Karlsruher Institut für Technologie

Institut für Technik der Informationsverarbeitung ITIV

Dr.-Ing. Jens Becker *(jens.becker@kit.edu)* ***Dipl.-Inform. Tanja Harbaum*** *(harbaum@kit.edu)*

Institut für Technik der Informationsverarbeitung (ITIV)

Communication Systems and Protocols

Exercise 5

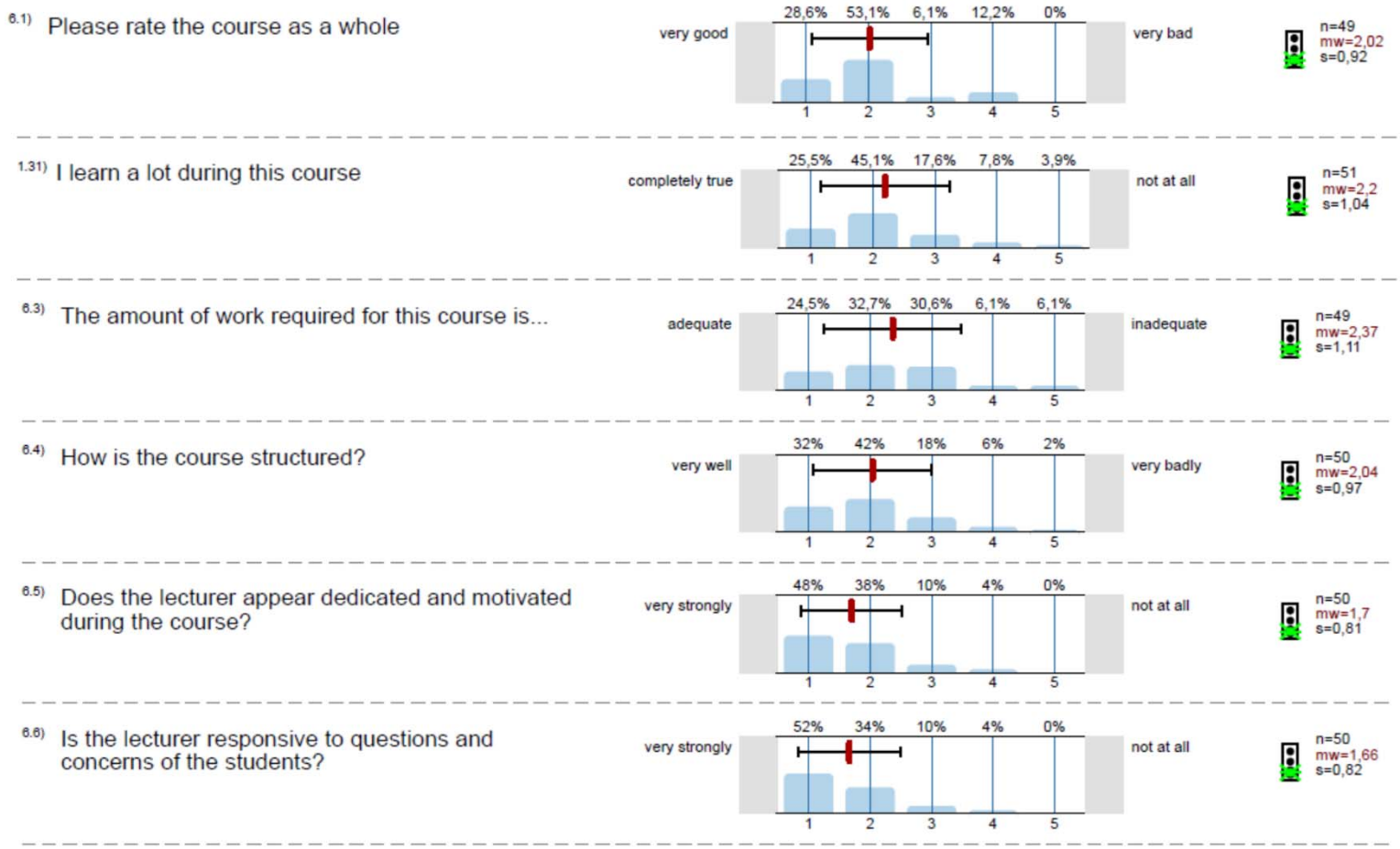
Schedule CSP

Monday: 09:45 – 11:15 (MTI)	Thursday: 09:45 – 11:15 (EAS)
24.04.2017 – Lecture 1	27.04.2017 – Lecture 2
01.05.2017 – May Day	04.05.2017 – Exercise 1
08.05.2017 – Lecture 3	11.05.2017 – Lecture 4
15.05.2017 – Exercise 2	
22.05.2017 – Lecture 5	25.05.2017 – Ascension Day
29.05.2017 – Lecture 6	01.06.2017 – No Lecture
05.06.2017 – Pentecost week	08.06.2017 – Pentecost week
12.06.2017 – Exercise 3	15.06.2017 – Corpus Christi
19.06.2017 – Lecture 8	22.06.2017 – Exercise 4
26.06.2017 – Lecture 9	
03.07.2017 – Lecture 10	06.07.2017 – Exercise 5
10.07.2017 – Lecture 11	13.07.2017 – Exercise 6
17.07.2017 – Lecture 12	20.07.2017 – Lecture 7
24.07.2017 – Exercise 7	27.07.2017 – Question time

Monday, 31.07.2017: CSP - EXAMINATION

Evaluation Results

- LQI: 100



Evaluation Comments

- Many different topics, but non of them is realy discussed in detail

Lecture „only“ gives an overview

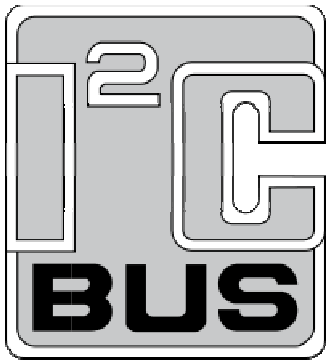
- Tasks in problem sheet are formulated unclear!
- Unclear formulation tasks/statements

I‘m sorry. We do our best!

- As a computer scientist the hardware realisations are very hard to understand

We try to give you a brief introduction

I²C-Bus (Inter-Integrated-Circuit-Bus)



- Serial Bus System
- Multi Master / Multi Slave
- CSMA/CA Arbitration Scheme
- Low Cost Bus System
- …

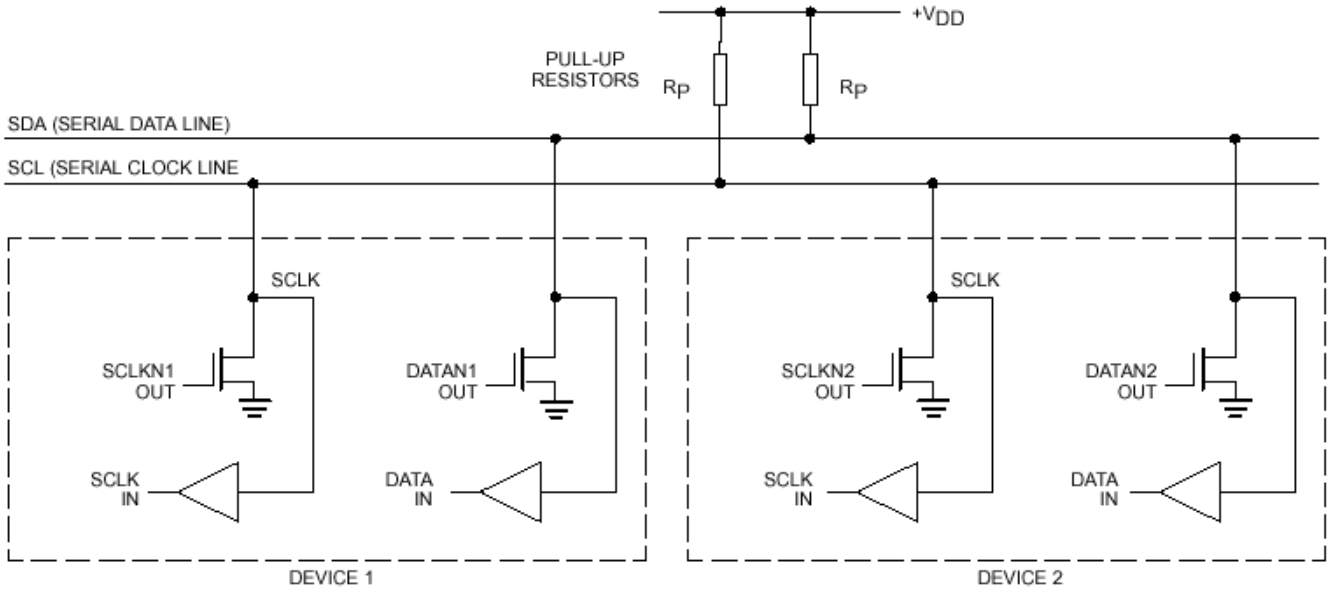
I²C-Bus

- Developed by Philips for interconnecting ICs on printed circuit boards
- Only two wires for data transmission
  - Clock line SCL (serial clock)
  - Data line SDA (serial data)
  - Lower cost and error-prone because of low pin-count
- Multi-Master, Multi-Slave
- Data transmission in packets of 8bit
  - 7bit for addresses → 128 addresses
  - 1bit for toggling between reading/writing
- Transfer rate:
  - 100kbit/s in standard mode
  - 400kbit/s in fast mode
  - 3.4Mbit/s in high-speed mode

Source of I²C images and diagrams: *NXP: UM10204, I2C-bus specification and user manual, Rev. 4 — 13 February 2012*

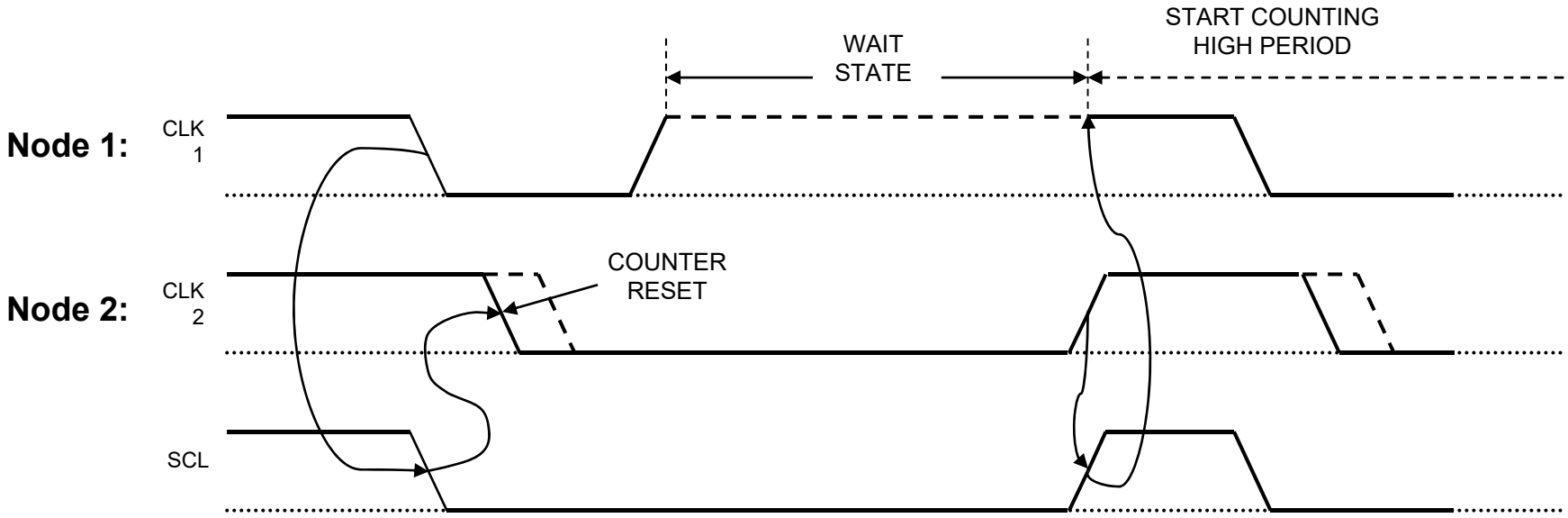
I²C Bus Connection

- Open-Collector outputs for every bus subscriber
  - Wired-AND with dominant ‚0‘ on the bus
  - Pull-Up resistors externally connected
- In idle mode, both SCL and SDA are HIGH
- Concurrent monitoring of the bus at every subscriber



Clock Synchronization

- Clock signal is the sum generated by all nodes
- A slower node can pull down the bus to ‚LOW' in order to insert wait states
- The next cycle is not started before SCL is back to ‚HIGH'



Task 1: I²C Synchronization

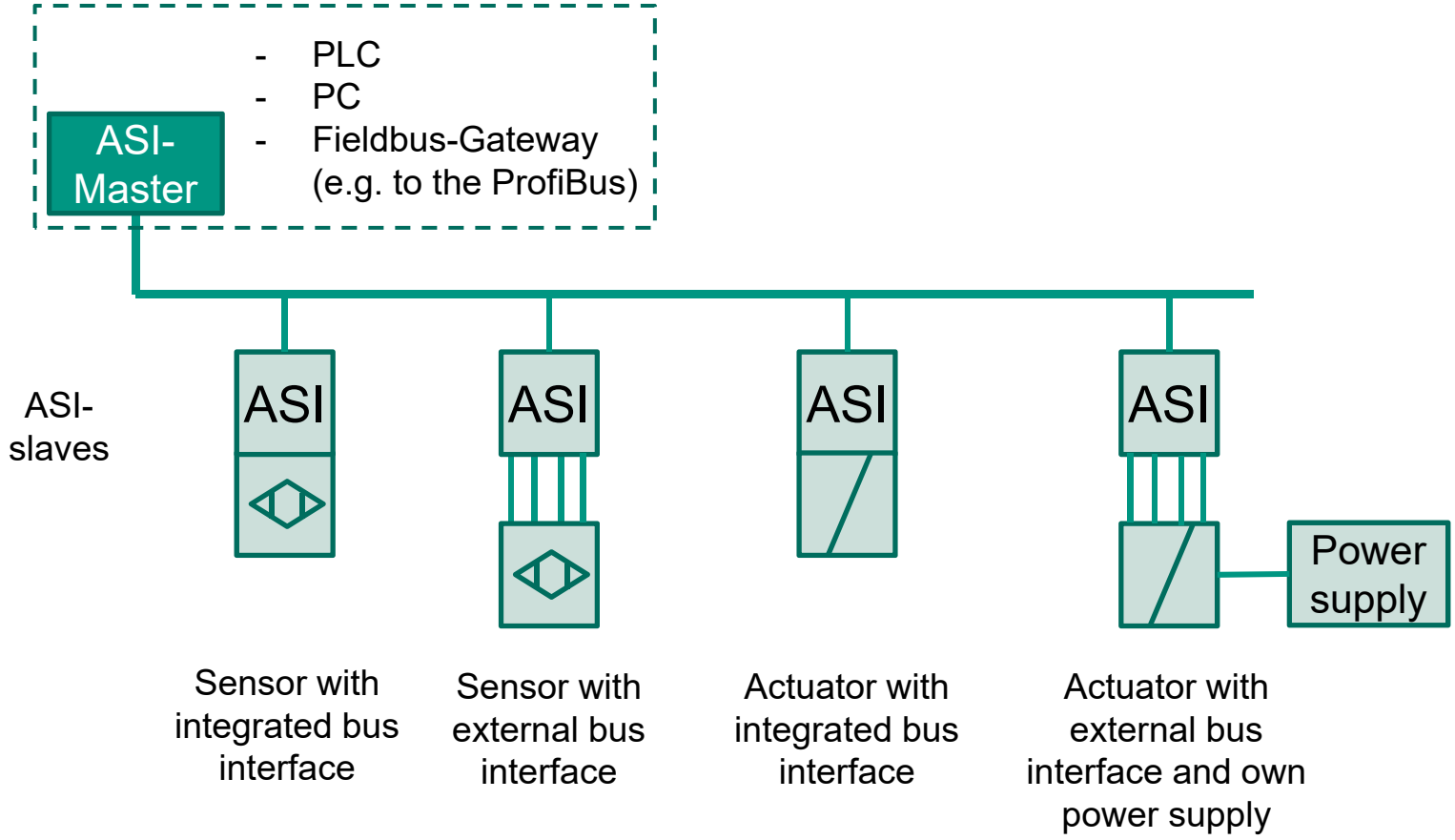
Time remaining ‘10

The Actuator Sensor Interface (ASI)

- The goal is to directly connect sensors/actuators to the control via a bus
- Requirements:
  - Easy installation
  - Easy commissioning and maintenance
  - Low-priced (because of many bus connections)
- Data and energy should be transmitted on a two-conductor cable for all sensors and most actuators
- Simple and robust transmission procedure without limitations concerning net topology
- Compact and inexpensive bus connection

Concept of the ASI

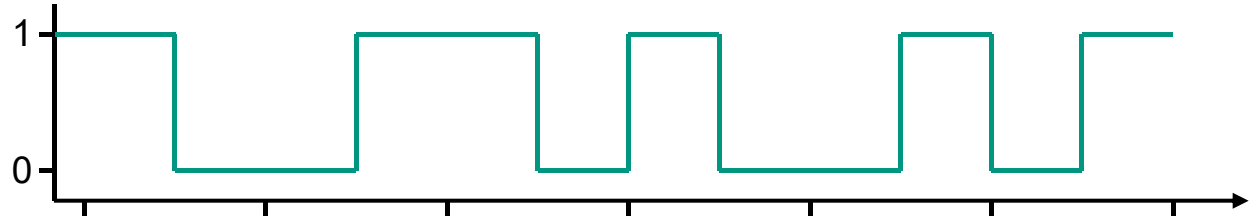
- Master slave concept using a single master



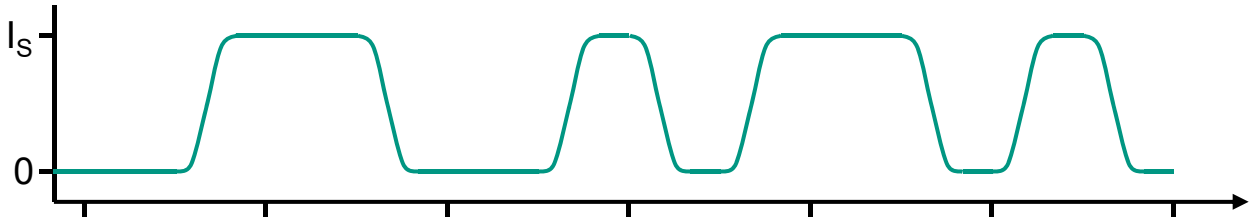
Examples for modulation

Sender bit string

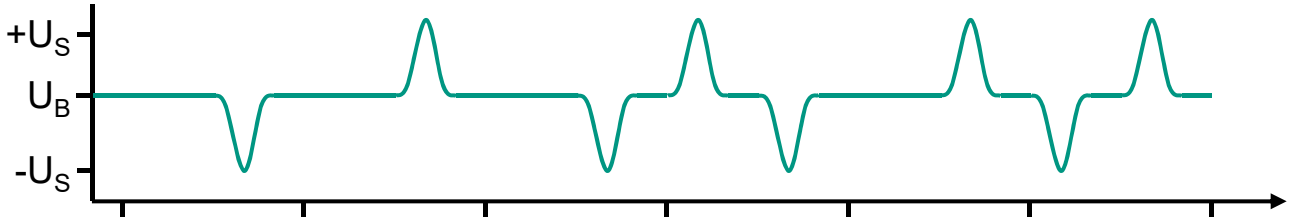
Manchester coded bit string



Sender current



Voltage on signal line



Task 2: ASI

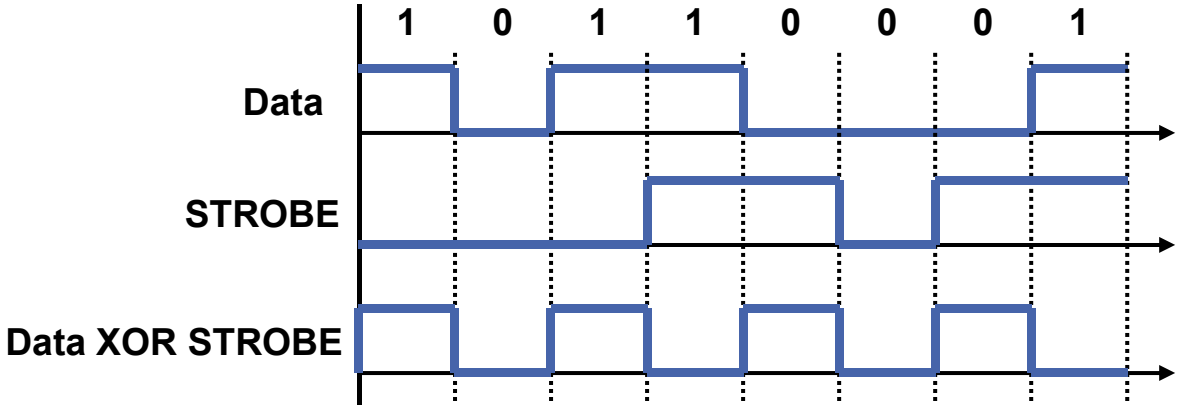
Time remaining ‘5

FireWire

- Multi Master System
- Self-Configuring
- Serial Bus System
- Hot-Plug Capability
- High Data rates
- Real-time capability
- …

Encoding

- Data is transmitted differential using two data lines
  - NRZ coding
  - No stuffing or other measures for clock recovery
- The STROBE signal is used to recover the clock signal
  - When equal data bits are to be transmitted, the STROBE signal changes its state
  - Recovery through XOR combination of both signals
  - Advantages
    - Insensitive for interference
    - Lower bandwidth required

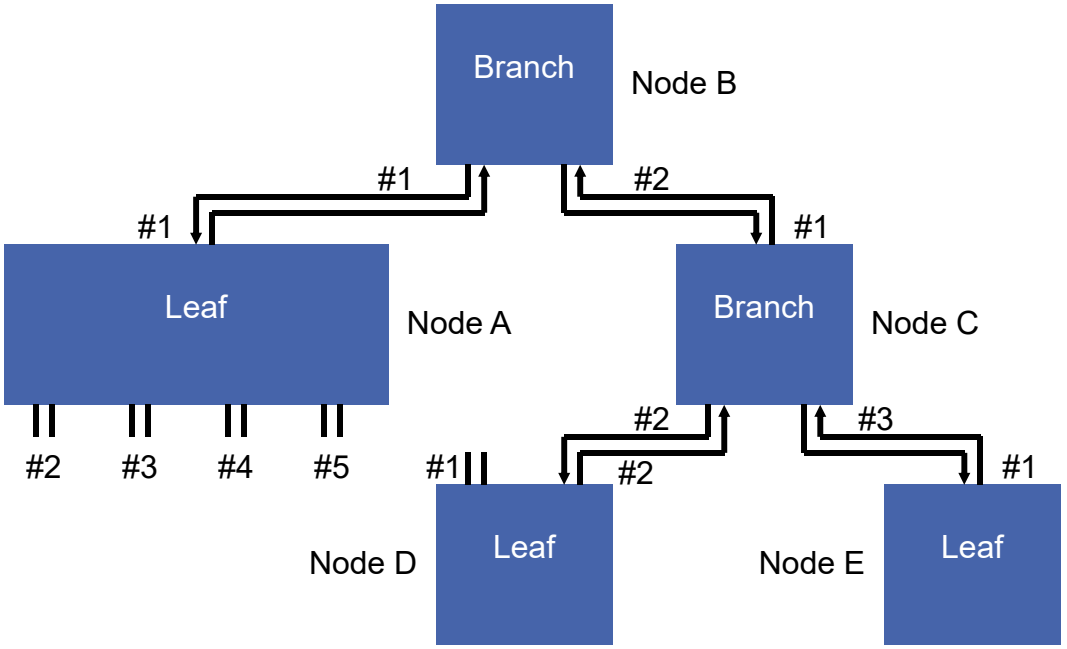


Configuration of the bus

- No user interaction for bus configuration required
- There is no dedicated controller, every bus member can become root of the tree
- Individual addresses and root of the bus is negotiated among the bus members
- Bus assignment is done in three steps
  - Initialization
  - Tree identification
  - Self identification

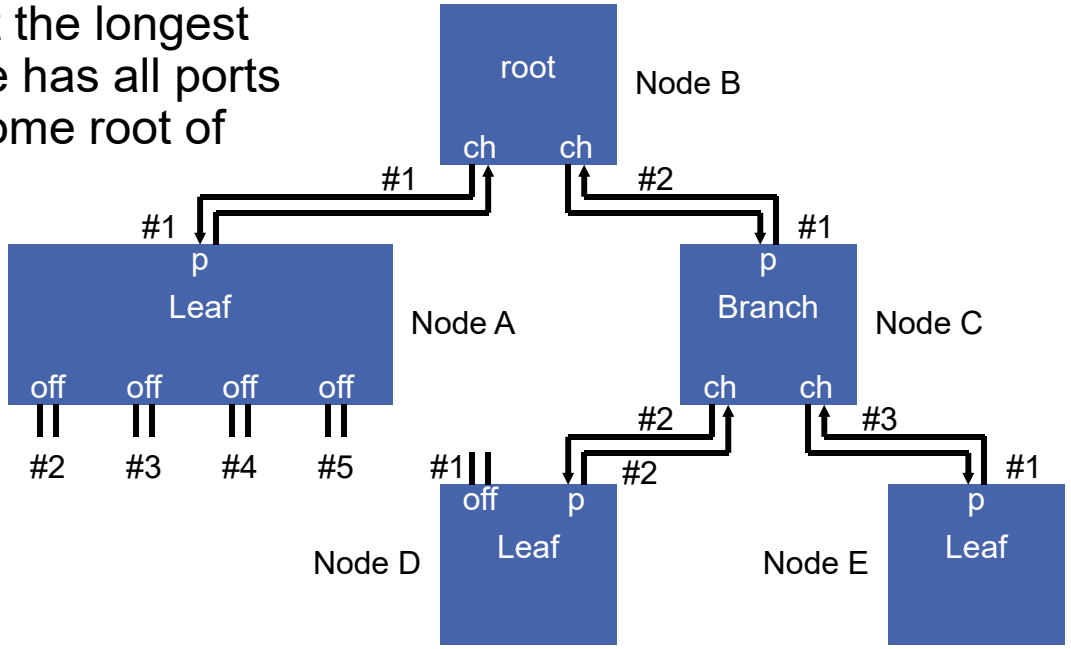
1. Initialization

- After a bus reset, every subscriber only knows the number of outgoing connections
  - One single connection → Leaf
  - Several connections → Branch
- If the bus configuration changes another reset is issued



2. Tree identification

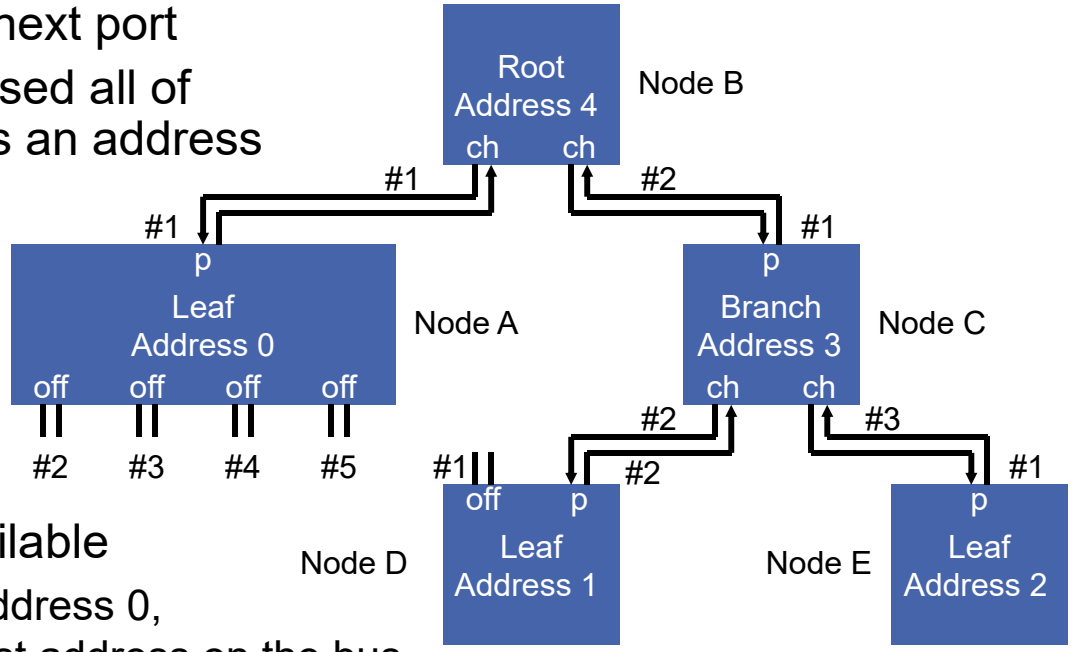
- Method:
  1. Every leaf sends a “parent-notify” packet on the bus and marks the corresponding port as “p” (parent)
  2. The node that receives such a “parent-notify” packet will mark the corresponding port as “ch” (child)
  3. If there is only one unmarked port left in a node, the node will send a “parent-notify” using the unmarked port and marks this port as parent
  4. The node that has to wait the longest time, that in consequence has all ports marked as child, will become root of the tree



- Waiting time and number of hops determine who will become root of the tree
- Also a leaf can become root of the tree if it is waiting long enough

3. Self identification

- Assigning the node address
  1. Root transfers control to the node that is connected to the first port of the root
  2. If this node has children on its own, he will forward control recursively to the first child node
  3. When a leaf is reached, it will claim the next address that is available and sends a packet with its address and additional status information to the bus.
  4. After that, control is returned to the parent node that processes the next port
  5. When a node has processed all of its children, it self-assigns an address and transfers control to its parent node



- Every node counts how many devices have already send their address and then chooses the next address available
  - Thus the first leaf gets the address 0, while the root gets the highest address on the bus

Task 3: Fire Wire

Time remaining