Development of a CAN Slave Module with SystemC

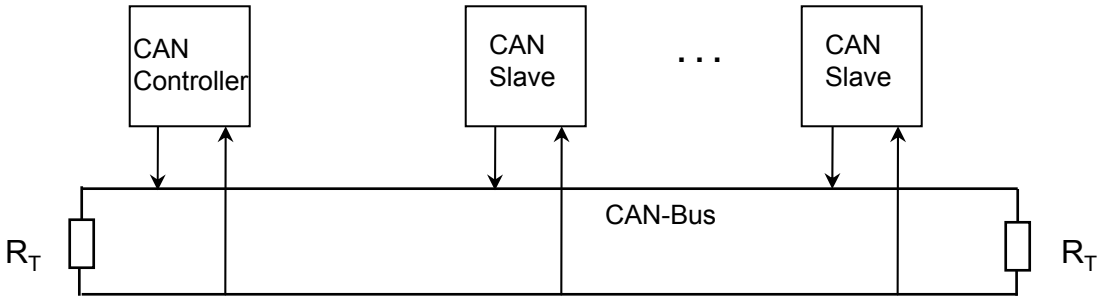
Igor Sachs
Shang Qihua

Agenda

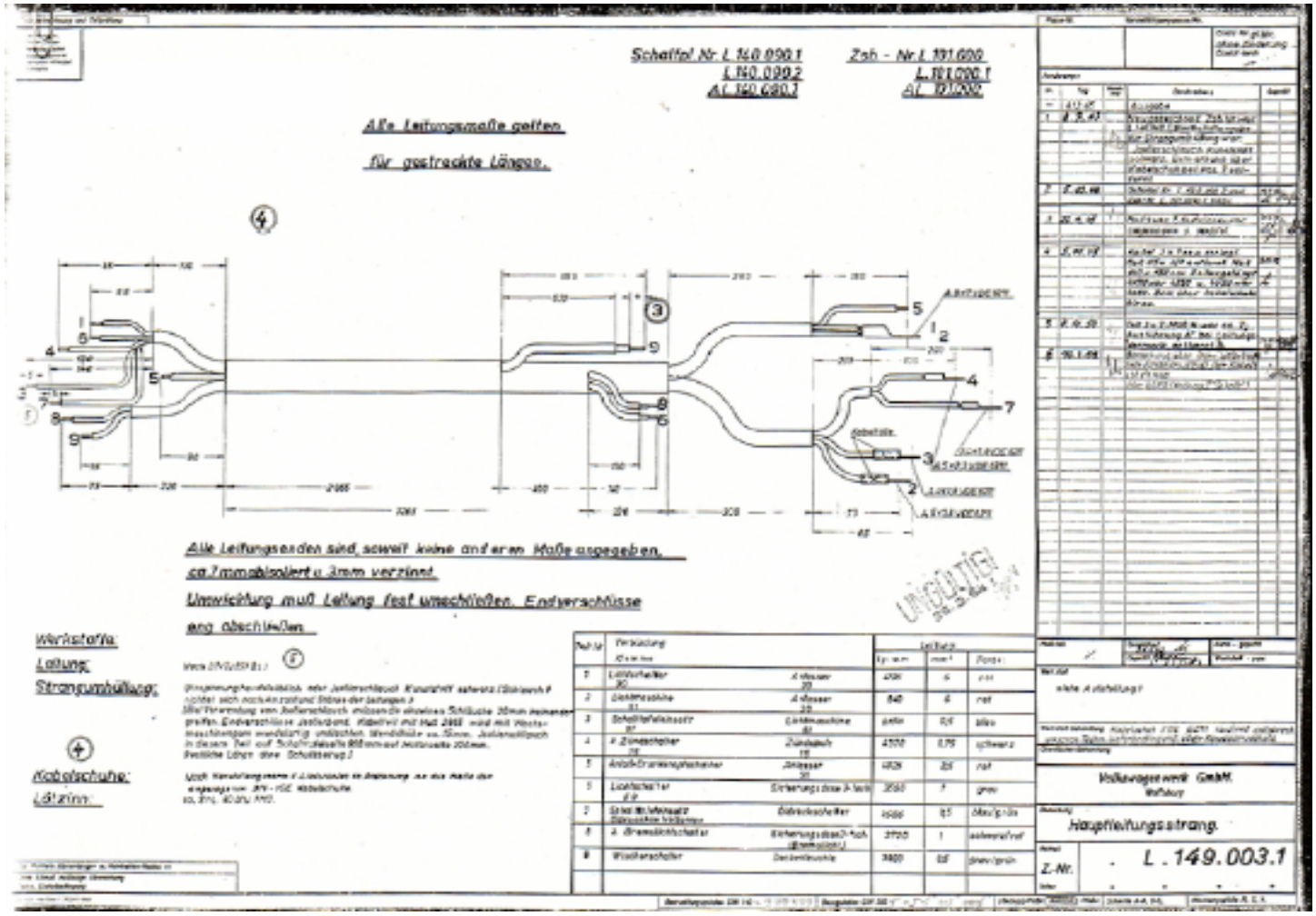
- 0. Motivation
- 1. Introduction to the CAN-Bus
  - 1.1 The CAN Message Format (Frame)
  - 1.2 Bus Arbitration
  - 1.3 Bit Stuffing
- 2. Development of the Slave Module
  - 2.1 The architecture of the Module
  - 2.2 The Finite State Machine (FSM)
  - 2.3 Problems
- 3. Presentation of the Hardware
- 4. Conclusion
- 5. Questions

Motivation

- The Controller Area Network (CAN) is a Fieldbus which has realtime capabilities and is mainly used in the automotive industry.
- The so-called Multimaster Bus was developed at Bosch in 1981 and is today the de-facto standard within Car-electronics.
- The aim of our project was to design a Slave module for the CAN Bus using the HDL SystemC and implement it into an Xilinx Spartan-2 FPGA

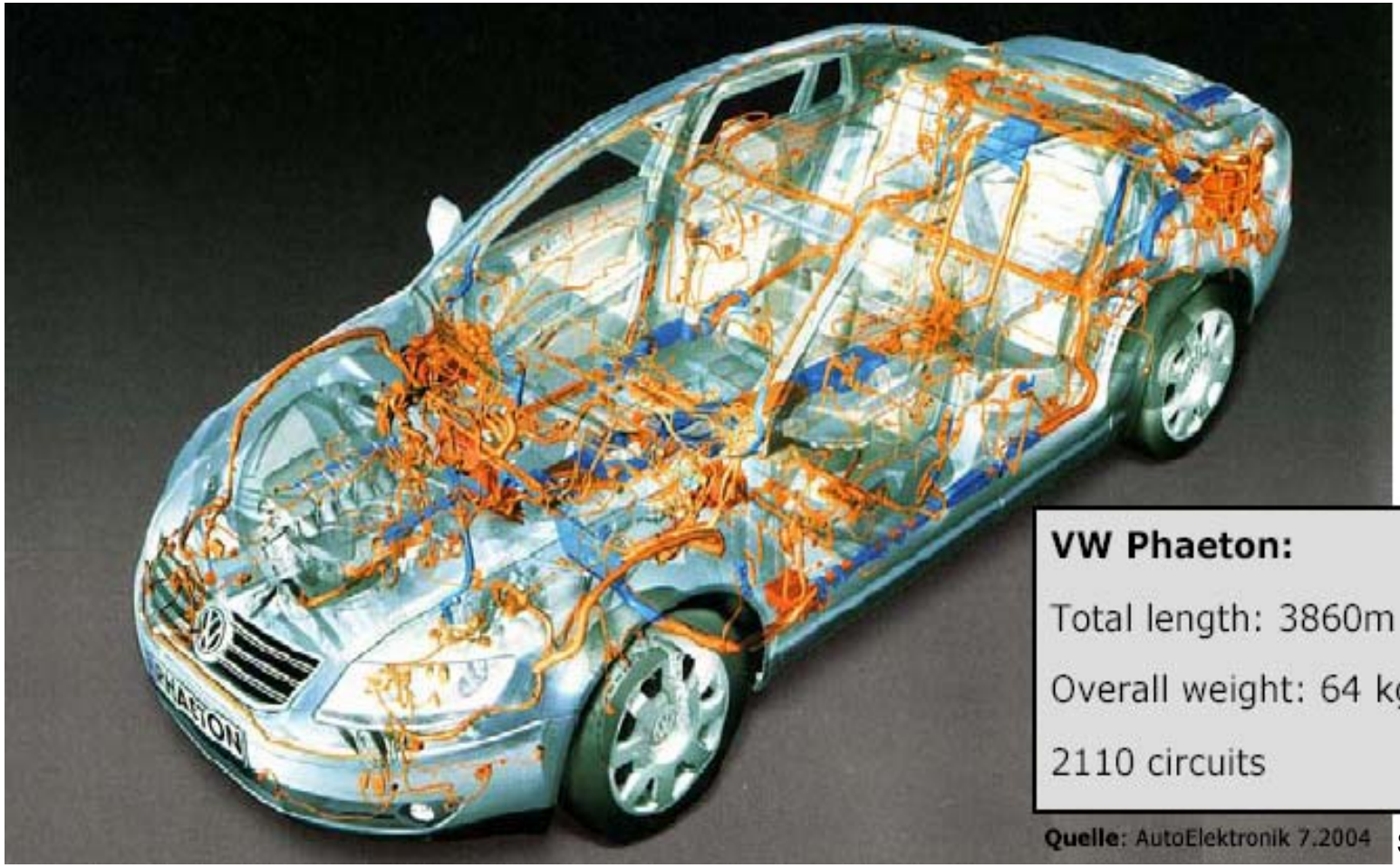


Wiring Harness Yesterday (VW Beetle 1950)



Source: Decomsys

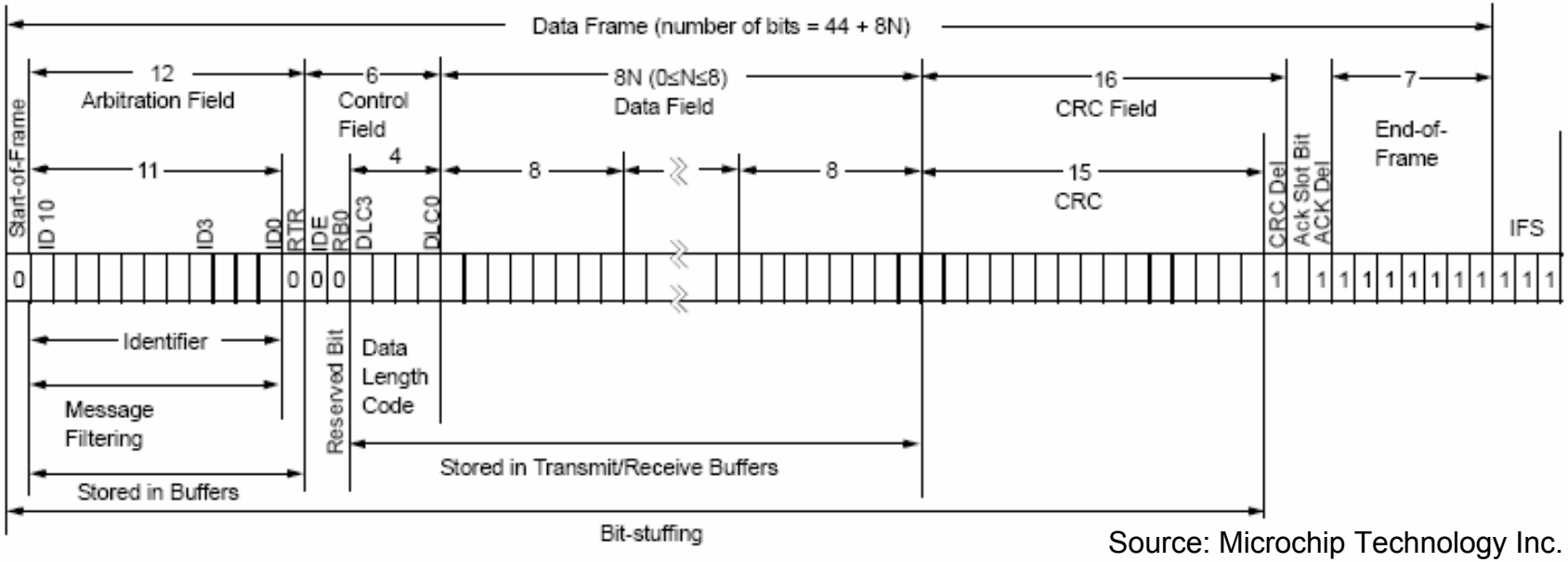
Wiring Harness Today (VW Phaeton)



Source: Decomsys

The CAN Message Format

- Every datagram within the CAN Bus consists of a number of bits, which are divided into different fields.



Source: Microchip Technology Inc.

SOF: 1 dominant bit (0) to indicate the beginning of a new message

Arbitration: 11 bit identifier of the source of the message (priority) + RTR

Control: 6 bits to identify the length of the data

Data: Data to be transmitted (payload)

CRC: Cyclic Redundancy Check (bit errors)

Bus Arbitration

- To avoid data collisions, CAN performs a bitwise and non-destructive arbitration on the bus

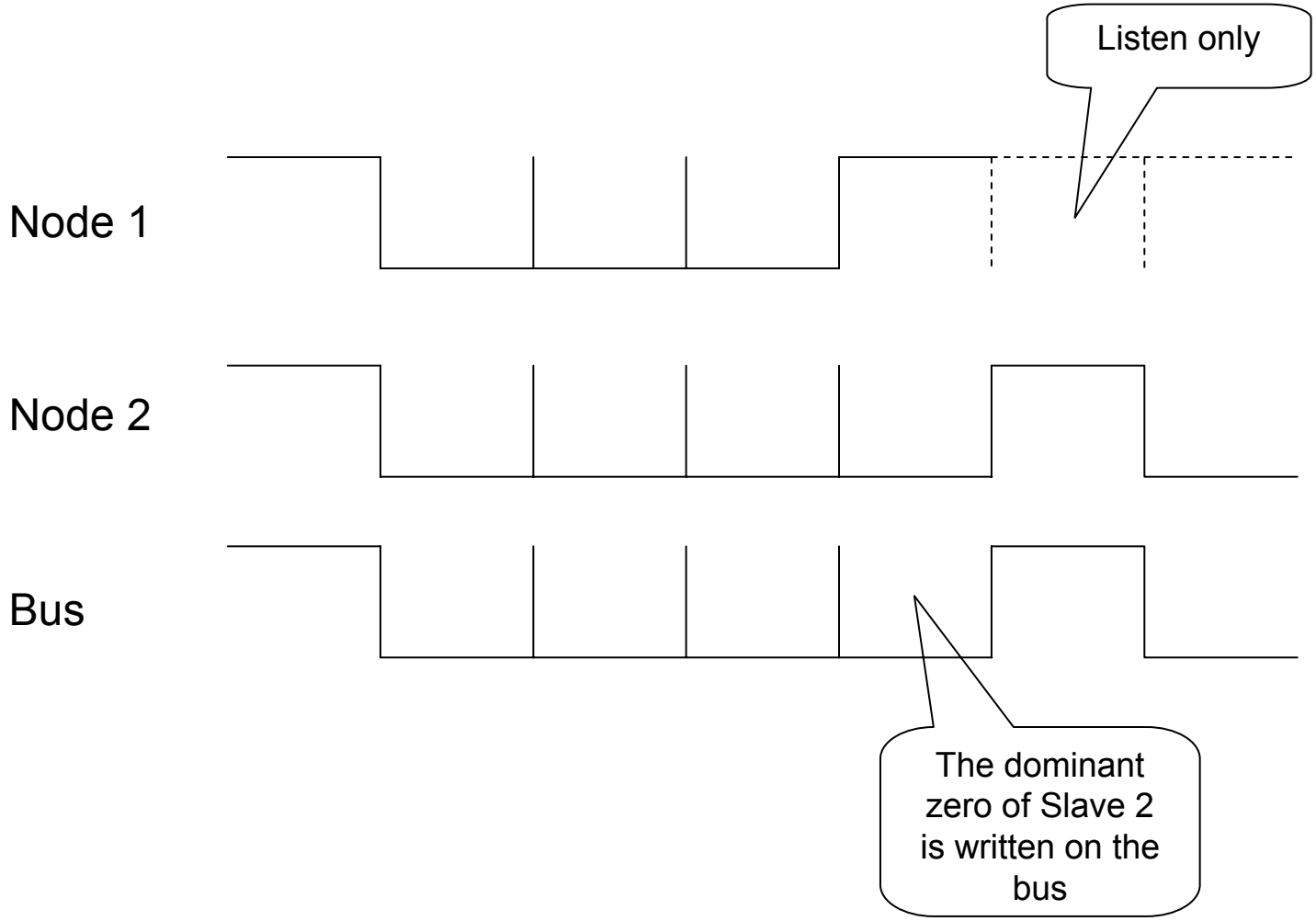
- Wired AND configuration:
  - 0-level: dominant level
  - 1-level: recessive level

- Whenever the bus is free (recessive level), any station can start to transmit data. => Multimaster functionality

- The lower the value of the identifier, the higher the priority of the frame

Bus Arbitration

2 Nodes writing to the bus:

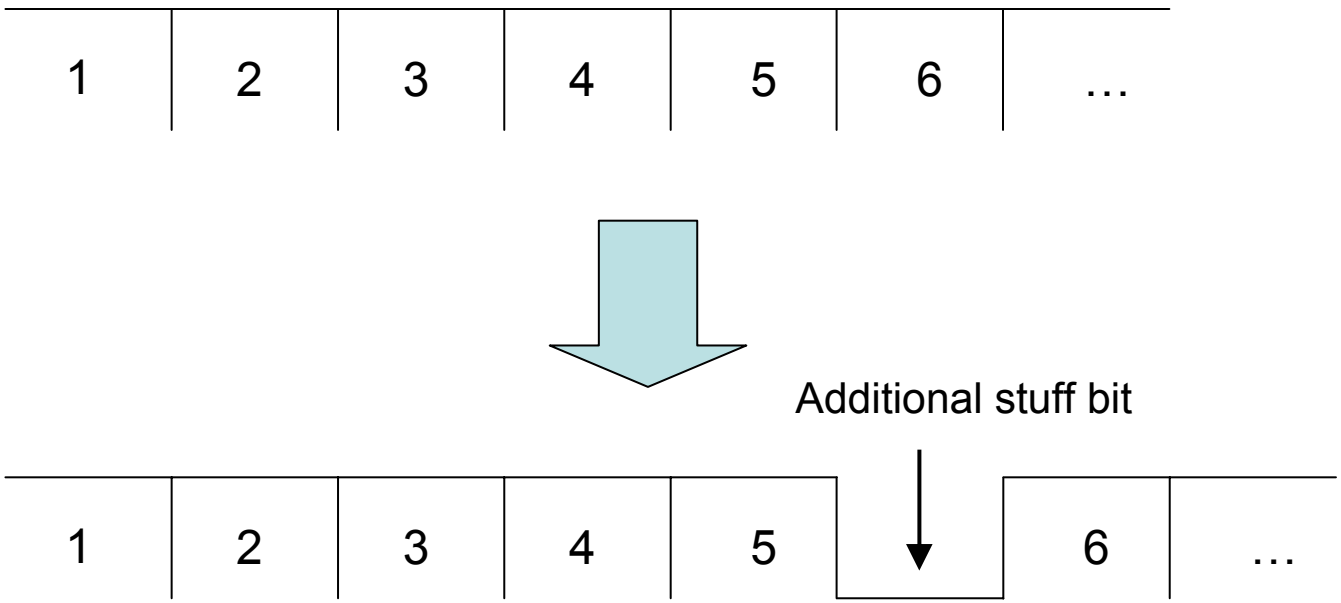


Bit Stuffing

- In order to enable synchronisation of the CAN Modules within the frame, there has to be some level-transitions of the data of the frame.

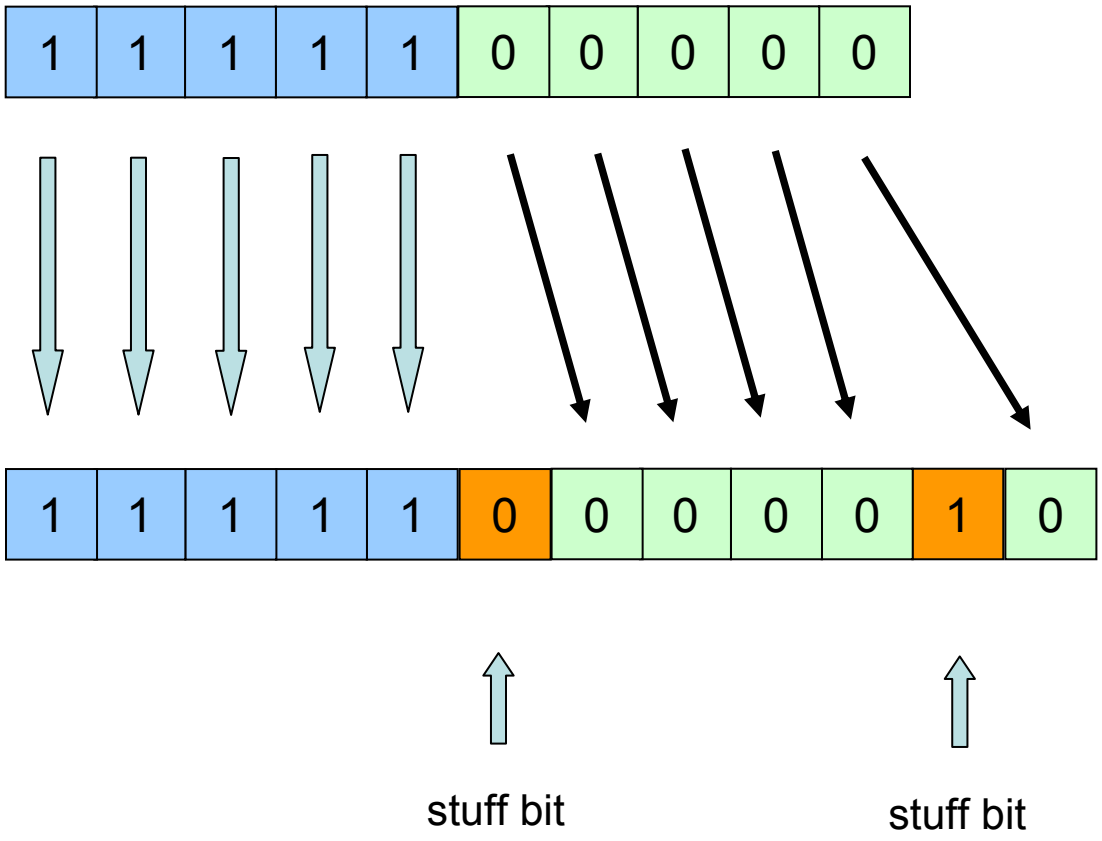
To ensure this, if it is naturally not the case, Bit Stuffing has to be applied

- Whenever 5 consecutive zeros or ones have been detected, an inverted stuff-bit has to be inserted into the bitstream

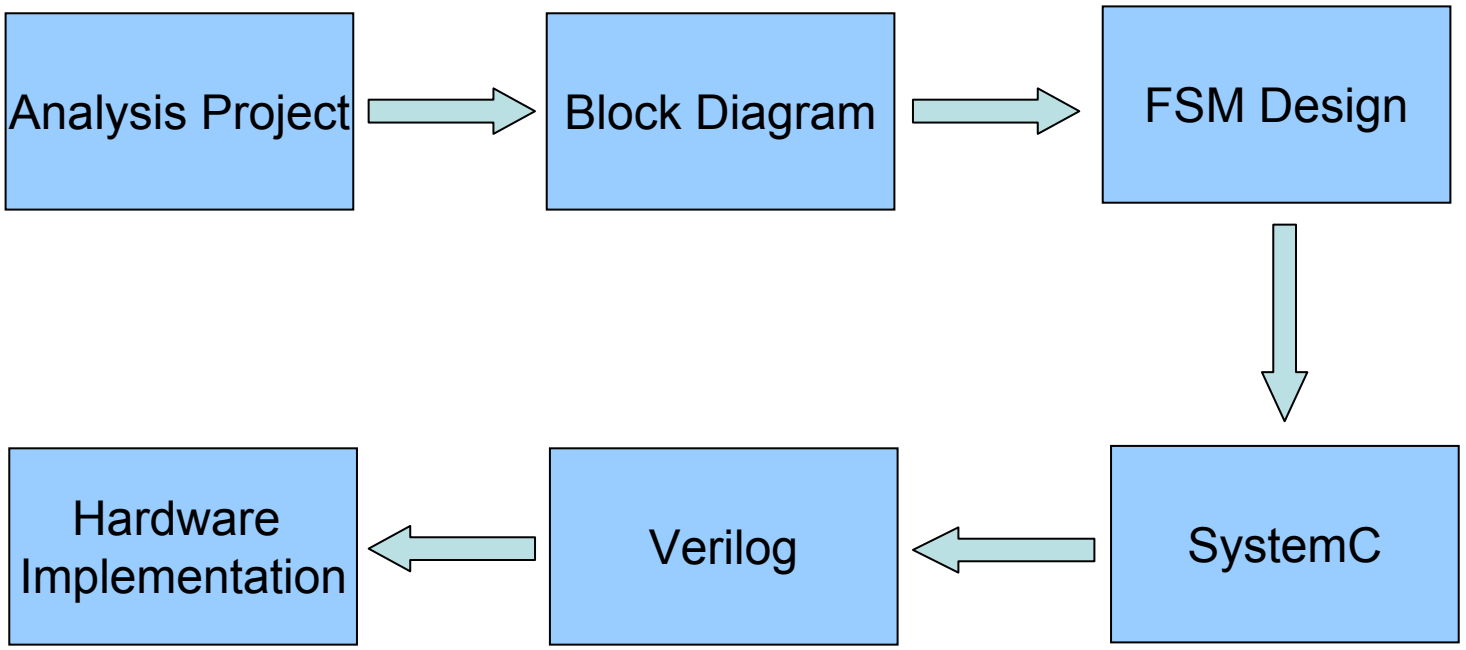


Special Example of Bit Stuffing

- **Transform the Bits stream 11111 00000 to the bus.**

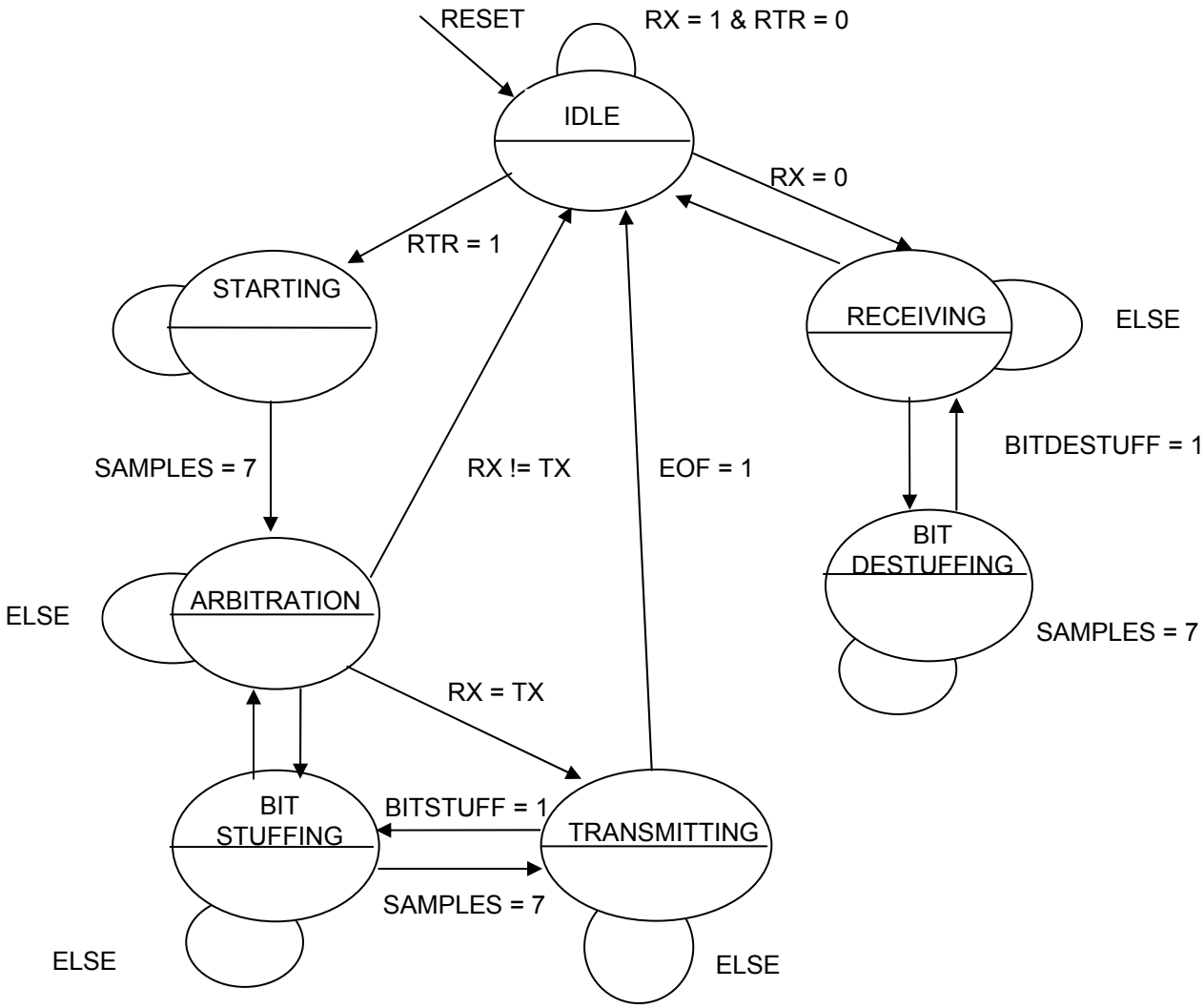


Development of the CAN Slave Module

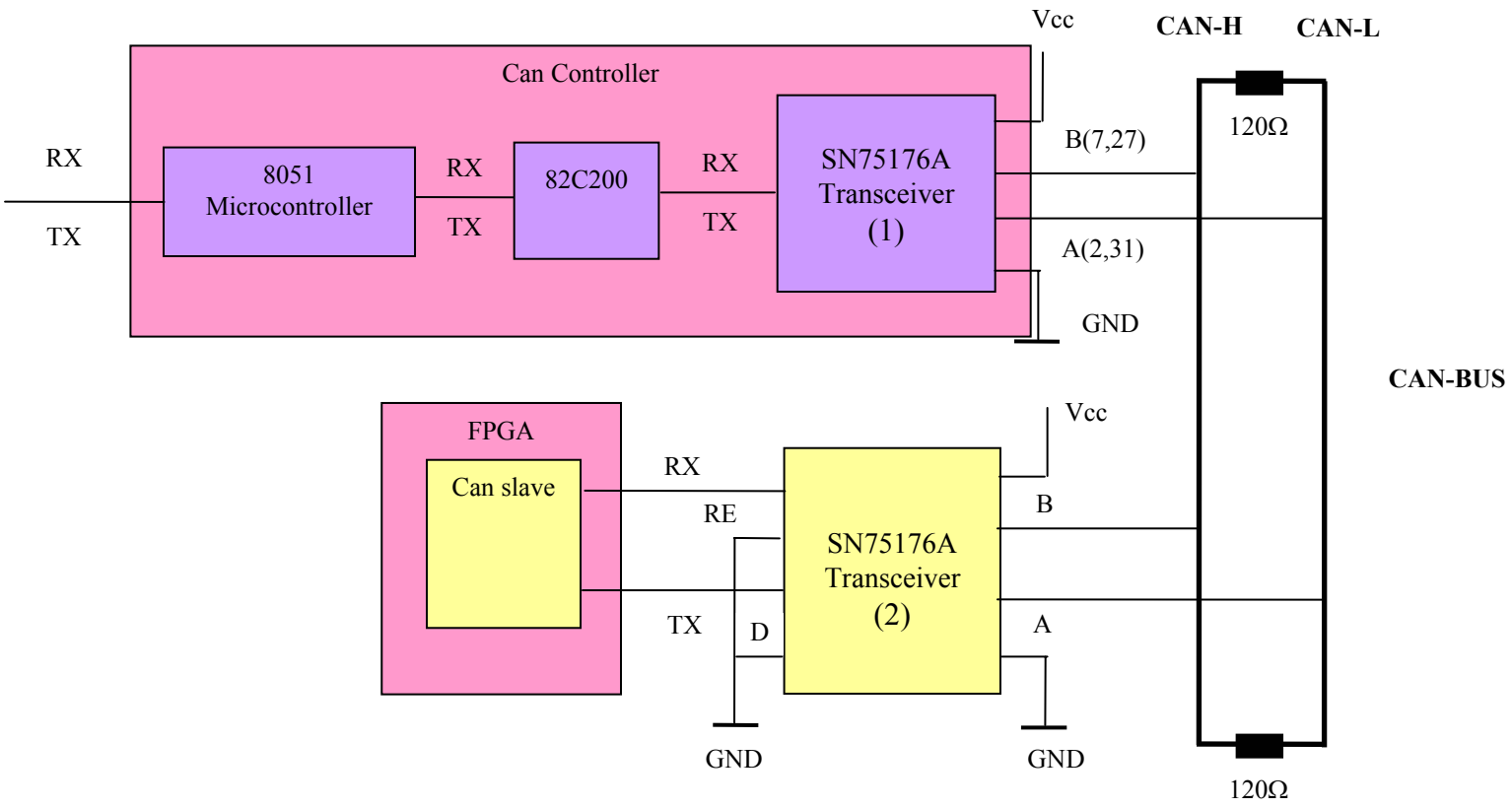


The Finite State Machine (FSM)

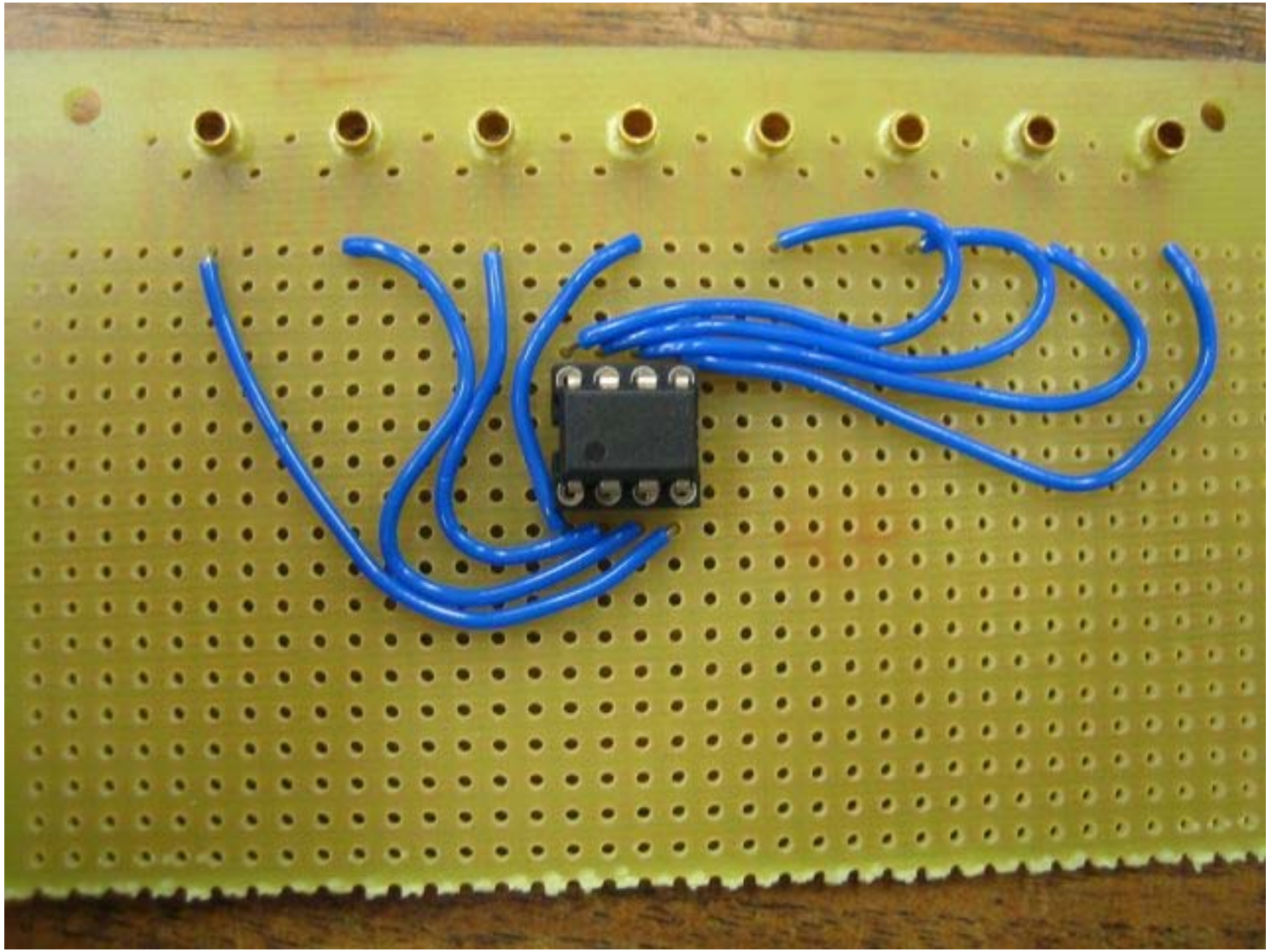
FSM:



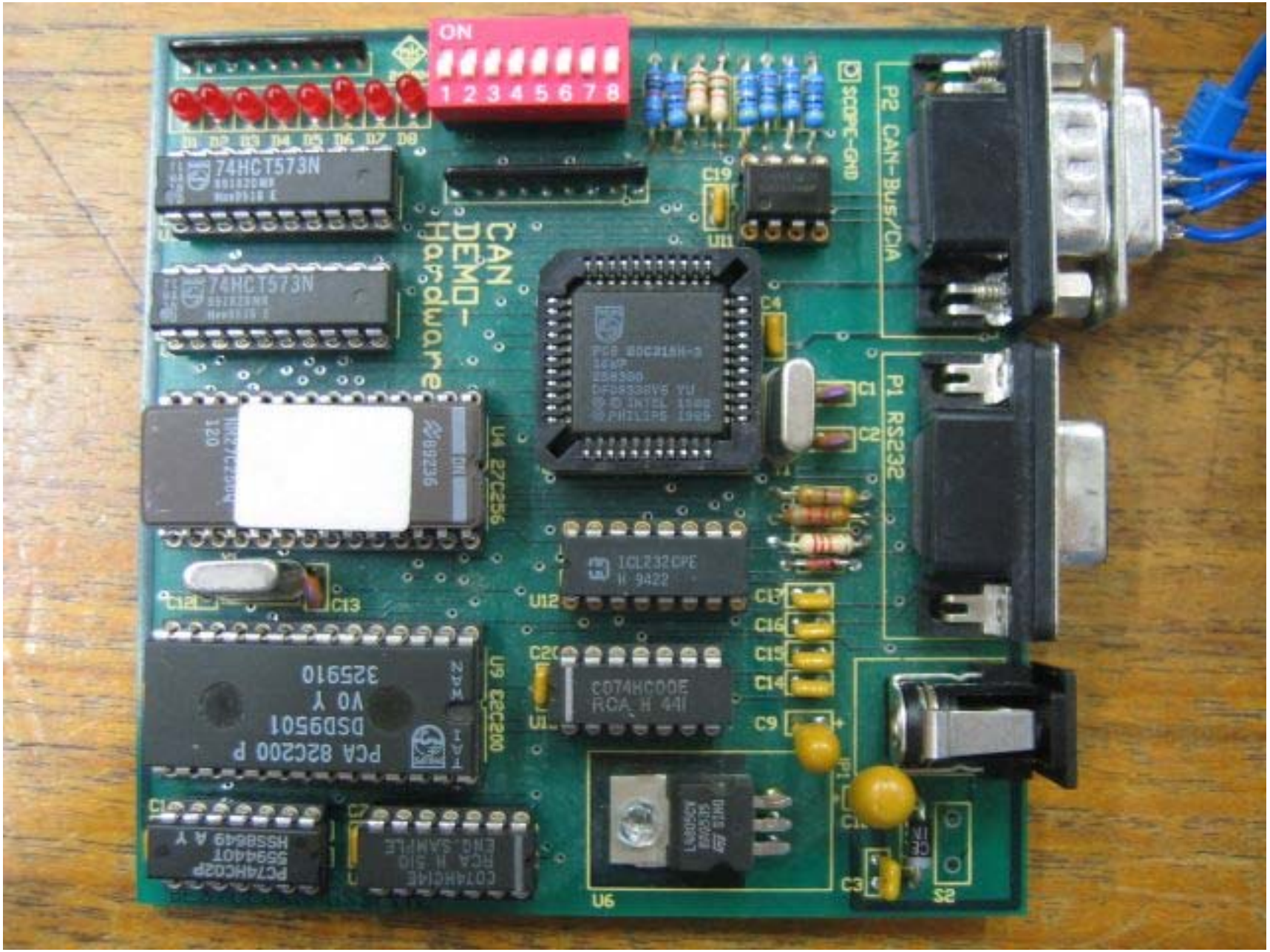
Architecture of the CAN Bus



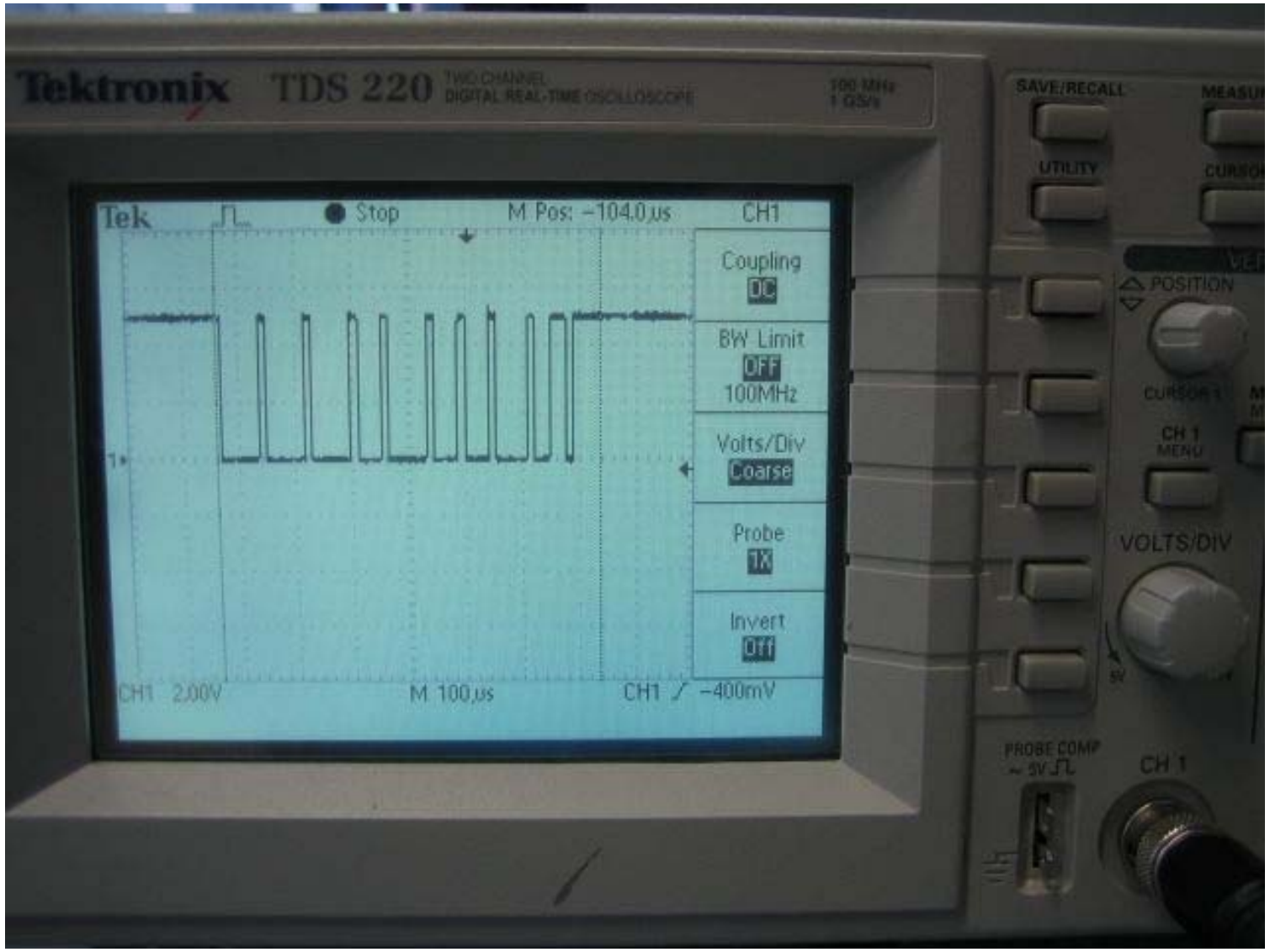
Interface between CAN Master and Salve



Hardware of the Candy (Can Master)



Stream from the CAN Master



Problems

- Bit (de-) stuffing: extraction and insertion of stuff-bits according to the CAN Protocol, including “special” stuff-bits

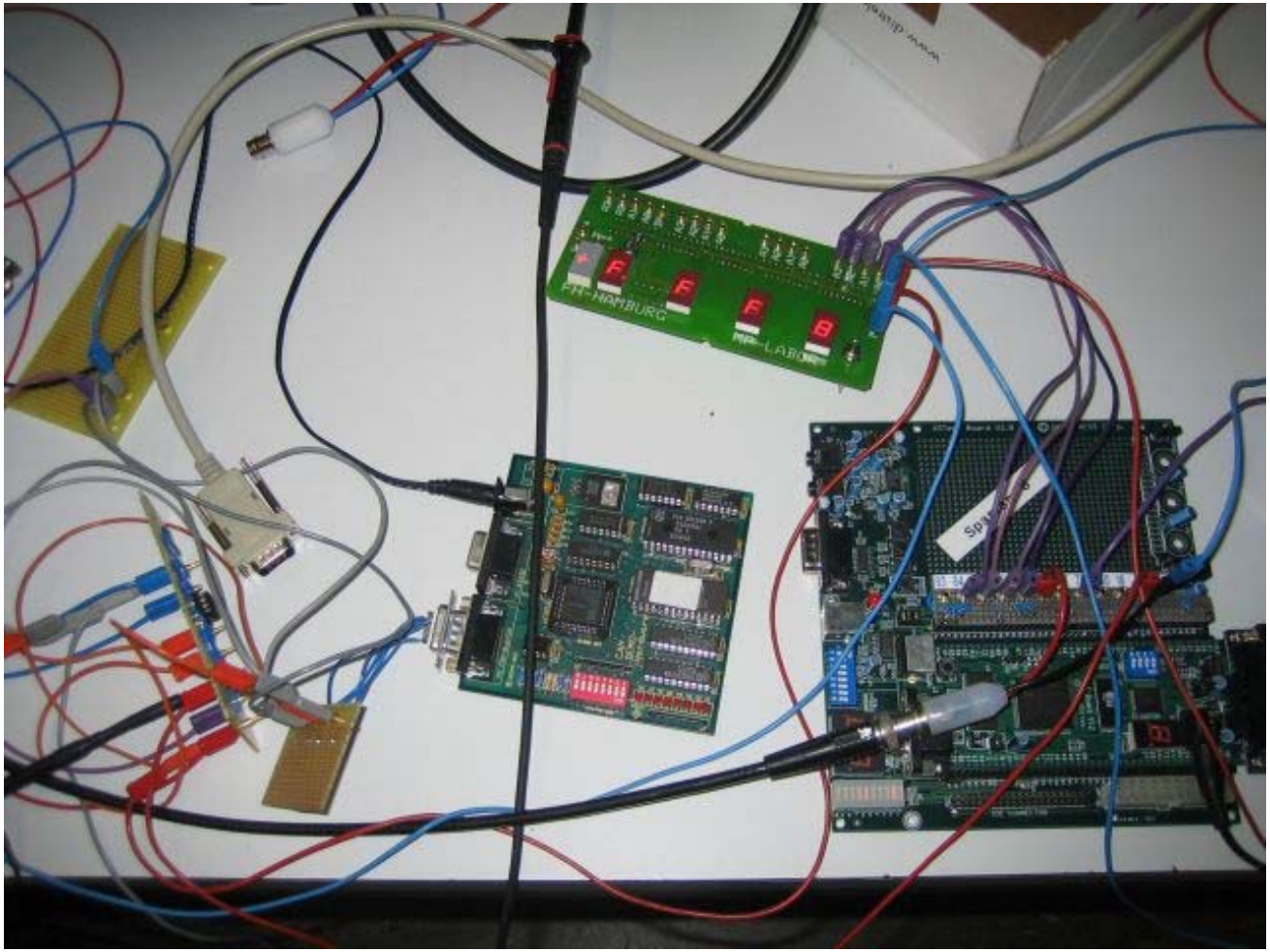
- Termination and connection of bus: Finding out the correct termination, wiring and connections of the bus

- Timing/synchronization: keep the different Slaves synchronous

- Clock: elaborate the correct frequencies

- SystemC -> Modelsim -> Xilinx
  = 3 different!!! Behaviors + cumbersome development steps (no IDE, but several different tools)

Presentation of Hardware



Conclusion

- CAN is a very mighty Bus, which uses its bandwidth efficiently and the access to the Bus is organized according to the priorities of the individual modules.

- The fieldbus has a guaranteed message latency (priorities through identifiers). Therefore it is very suitable for applications were real-time capabilities are needed.

- The successor of the CAN Bus will be Flexray, which has in addition Time- and Frequency-Mux capabilities. But CAN will coexist for at least the next one or two decades.

Questions

Many thanks to:
Prof. J. Reichard
Prof. B. Schwarz
Dipl.-Ing. D. Palme
J. Pflüger

Any questions left ?

References

- Decomsys Presentation on Flexray from 08.12.05 at HAW, obtained from Prof. B. Schwarz
- Stand-Alone CAN Controller With SPI Interface, Microchip Technology Inc.
- CAN Controller Area Network – Grundlagen und Praxis, Wolfhard Lawrenz
- Controller-Area-Network, Konrad Etschberger