Karlsruhe Institute of Technology

Institut fuer Technik der Informationsverarbeitung

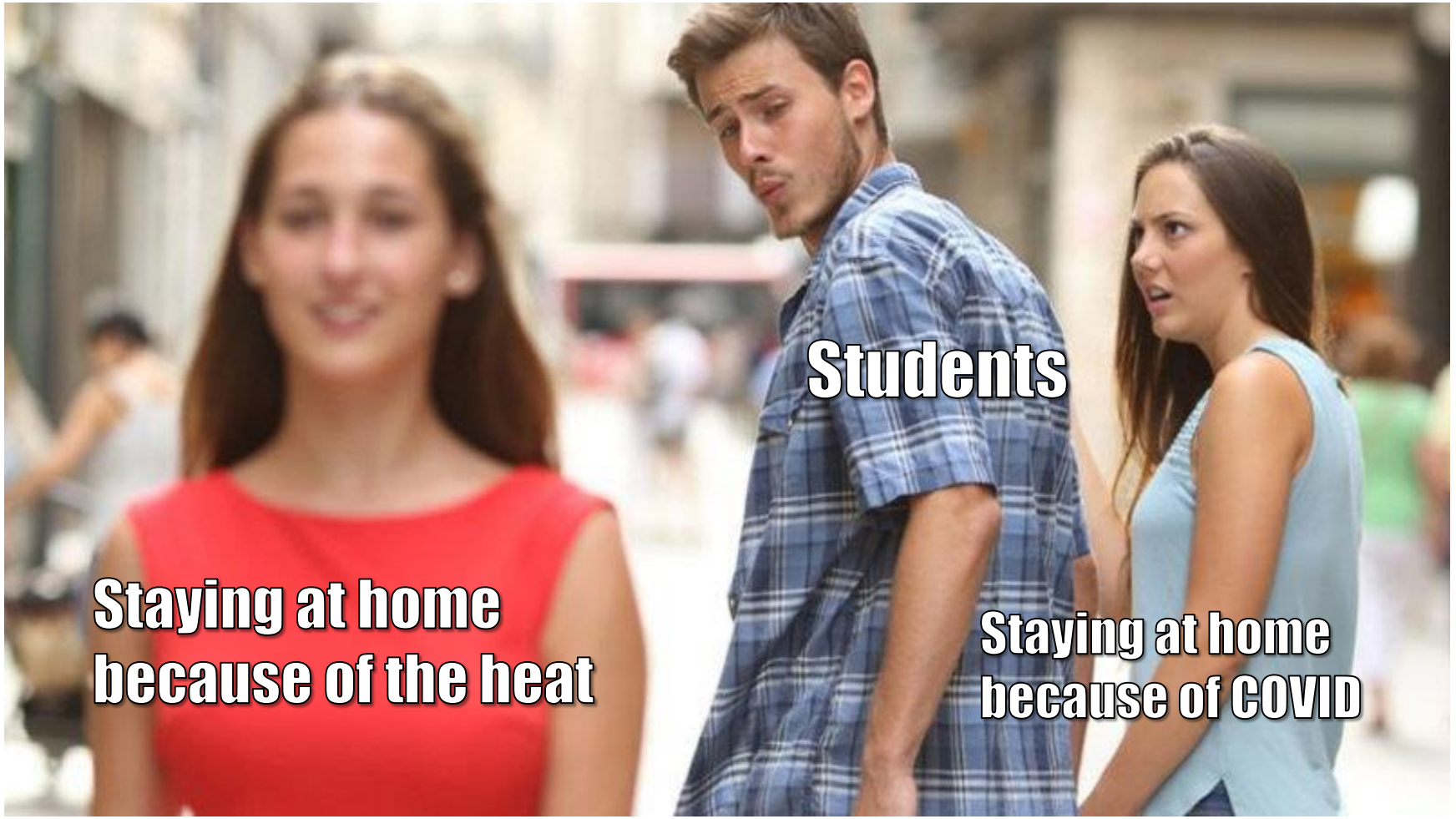
Dr.-Ing. Jens Becker *(jens.becker@kit.edu)* **M. Sc. Matthias Stammler** *(stammler@kit.edu)*

Communication Systems and Protocols

Exercise 4

General Information
Communication Systems and Protocols (Exercises)

Matthias Stammler, M. Sc.
Institut fuer Technik der Informationsverarbeitung (ITIV)
Building 30.10, Room 218
Email: stammler@kit.edu, csp@itiv.kit.edu
Website: https://www.itiv.kit.edu/21_7686.php


Students
Staying at home
because of the heat
Staying at home
because of COVID

Schedule CSP

Monday: 09:45 – 11:15 (NTI)

- 18.04.2022 – Easter Monday
- 25.04.2022 – Lecture 2
- 02.05.2022 – Exercise 1
- 09.05.2022 –
- 16.05.2022 – Exercise 2
- 23.05.2022 – Lecture 7
- 30.05.2022 – Exercise 3
- 06.06.2022 – Pentecost week
- 13.06.2022 – Lecture 9
- 20.06.2022 – Exercise 4
- 27.06.2022 – Lecture 11
- 04.07.2022 – Exercise 5
- 11.07.2022 – Exercise 6
- 18.07.2022 – Exercise 7
- 25.07.2022 –

Thursday: 09:45 – 11:15 (NTI)

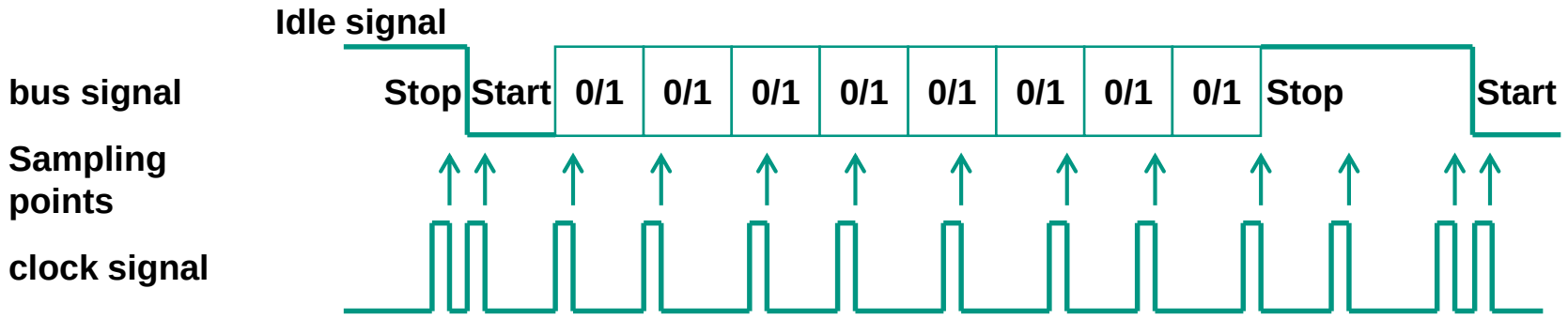
- 21.04.2022 – Lecture 1
- 28.04.2022 – Lecture 3
- 05.05.2022 – Lecture 4
- 12.05.2022 – Lecture 5
- 19.05.2022 – Lecture 6
- 26.05.2022 – Ascension Day
- 02.06.2022 – Lecture 8
- 09.06.2022 – Pentecost week
- 16.06.2022 – Corpus Christi
- 23.06.2022 – Lecture 10
- 30.06.2022 – Lecture 12
- 07.07.2022 – Lecture 13
- 14.07.2022 – Lecture 14
- 21.07.2022 – Question time
- 28.07.2022 –

Synchronisation Schemes

	Synchronous Transmission	Asynchronous Transmission
Parallel Transmission	Shared clock line	Flow-Control
Serial Transmission	Suitable line code or scrambler (shared clock line)	Start-Stop-mode

Start-Stop mode

- When no transmission occurs, bus is in idle state → no signal changes occur on the bus
- Begin of a transmission is signaled using a well defined edge on the bus signal → start bit
- This edge is used to synchronize sender and receiver clock
- Every bit has a well defined step size and thus can be separated using the internal clocks → baud rate
- At the end, it has to be ensured that the bus goes to idle level in order to be able to detect a new transmission → stop bit

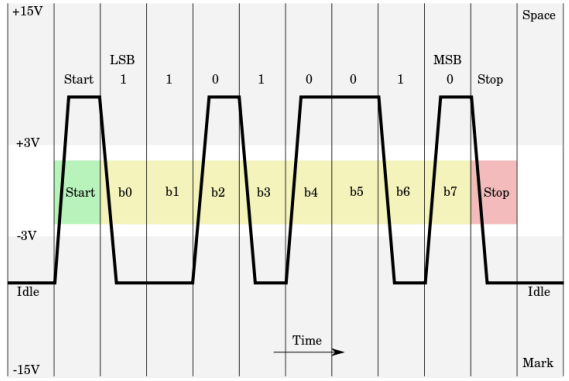


Start-Stop mode: Format definition

- Format definition for serial transmission using start-stop mode
  - X-Y-Z
- With:
  - X: number of data bits (normaly 5-8)
  - Y: parity check (none, even, odd)
  - Z: number of stop bits (1, 2)
- Notes:
  - There is always on start bit (denotes begin of transmission)
  - The number of stop bits in the definition is the minimum number. There can be more „stop bits“ if no transmission is going on

Recommended Standard 232 RS-232

- Standard for serial communication transmission of data
  - used for connections to modems, printers, data storage, …
- RS-232 was first introduced in 1962
  - by the Radio Sector of the EIA (Electronic Industries Alliance)
- USB has displaced RS-232 from most of its peripheral interface roles
  - still used in areas of micro controllers and programming



Source: https://de.wikipedia.org/wiki/RS-232

Task 1: Serial Interface

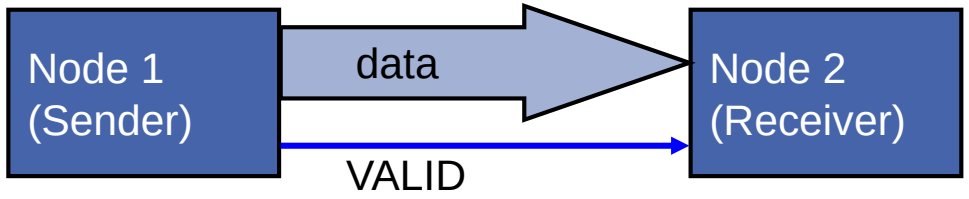
Time Allocated 10 min

Synchronisation Schemes

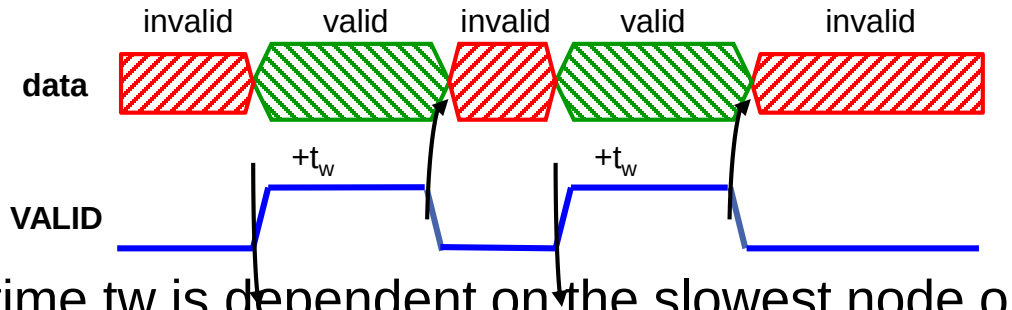
	Synchronous Transmission	Asynchronous Transmission
Parallel Transmission	Shared clock line	Flow-Control
Serial Transmission	Suitable line code or scrambler (shared clock line)	Start-Stop-mode

Open-loop Flow Control

If data is transmitted in multiple subsequent cycles, it has to be regulated when data of one cycle is processed and when a new cycle may be started.



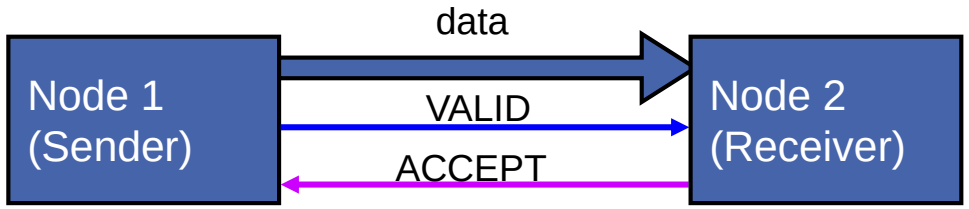
- One always waits for a fixed period of time tw



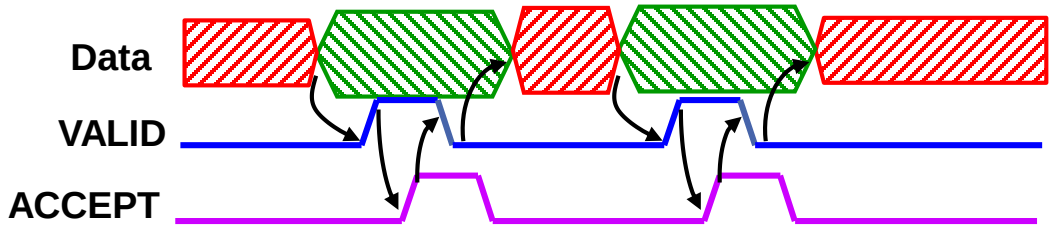
  - Disadvantage: Waiting time tw is dependent on the slowest node on the bus and is always invariant in length (synchronous mode)

Level-triggered Closed-loop Flow Control

Compared to open-loop flow control, closed-loop flow control uses additional signals from receiver to signal the sender its state.



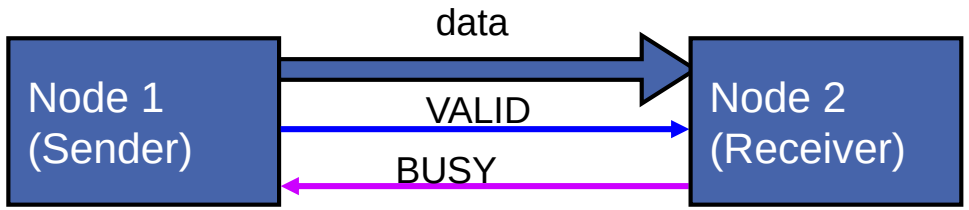
- Receiver asserts ACCEPT signal if data is not needed any longer.



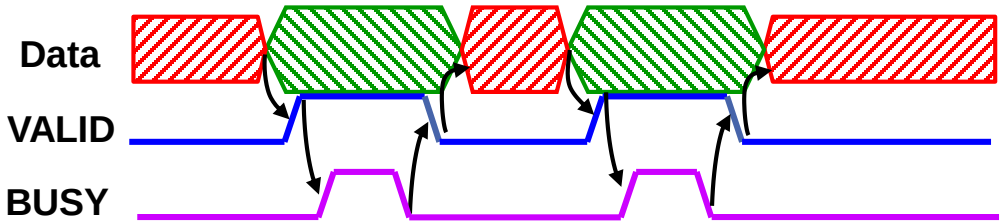
- Disadvantage: Problems with node synchronization if ACCEPT signal is asserted for too long (sender will remove subsequent data from the bus before receiver has read it)

Level-triggered Closed-loop Flow Control II

If data is transmitted in multiple subsequent cycles, it has to be regulated when data of one cycle is processed and when a new cycle may be started.



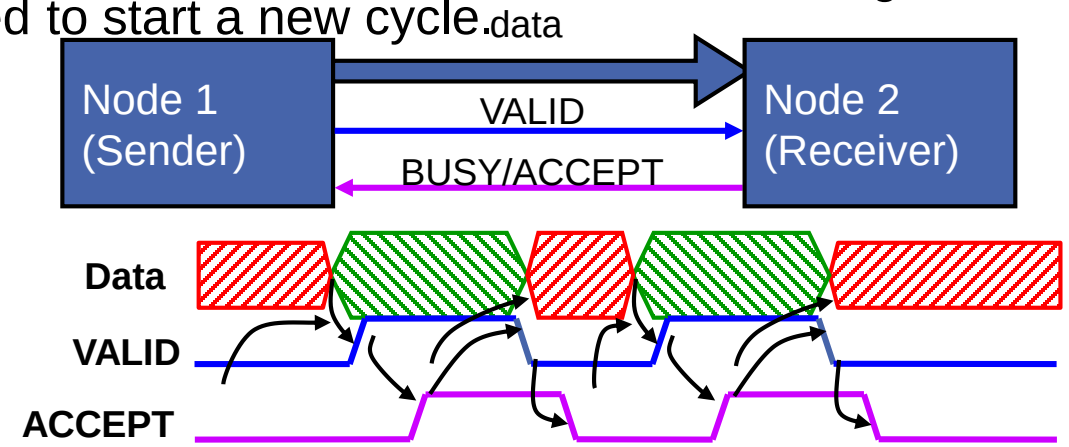
- Receiver asserts BUSY-Signal as long as data has to be available.



  - Disadvantage: Problems with node synchronization if BUSY signal is asserted too late (data will be removed from the bus too soon)

Edge-triggered Closed-loop Flow Control

- Safe synchronization by regarding all edges of the control signals
  - Sender is only allowed to put data on the bus if ACCEPT=0. Valid data on the bus is signaled with VALID=1.
  - Via VALID=1 the receiver recognizes new valid data and processes it. As soon as the data has been consumed, the receiver asserts ACCEPT=1.
  - Only if the receiver has acknowledged the data (ACCEPT=1), the sender is allowed to remove the data from the bus and sets VALID=0.
  - When the receiver detects the de-asserted VALID, it sets the signal ACCEPT=0 as well. Only after this the sender is allowed to start a new cycle.



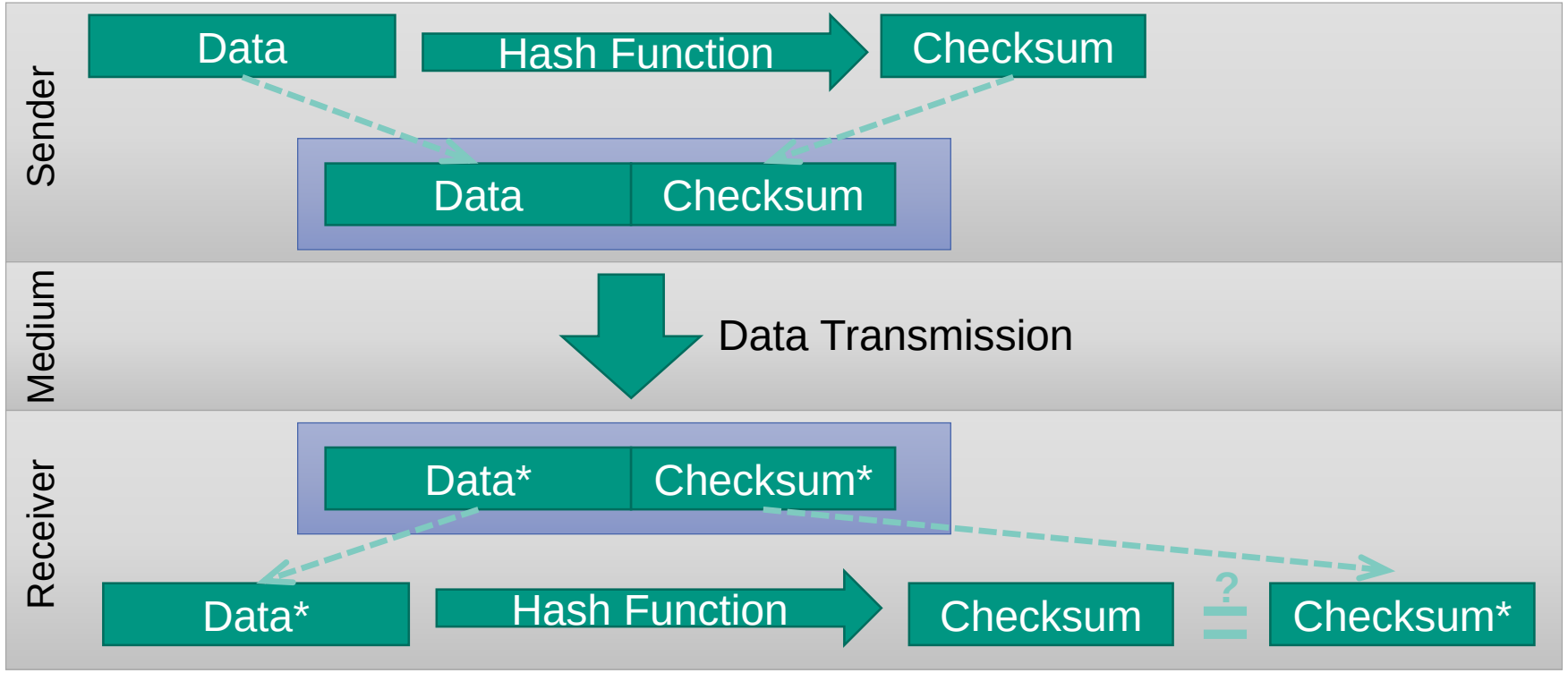
- Always safe independent of the sender's or receiver's speed.

Task 2: Flow-Control

Time Allocated 5 min

Cyclic Redundancy Check (CRC) – Approach

- Generate a checksum (Hash)
- Send checksum together with raw data
- Receiver checks received data and received checksum match



CRC – mathematical description

- Generation of checksum uses a generator polynomial G(x) on sender and receiver side
  - most significant bit (MSB) and least significant bit (LSB) must be ‚1‘
- Calculation on sender side
  - Data vector M(x) of length m bit (exceeding length of G(x))
  - Checksum equals the modulo of the division $(x^rM(x))/G(x)$
    - r: degree of generator polynomial
    - $x^rM(x)$ adds r zero bits to the end of the data vector
  - Checksum is appended to the data
    - Corresponds to the addition of the modulo: $x^rM(x) + R$
- Calculation on receiver side
  - Division of received data by G(x)
    - If the modulo of the division is not equaling zero an error has occurred
    - If the modulo of the division equals zero, no error has been detected

CRC – Hardware Realization

- Can be easily and efficiently realized in hardware.
- Polynomials are expressed as bit strings
  - e.g.: $M(x)=x^5+x^4+x^0$ corresponds to 110001

- Calculations based on division modulo-2 arithmetic, no carries
  - Corresponds to a bitwise XOR operation (⊕)

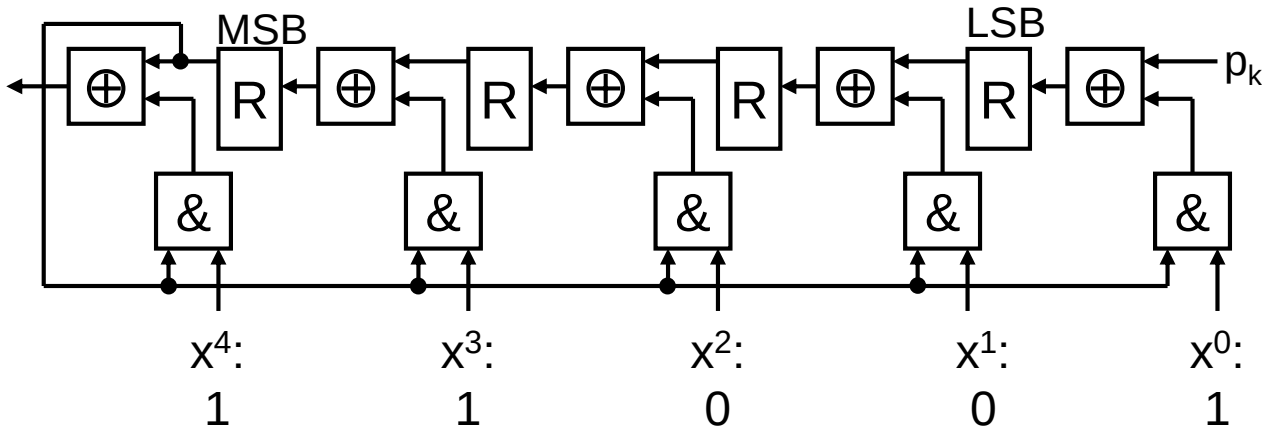
⊕	0	1
0	0	1
1	1	0

- Example for bitwise XOR operation:

```
 10011011
⊕11001010
 01010001
```

Hardware Implementation

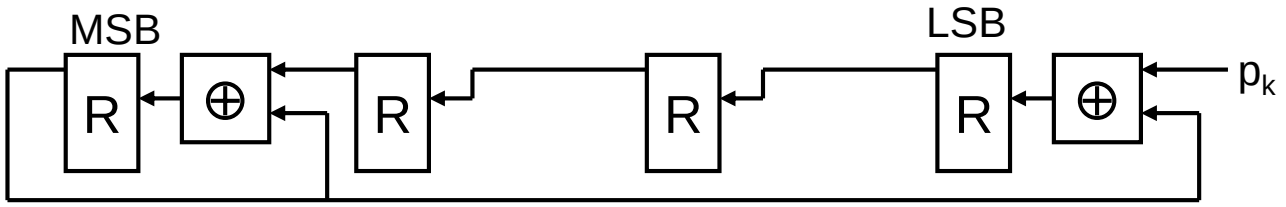
- Utilization of linear feedback registers with XOR-operations
  - Example: Generator polynomial $\mathbf{x^4+x^3+1 = 11001}$



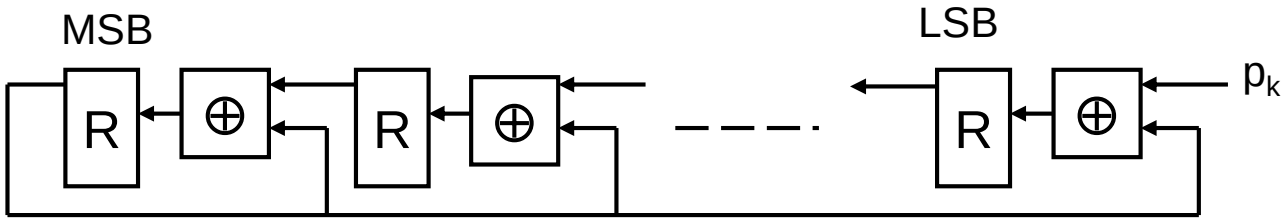
- Data vector is shifted bitwise through the register
- Content of MSB denotes the current quotient bit
- If MSB=1, the generator polynomial is added (XOR gate) and the dividend is shifted left (corresponds to a right-shift of the generator)
- If MSB=0, only the shift is executed
- Input data is shifted until a logic 1 is available in the MSB

Hardware Implementation

- Simplification for fixed generator polynomial:
- Example: Generator polynomial **$x^4+x^3+1 = 11001$**
  - AND-Gates can be omitted
  - XOR-gates only required for factors $\neq 0$ in the generator polynomial
  - last XOR-gate can be omitted as the output is not used



- General representation



CRC Properties

- CRC can detect the following errors (without proof):
  - All single-bit errors
  - Two single-bit errors if (x^k+1) is not divisible by the generator polynomial ($k \leq$ frame size)
  - Odd number of bit errors if $G(x)$ contains factor $(x+1)$
  - All error bursts $\leq$ degree of generator polynomial

- Examples for standardize polynomials:
  - CRC-12 = $x^{12}+x^{11}+x^3+x^2+x+1$
  - CRC-16 = $x^{16}+x^{15}+x^2+1$
  - CRC-CCITT = $x^{16}+x^{12}+x^5+1$
  - CRC-32 = $x^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^8+x^7+x^5+x^4+x^2+x+1$

Video Recommandations

- Reliabale Data Transmission - https://www.youtube.com/watch?v=eq5YpKHXJDM
- Error Correction: Parity - https://www.youtube.com/watch?v=MgkhrBSjhag
- Error Correction: CRC - https://www.youtube.com/watch?v=izG7qT0EpBw and https://www.youtube.com/watch?v=sNkERQIK8j8

Task 3 and Task 4: CRC
Please present your solution

Time Allocated 10 min

Thank you for your attention