO COMMON MODE NOISE

SIGNALING CHANGE



FROM *FULL DUPLEX* TO *HALF DUPLEX*

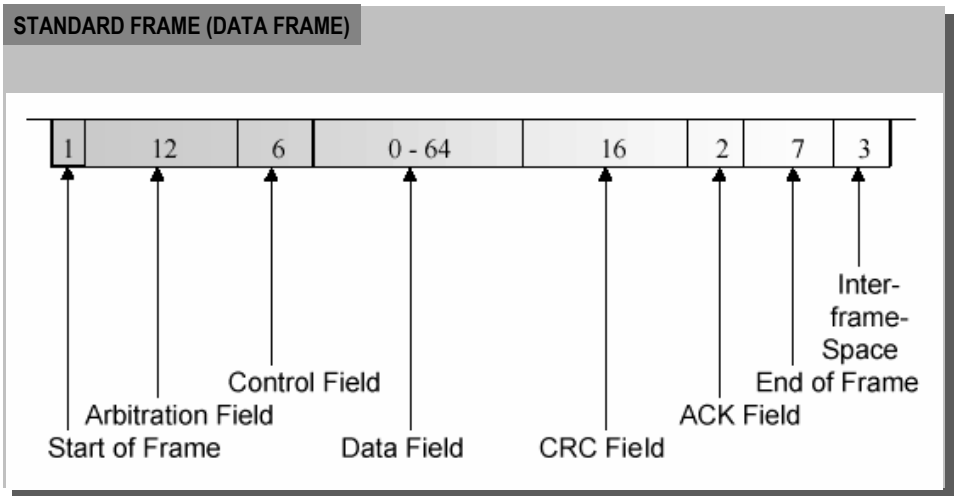
SPEED-LENGTH RELATIONSHIP



Data Link Layer

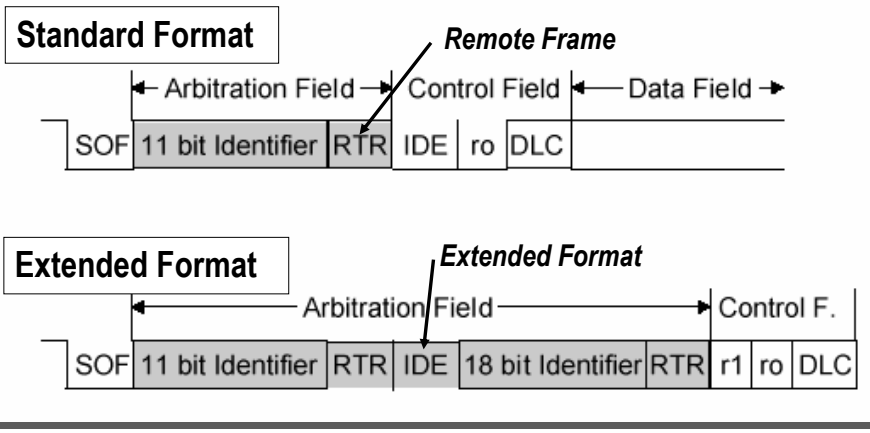
MESSAGE TYPES

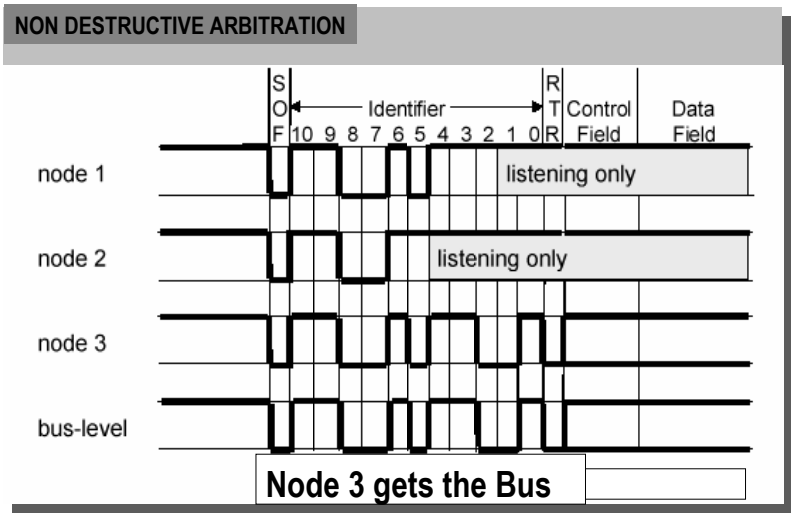
- ***DATA FRAME*: INITIATED BY TRANSMITTER (Contains data)**
- ***REMOTE FRAME*: INITIATED BY RECEIVER (Data inquiry)**
- ***ERROR FRAME*: ERROR SIGNALING (Active or passive)**
- ***OVERLOAD FRAME*: RETRANSMISSION INQUIRY (Delayed)**



It is not an address oriented bus but a message oriented bus. Arbitration field DOES NOT MAKE REFERENCE TO THE NODE BUT TO THE MESSAGE ITSELF. The same identifyer may be used by more than one node.

ARBITRATION FIELD FORMAT





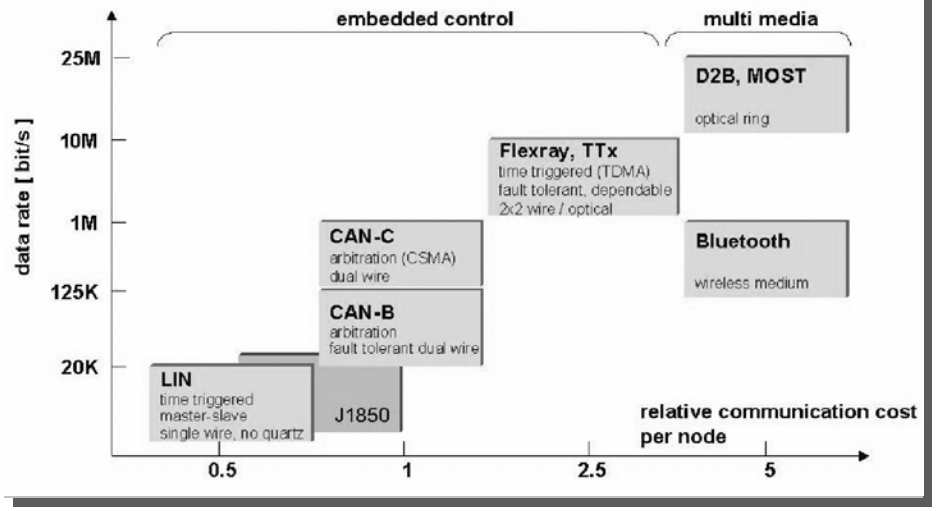
Same that I2C.

LIN Bus

WHY *LIN* BUS

- COST REDUCTION IS VERY IMPORTANT IN THE AUTOMOTIVE INDUSTRY
- *CAN* BUS FULFILLS THE ROLE BUT TOO EXPENSIVE FOR SIMPLE FUNCTIONS
- SIMPLER PROTOCOL NEEDED FOR SHORT DISTANCE AND LOW SPEED

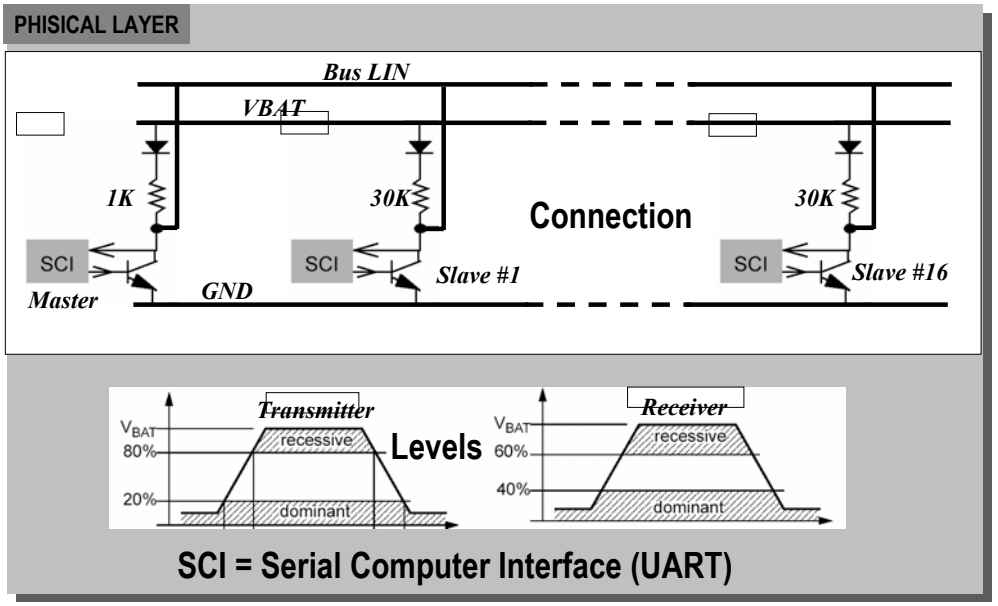
COST-SPEED COMPARISON



MAIN CARACTERISTICS

- Bidirectional communication through a single wire
- One master and up to 16 slaves
- Dominant-recessive voltage signaling
- Mixture of I2C and RS232
- RC clock with resincronization
- 20 Kbaud maximum speed
- 40 m maximum length
- Can operate along with CAN Bus

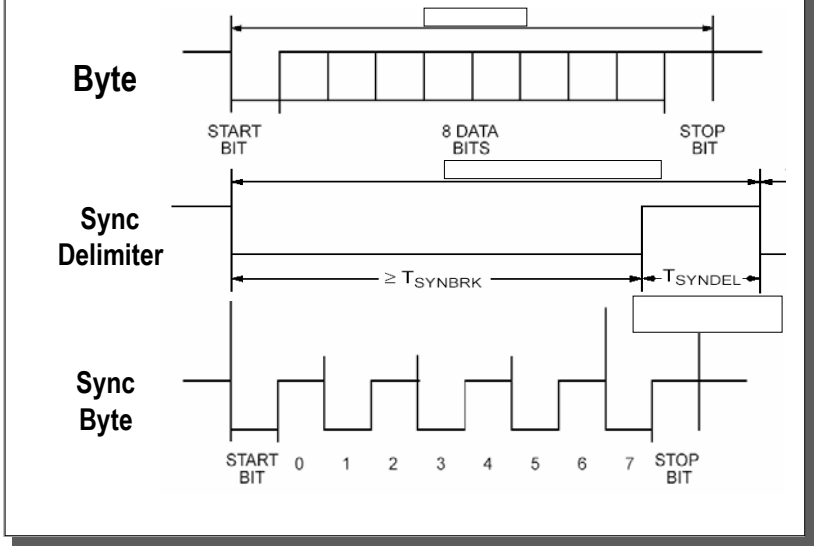
Physical Layer

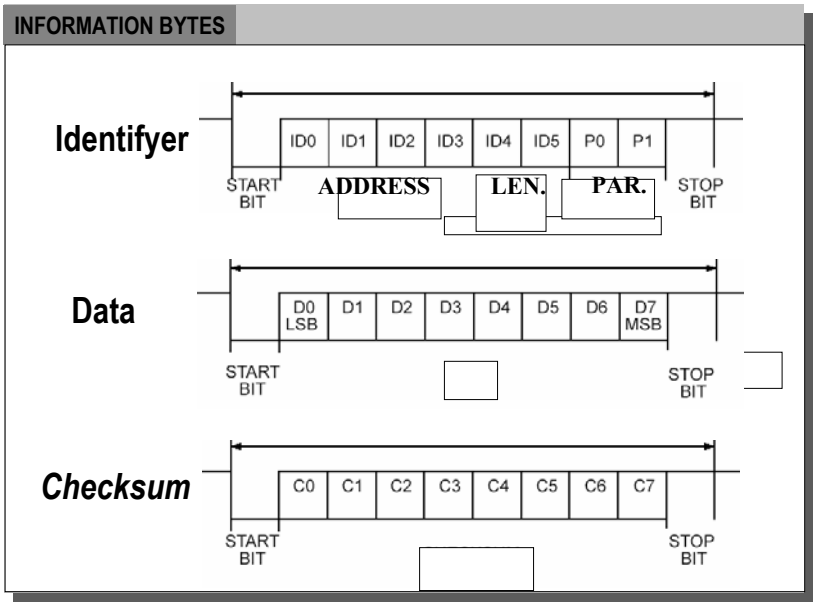


DOMINANT-RECESSIVE signaling. The master has a P.U. resistor of 1kohm. Each slave has a week P.U. resistor of 30kohm. The series diode is to protect against polarity reversal (always required in automotive environment)

Data Link Layer

BYTE STRUCTURE AND SPECIAL BYTES





Identifyer has:

1) Four address bits (16 slaves)
2) Two length bits (2,4 or 8 bytes message)
3) Two parity bits

MESSAGE FRAME