

Communication (II)

Kai Huang



Google's Army of Robots 2013

- Robotics companies acquired by Google this year include Schaft, Industrial Perception, Meka, Redwood Robotics, Bot & Dolly, Autofuss and Holomni.



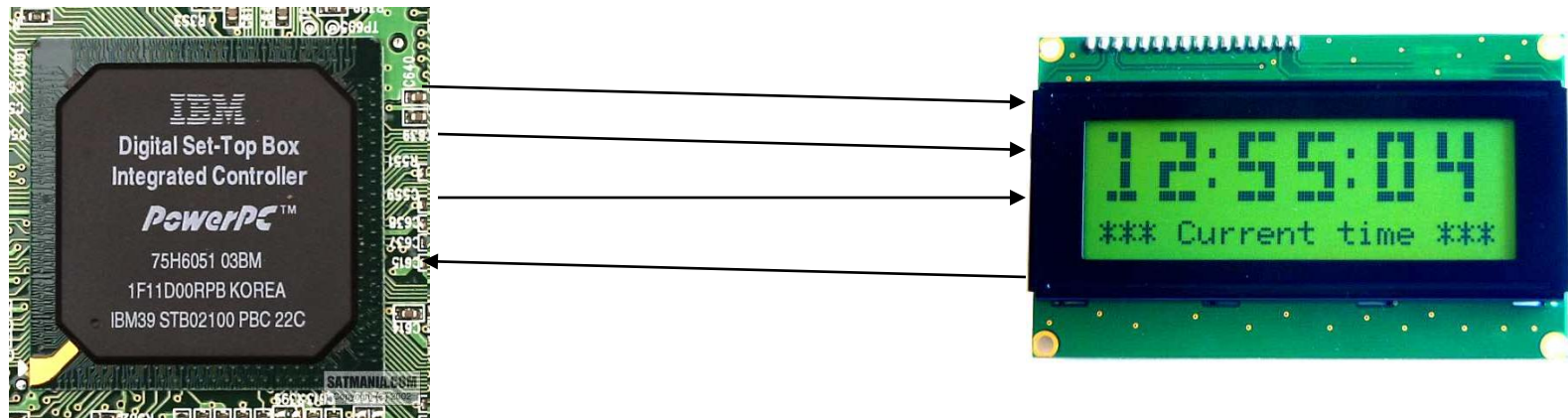
Outline

- Bus basics
- Multiple Master Bus
 - Centralized arbitration
 - Distributed arbitration
- Network-on-Chip
- **Examples**
 - SPI
 - CAN
 - FlexRay
 - Ethernet



What is SPI?

- Serial Peripheral Interface (SPI)
 - Serial Bus protocol (developed by Motorola)
 - Fast, Easy to use, Simple
 - Everyone supports it



SPI Basics

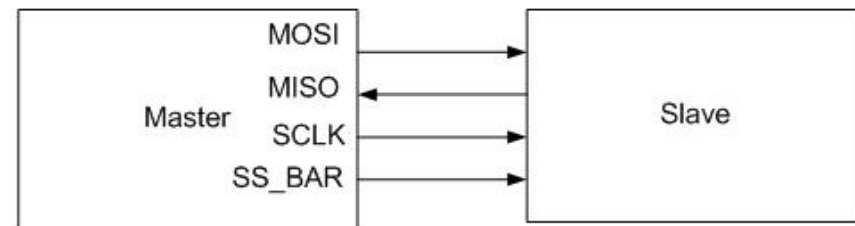
- A communication protocol using 4 wires
 - Also known as a 4 wire bus
- Used to communicate across small distances
- **Multiple Slaves, Single Master**
- Synchronized

- Capabilities
 - Always Full Duplex
 - Communicating in two directions at the same time
 - Transmission need not be meaningful
 - Multiple Mbps transmission speed
 - Transfers data in 4 to 16 bit characters Multiple slaves
 - Daisy-chaining possible



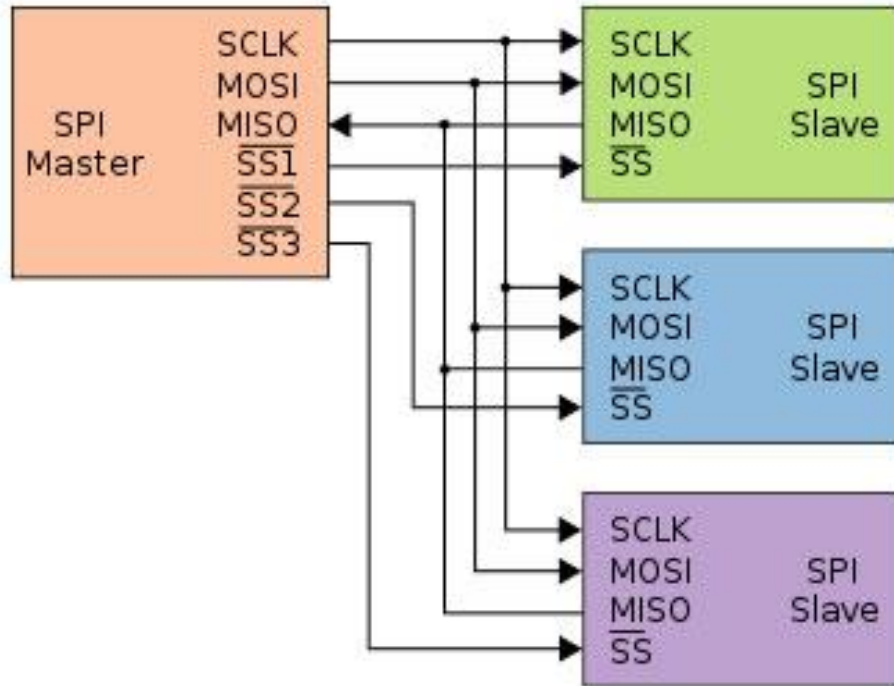
SPI Protocol

- Wires:
 - MOSI: Master Out Slave In
 - MISO: Master In Slave Out
 - SCLK: System Clock
 - SS_BAR: Slave Select 1...N

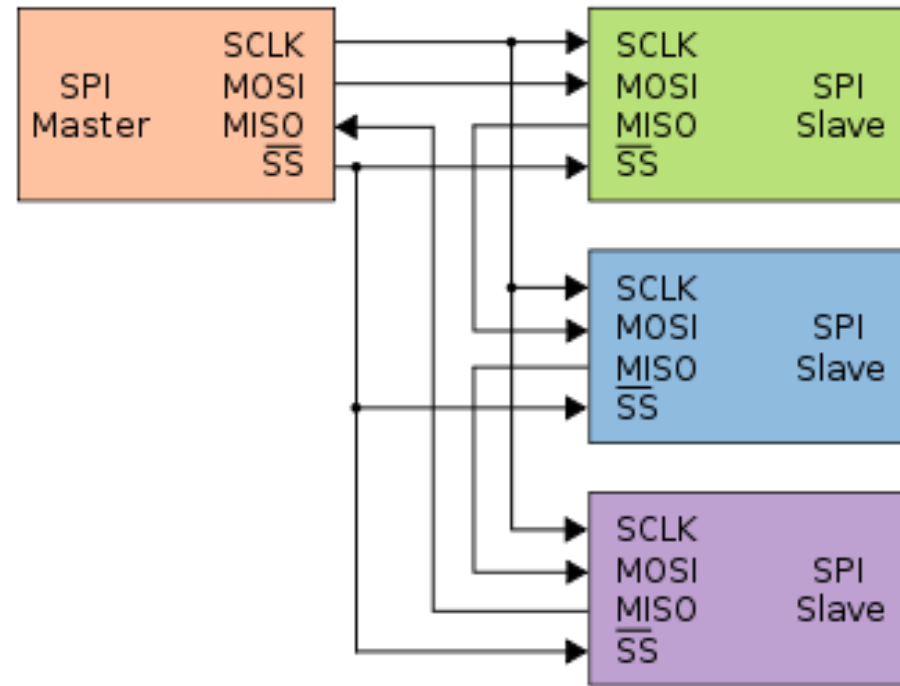


- Master Set Slave Select low
- Master Generates Clock
- Shift registers shift in and out data

Diagram



Master and multiple independent slaves



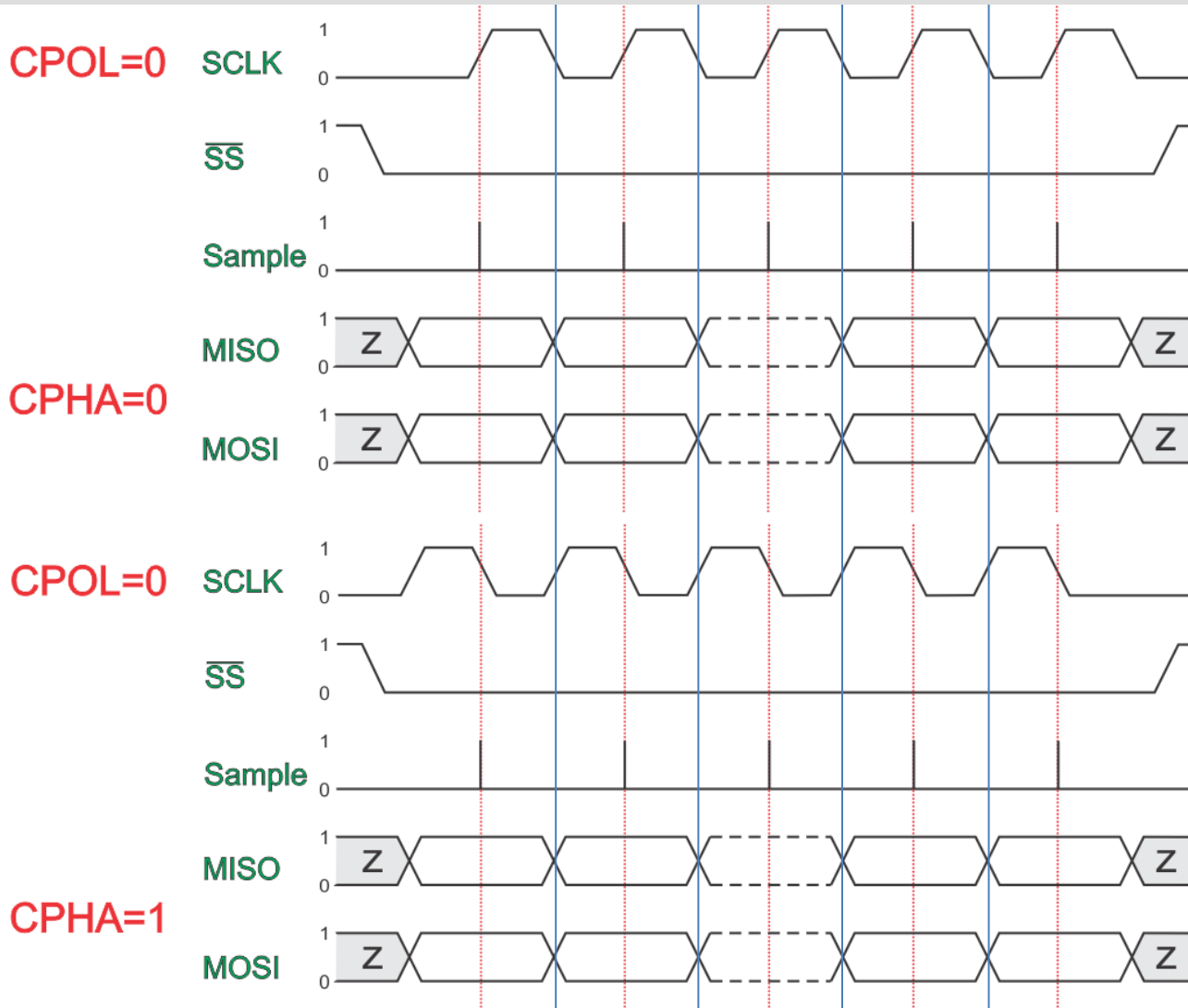
Master and multiple daisy-chained slaves

Clock Phase

- Two phases and two polarities of clock
 - Four modes defined by
 - CPOL (Clock POLarity)
 - CPHA (Clock PHAse)
- Master and selected slave must be in same mode
- Master must change polarity and phase to communicate with slaves of different numbers



CPOL = 0



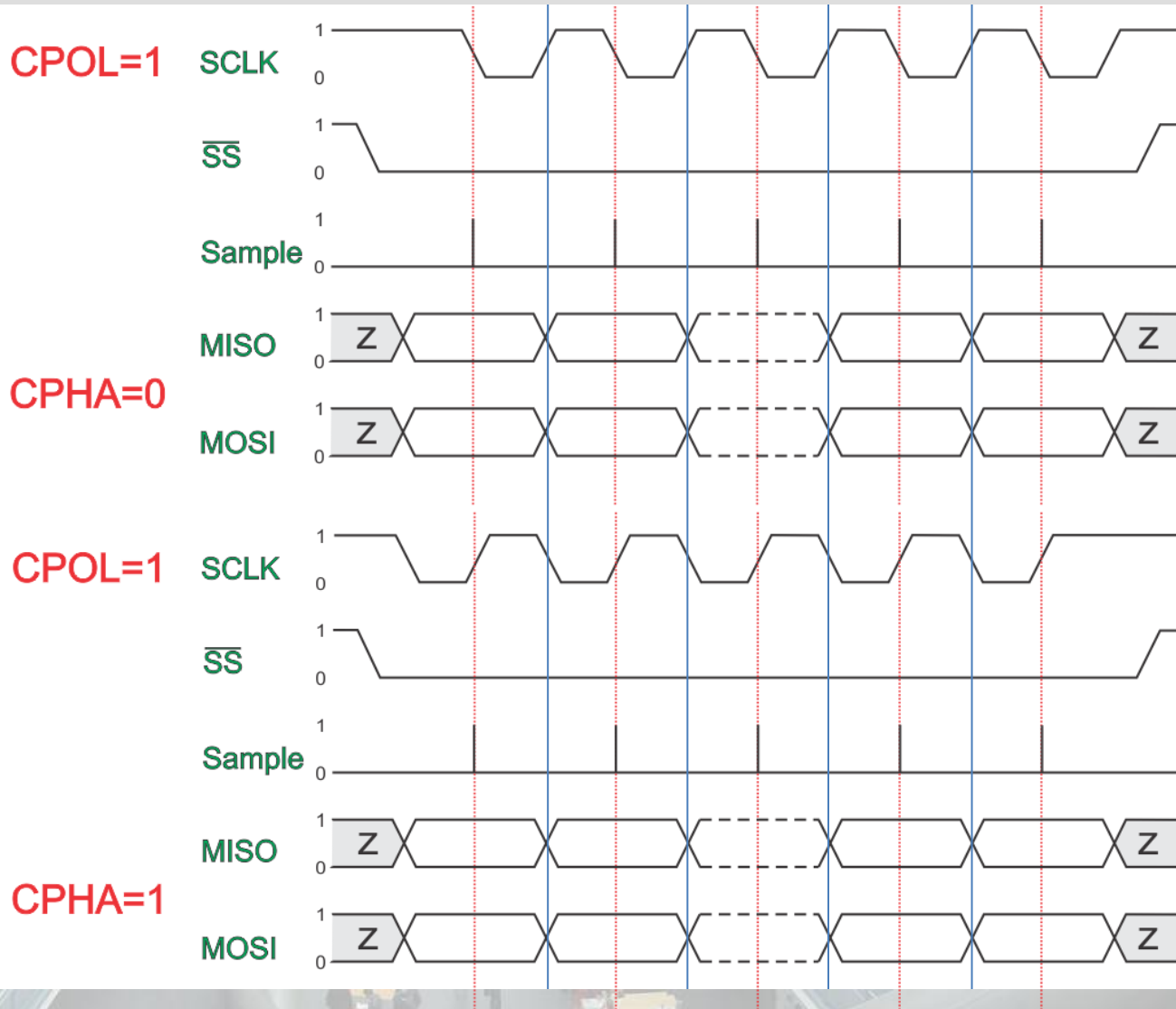
CPOL=0: Base value of the clock is zero

CPHA=0:
- Data captured on rising edge,
- Data propagated at falling edge

CPHA=1:
- Data captured on falling edge,
- Data propagated at rising edge



CPOL = 1



CPOL=0: Base value of the clock is one

CPHA=0:
- Data captured on rising edge,
- Data propagated at falling edge

CPHA=1:
- Data captured on falling edge,
- Data propagated at rising edge



Pros and Cons

■ Pros:

- Fast and easy
 - Fast for **point-to-point** connections
 - Easily allows streaming/Constant data inflow
 - No addressing/Simple to implement
- Everyone supports it

■ Cons:

- SS makes multiple slaves very complicated
- No acknowledgement ability
- No inherent arbitration
- No flow control



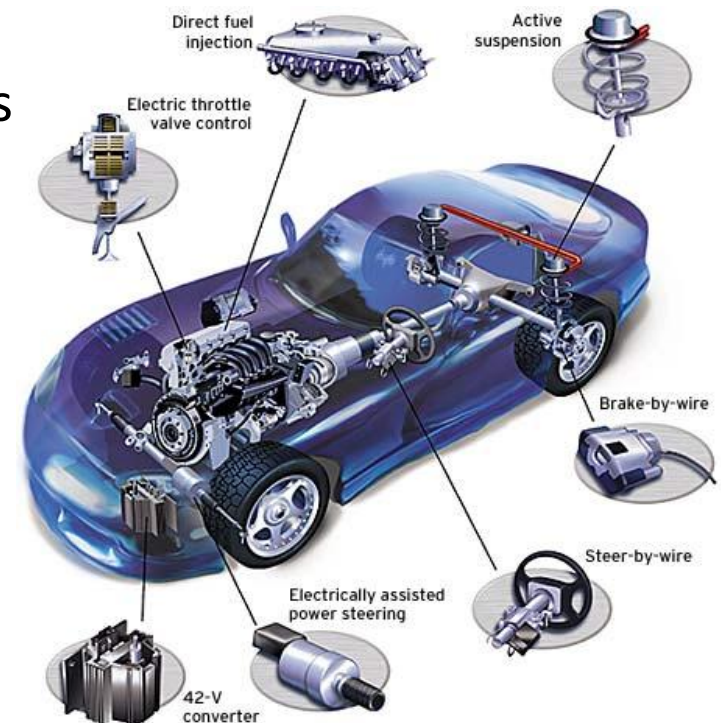
Outline

- Bus basics
- Multiple Master Bus
 - Centralized arbitration
 - Distributed arbitration
- Network-on-Chip
- **Examples**
 - SPI
 - **CAN**
 - FlexRay
 - Ethernet



What is CAN?

- CAN (Controller Area Network) was developed in 1983 by Bosch
- CAN is a **multi-master broadcast serial bus**
 - All devices on the network receive every bit of information sent on the BUS
- CAN is message-based (each message is uniquely identified by an ID)
 - **Event triggered**
- CAN uses CSMA/CR for media access control
- Typically used in automotive and automation tasks/scenarios



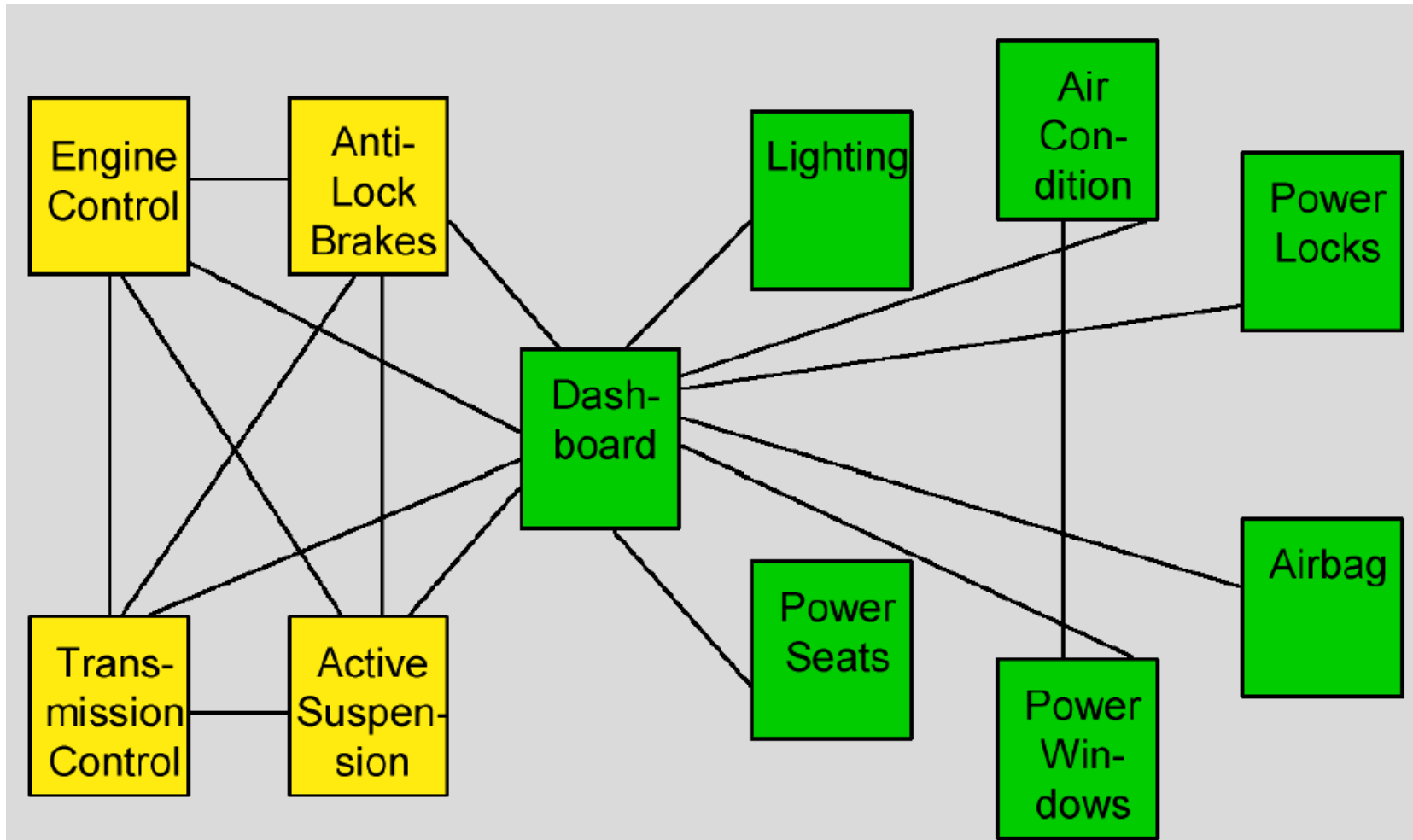
Why is CAN used?

- **Simple** Transmission Medium
 - Twisted pair of wires is the standard, but also just one wire will work
- Excellent Error Handling
 - CRC error detection mechanism
- Fault Confinement
 - Built-in feature to prevent faulty node to block system
- Maximum bitrates depend on bus length
- Mature Standard
 - CAN protocol more than 16 years
 - Numerous CAN products and tools on the market
 - ISO 11898
- ❖ License fee to Borsch for every controller ...

Max. Length	Max Bitrate
40 m	1 Mbit/s
100 m	500 kbit/s
500 m	125 kbit/s

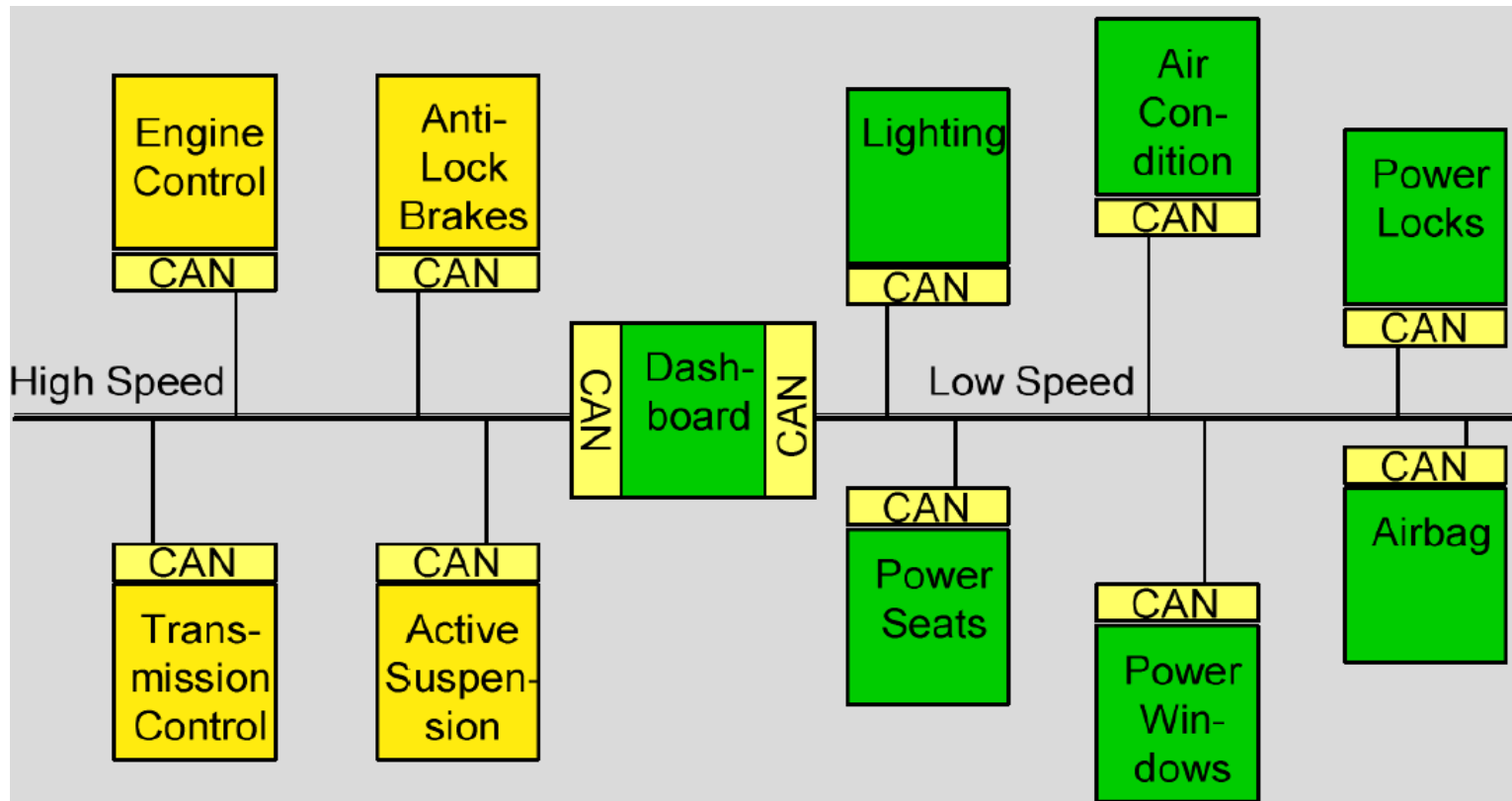


Before CAN



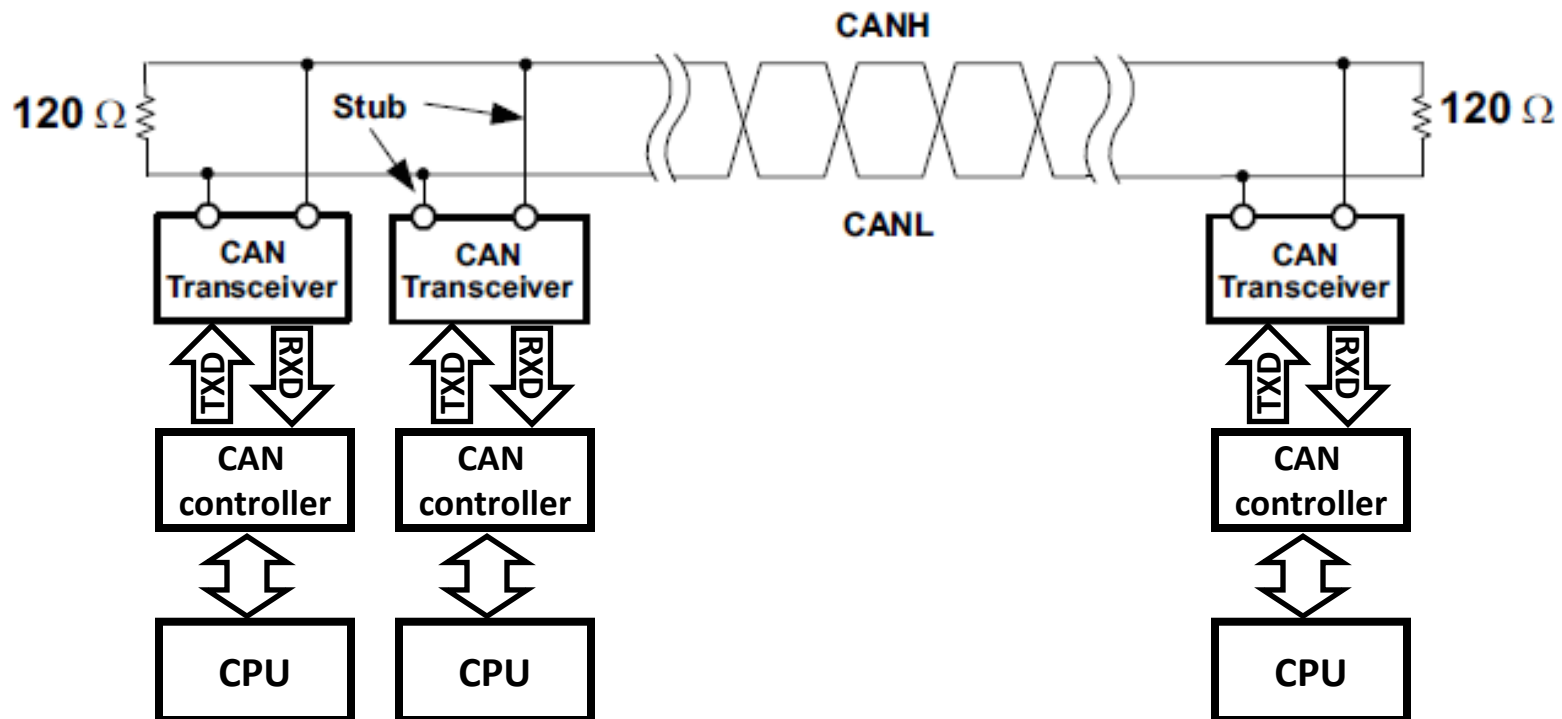
With CAN

- Point-to-point wiring is replaced by one serial bus connecting all control systems



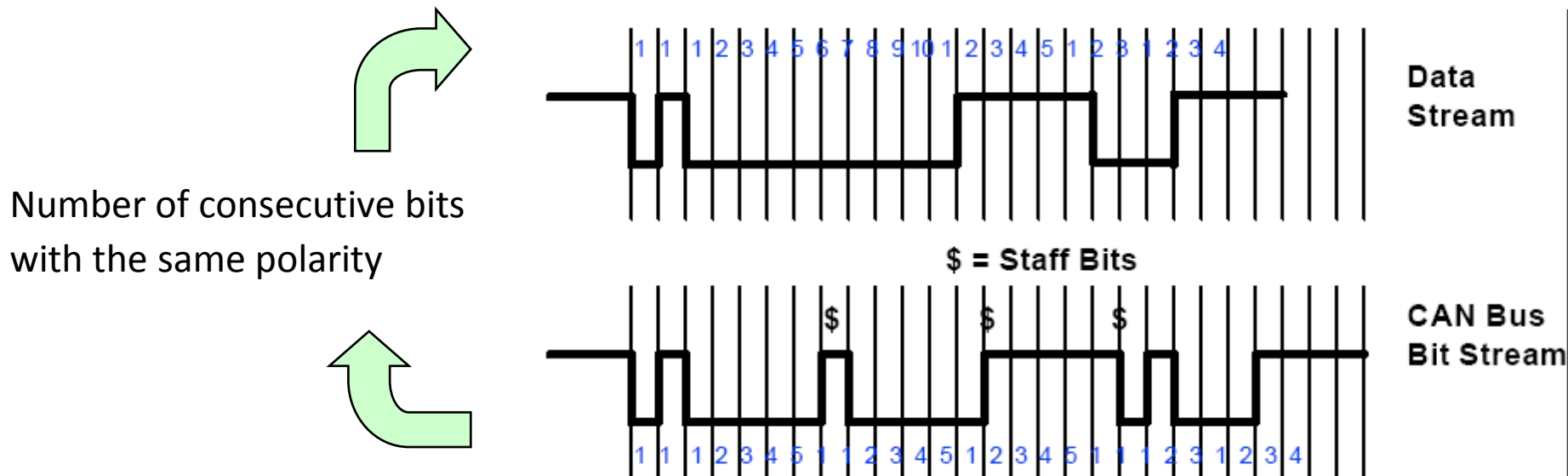
Basic Configuration

- Terminated on each side by a 120 Ohm resistor
- A simple 2-wire differential serial bus



Bit Coding & Bit Stuffing

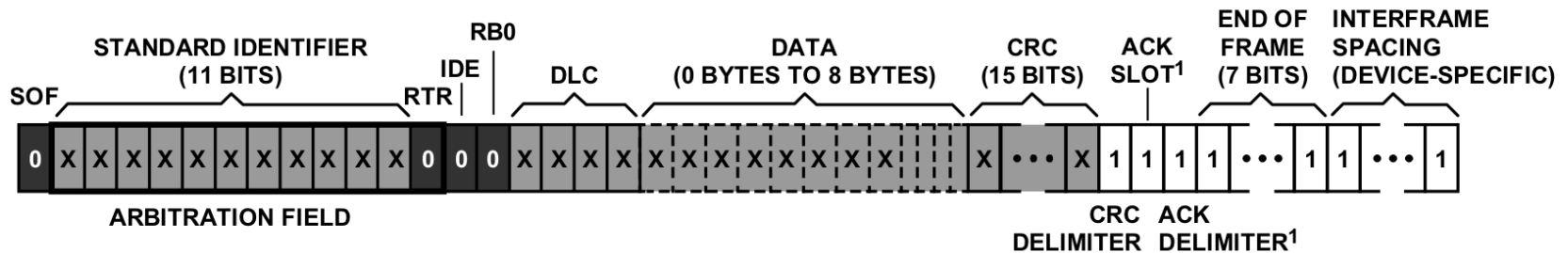
- Bit Coding : NRZ (Non-Return-To-Zero code) does not ensure enough edges for synchronization
- Stuff Bits are inserted after 5 consecutive bits of the same level
- Stuff Bits have the inverse level of the previous bit.
- No deterministic encoding, frame length depends on transmitted data



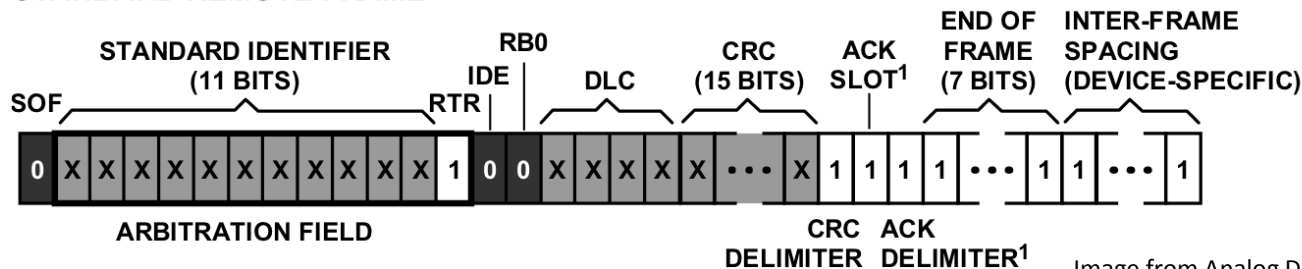
CAN Standard Message Frame Fields

- Four types of frames:
 - Data frame: Used for data transmission
 - Remote frame: Used for requesting data
 - Error frame: Used to notify errors
 - Overload frame: Enforces delay before next data or remote frame

STANDARD DATA FRAME



STANDARD REMOTE FRAME



KEY:

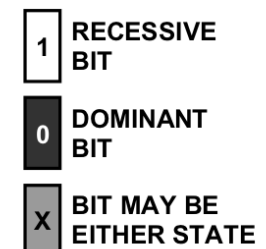
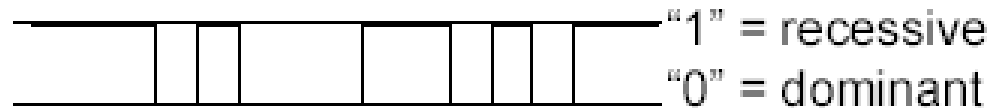


Image from Analog Devices AN-1123 Application Note datasheet



Wired AND

Two logic states possible on the bus:
"1" = recessive
"0" = dominant



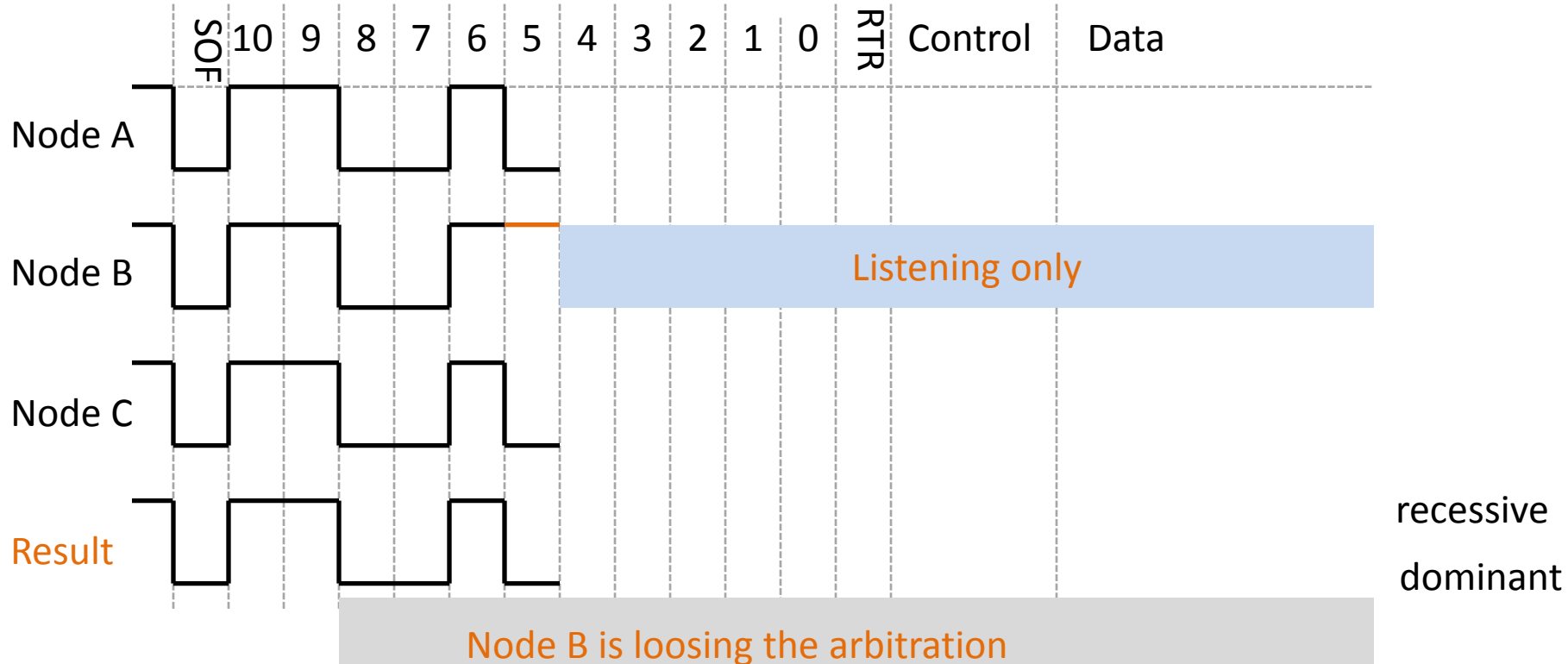
A	B	C	BUS
D	D	D	D
D	D	R	D
D	R	D	D
D	R	R	D
R	D	D	D
R	D	R	D
R	R	D	D
R	R	R	R

As soon as one node nodes transmits a dominant bit (zero):
Bus is in the dominant state.

Only if all nodes transmit recessive bits (ones):
Bus is in the recessive state.

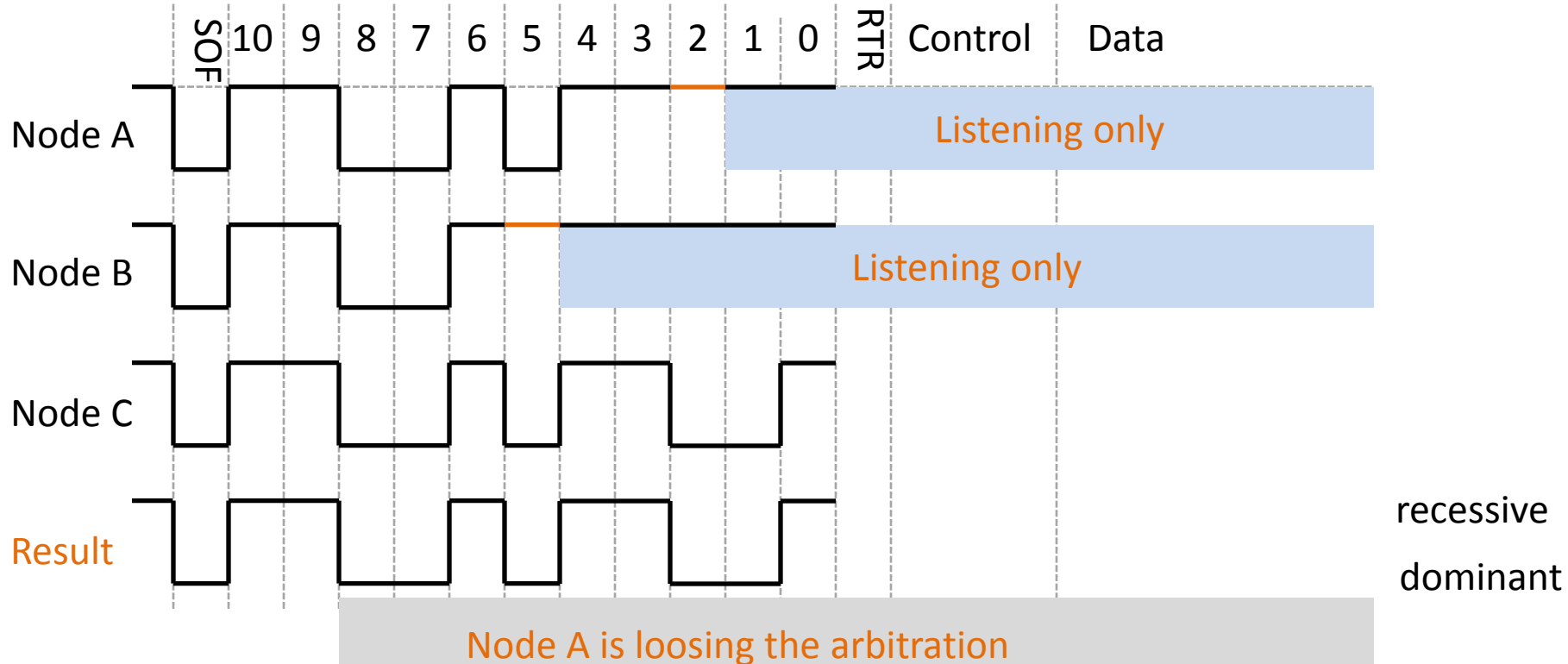
Arbitration Scheme

- CAN implements an arbitration scheme based on message identifiers to achieve collision avoidance
- Transmitted bits are either **dominant** (typically 0) or **recessive** (1).



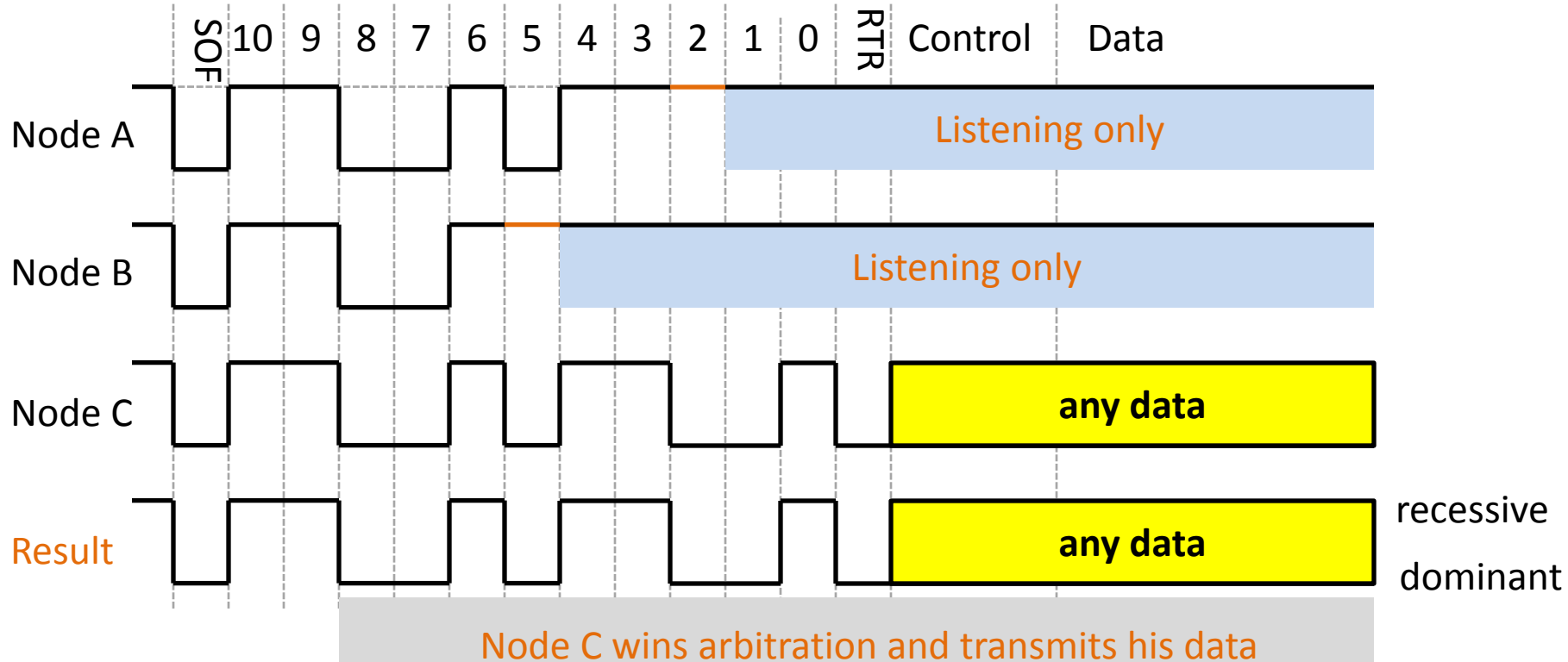
Arbitration Scheme

- CAN implements an **arbitration** scheme based on message identifiers to achieve collision avoidance (CA)
- Transmitted bits are either **dominant** (typically 0) or **recessive** (1).



Arbitration Scheme

- CAN implements an **arbitration** scheme based on message identifiers to achieve collision avoidance (CA)
- Transmitted bits are either **dominant** (typically 0) or **recessive** (1).



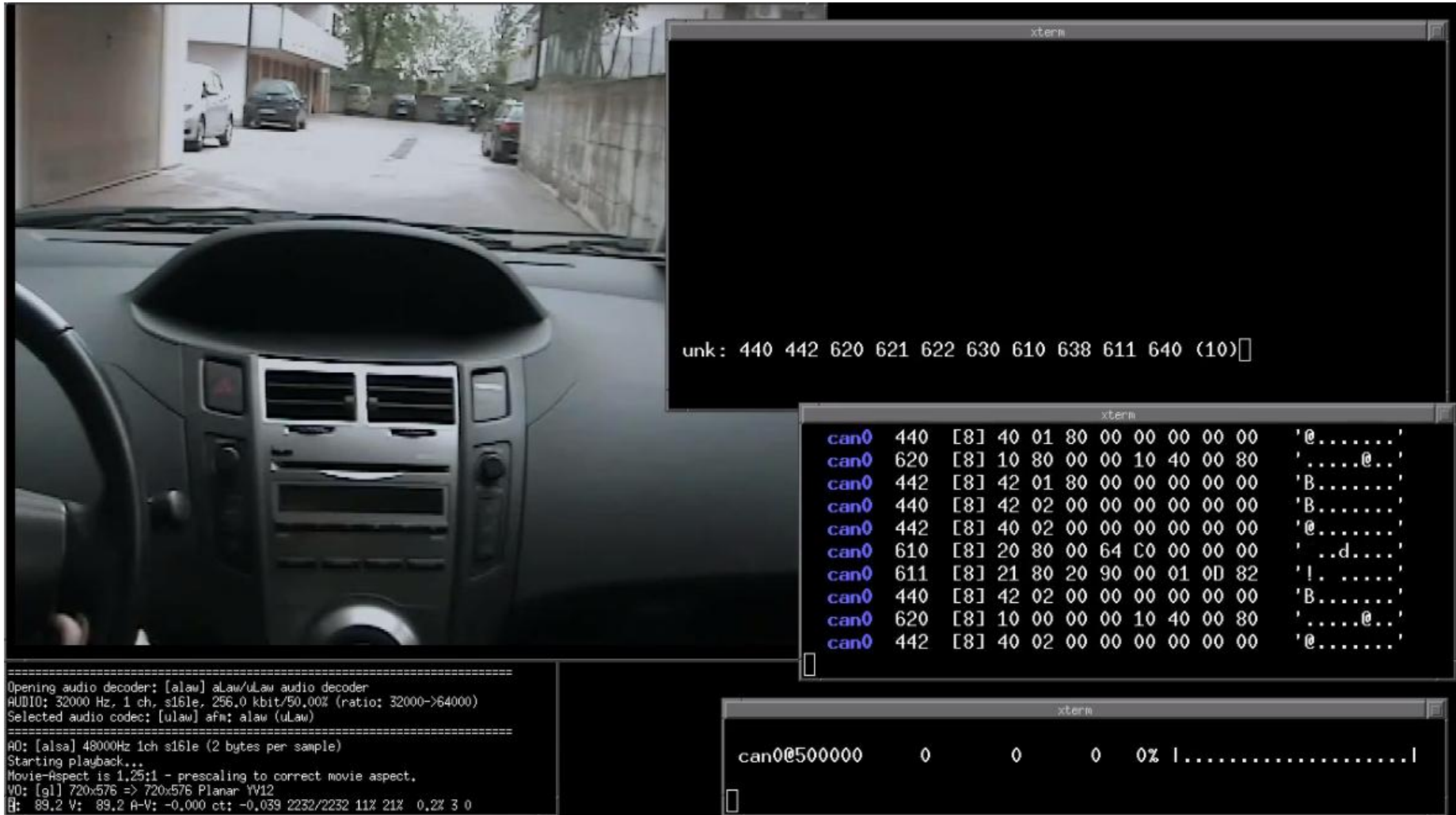
SocketCAN

- CAN bus stack for the Linux kernel, contributed by a team at Volkswagen Research
 - From an application point-of-view this is great news, as it means that CAN bus is handled by the well know Berkley sockets APIs and you have full software portability
 - <http://developer.berlios.de/projects/socketcan/>
- Commands:
 - ip: control the CAN interface
 - candump: record the packages on a CAN interface
 - canplayer: playback

<http://fabiobaltieri.com/2013/07/23/hacking-into-a-vehicle-can-bus-toyothack-and-socketcan/>



Example of SocketCAN



The image is a composite showing a car's interior dashboard on the left and three terminal windows on the right displaying CAN bus data. The terminal windows show the following data:

```
unk: 440 442 620 621 622 630 610 638 611 640 (10)[]
```

```
can0 440 [8] 40 01 80 00 00 00 00 00 '@.....'
can0 620 [8] 10 80 00 00 10 40 00 80 '.....@..'
can0 442 [8] 42 01 80 00 00 00 00 00 'B.....'
can0 440 [8] 42 02 00 00 00 00 00 00 'B.....'
can0 442 [8] 40 02 00 00 00 00 00 00 '@.....'
can0 610 [8] 20 80 00 64 C0 00 00 00 '...d....'
can0 611 [8] 21 80 20 90 00 01 0D 82 '!.....'
can0 440 [8] 42 02 00 00 00 00 00 00 'B.....'
can0 620 [8] 10 00 00 00 10 40 00 80 '.....@..'
can0 442 [8] 40 02 00 00 00 00 00 00 '@.....'
```

```
can0@500000 0 0 0 0% |.....|
```

Opening audio decoder: [alaw] aLaw/uLaw audio decoder
AUDIO: 32000 Hz, 1 ch, s16le, 296.0 kbit/50.00% (ratio: 32000->64000)
Selected audio codec: [ulaw] afm: alaw (uLaw)

AO: [alsa] 48000Hz 1ch s16le (2 bytes per sample)
Starting playback...
Movie-Aspect is 1.25:1 - pre-scaling to correct movie aspect.
VO: [gl] 720x576 => 720x576 Planar YV12
B: 89.2 V: 89.2 A-V: -0.000 ct: -0.039 2232/2232 11% 21% 0.2% 3 0

<http://fabiobaltieri.com/2013/07/23/hacking-into-a-vehicle-can-bus-toyothack-and-socketcan/>



Outline

- Bus basics
- Multiple Master Bus
 - Centralized arbitration
 - Distributed arbitration
- Network-on-Chip
- **Examples**
 - SPI
 - CAN
 - **FlexRay**
 - Ethernet



What is FlexRay ?

- FlexRay Consortium (2000-2010): BMW, Daimler, Motorola (Freescale), Philips (NXP), Bosch, GM, VW
- Combination of a TDMA and the Byteflight [Byteflight Consortium, 2003] (Flexible TDMA, close to CSMA/CA) protocol
- FlexRay currently under conversion into ISO standard
- Operation principle:
 - Cycle is subdivided into a static and a dynamic segment.
 - Static segment is based on a fixed allocation of time slots to nodes.
 - Dynamic segment for transmission of ad-hoc communication with variable bandwidth requirements.



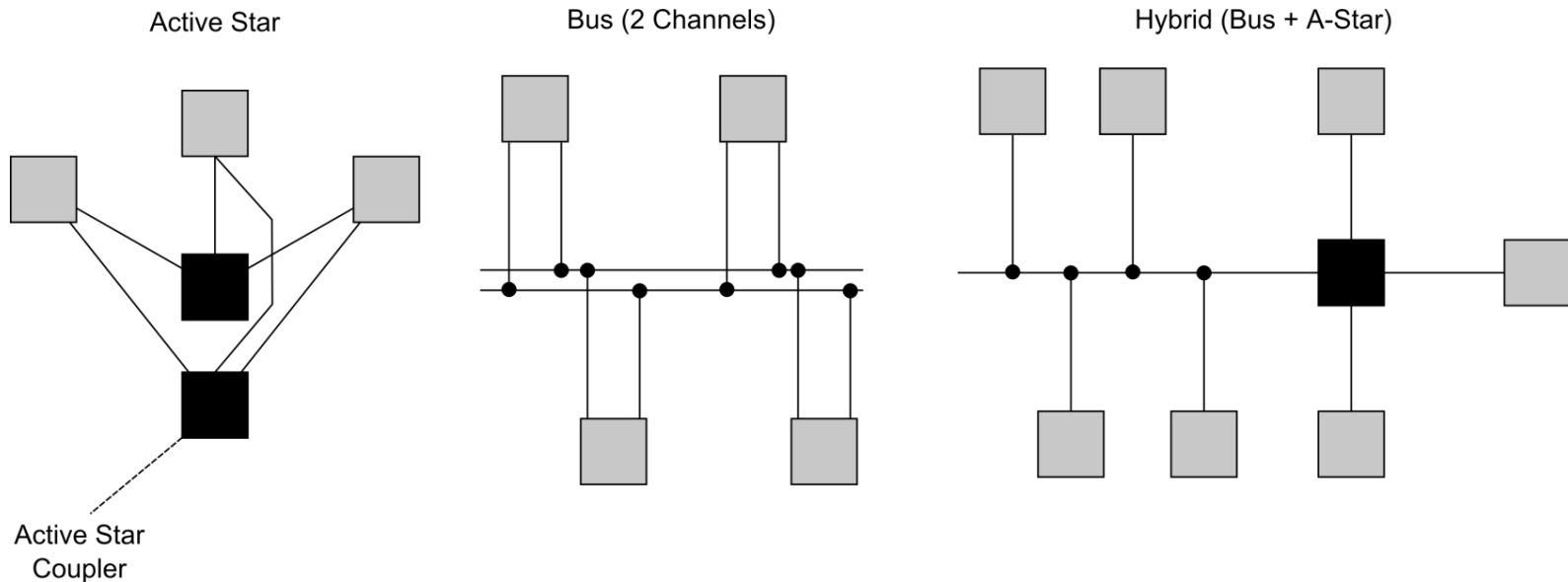
Why FlexRay ?

- High data rates can be achieved:
 - initially targeted for $\sim 10\text{Mbit/sec}$;
 - design allows much higher data rates
- Time- and event-triggered data packages
- Fault-tolerant (redundant channels, e.g. for X-by-wire systems)
- Used by premium cars (e.g. Audi A8, BMW X5)



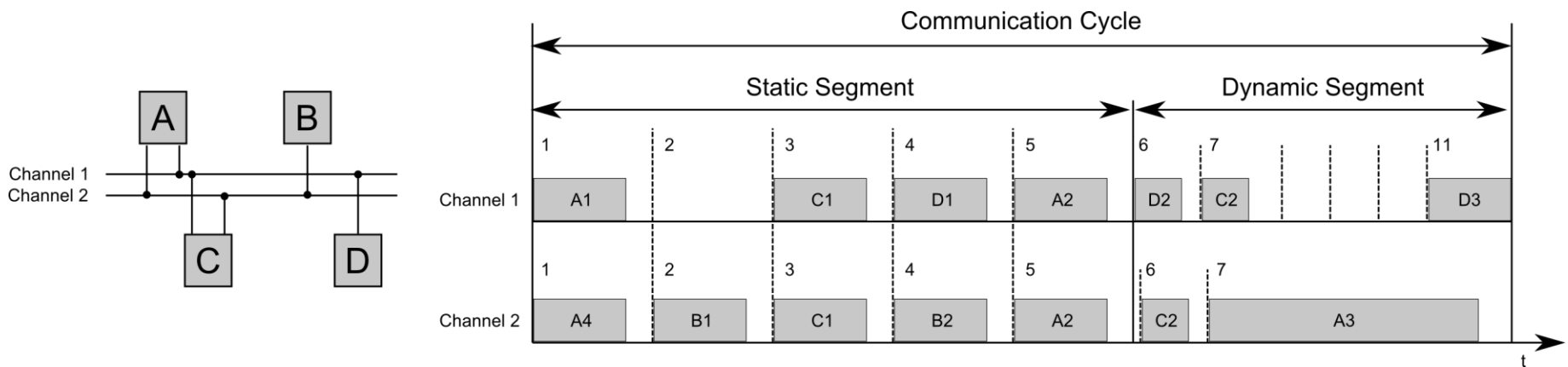
Basic topologies

- Typical Network-Topology: Active Star
- Bus or hybrid topologies are also possible
- 1 or 2 channel topology – depending on fault tolerance / bandwidth requirements
- Use of two independent channels to eliminate single-point failures



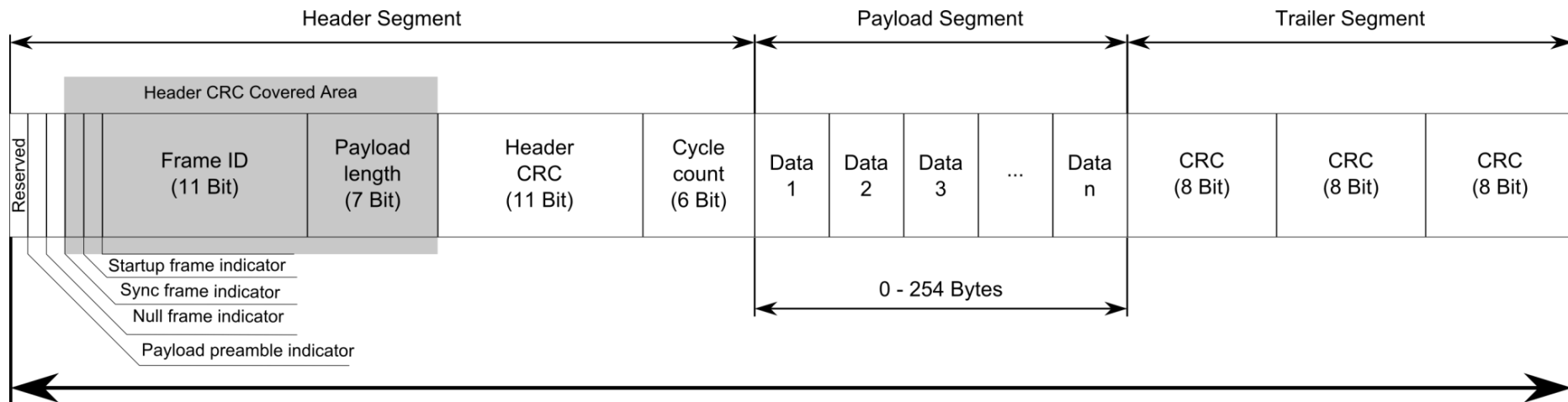
Communication

- FlexRay communication is divided in:
 - a **static** segment: Deterministic media access based on TDMA
 - a **dynamic** segment: Non-deterministic media access based on FTDMA (Flexible Time Division Multiple Access)
- Time base is established at wake-up of control units



Frame Format

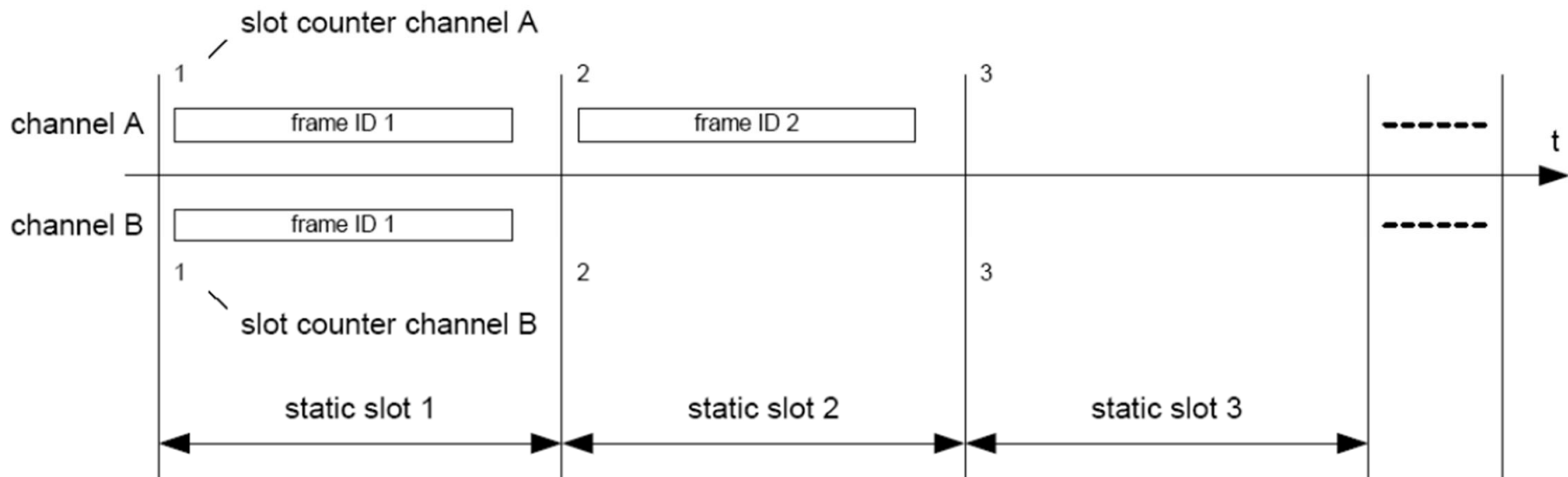
- A FlexRay **frame** is divided into **three** segments: Header, Payload and Trailer
- **Variable payload** length from 0 to 254 Bytes
- **CRC** for **header** (partly) and **payload**
- **Frame ID**: Defines the slot in which the frame should be transmitted (Range: 1-2047, ID 0 is an invalid frame ID)
- **Payload Length**: Payload data bytes



FlexRay Frame: 5 + (0-254) + 3 Byte

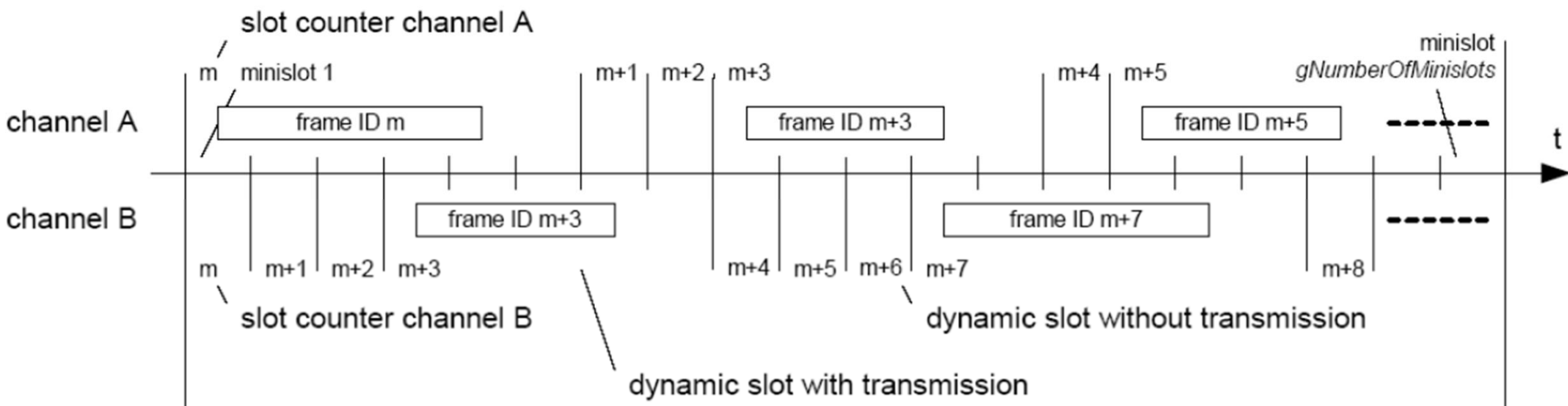
Static Segment (TDMA)

- all static slots are the same length whether used or not
- all slots are repeated in order every communication cycle
- slots are lock-stepped in order on both channels



Dynamic Segment

- Each minislot is an opportunity to send a message
- If message isn't sent, minislot elapses unused (short idle period)
- All nodes watch whether a message is sent so they can count minislots



FlexRay Tradeoffs

- Advantages
 - Probably has primitives necessary for critical x-by-wire applications
 - Static segment provides timing guarantees and some fault tolerance
 - Dynamic segment gives flexibility for event triggered messages
 - Big industry consortium behind it
 - It's "flexible"
- Disadvantages
 - Relatively new protocol – these things take time to mature
 - For example, no fault hypothesis published upon which to build safety case
 - Does not provide as complete a set of primitives as TTP
 - Group membership is an application problem, but will be needed for x-by-wire
 - Any safety critical operation on host might complicate safety case
- Others
 - Time triggered Ethernet is coming ...

