# Exercise (SS 2022)
# Comunication Systems and Protocols

Institut fuer Technik der Informationsverarbeitung
Dr. Jens Becker, Matthias Stammler M. Sc.

## Task 1: Serial Interface

7

In the figure 1.1 the pulse diagram of a RS232 interface is given. Different transmission frames have been used for the communication. A transmission frame is composed of a start bit ('0'), 5-8 data bits, no (N, none) or one bit for even (E, even) or odd (O, odd) parity, as well as at least 1 or 2 stop bits (,1'). Possible frame formats are [5..8][N,O,E][1,2], for example 8N1 for 8 data bits, no parity bit and at least 1 stop bit.

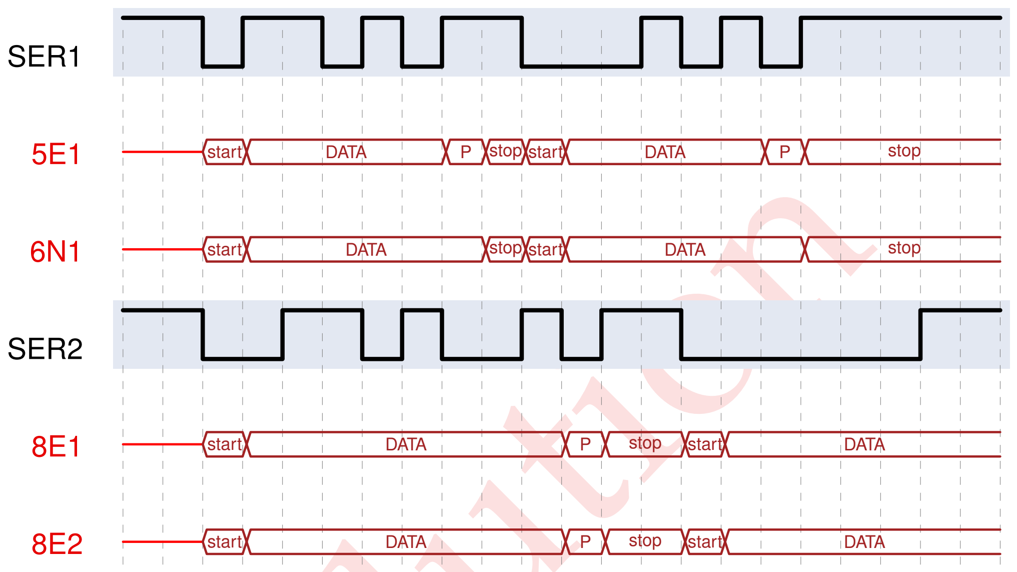


Figure 1.1: Serial interface pulse diagram

1.1 Give all possible frame formats for the pulse sequences as shown in figure 1.1. All given pulse sequences are describing a correct transmission. Start of a transmission is always the startbit in the third timestep. 4

For SER1, 5E1 and 6N1 are possible frame formats as both meet the requirements.
For SER2, since it is not mentioned if the Sender is sending continously or not, there are multiple frame formats possible as well.
For SER2, 8E1 and 8E2 are both possible.
8E1 allows atleast one end bit and 8E2 allows atleast 2 end bits, and both are meeting the requirements here.

1.2 In the figure below different pulse sequences for a RS232 interface are given. Derive from the figure and the given frame formats if the transmission was error free. Mark the erroneous parts in the pulse diagrams. 2

1.3 Is it possible to detect errors without knowing the frame formats? 1

Not in general. The frame format gives the position and meaning of a parity bit for example.

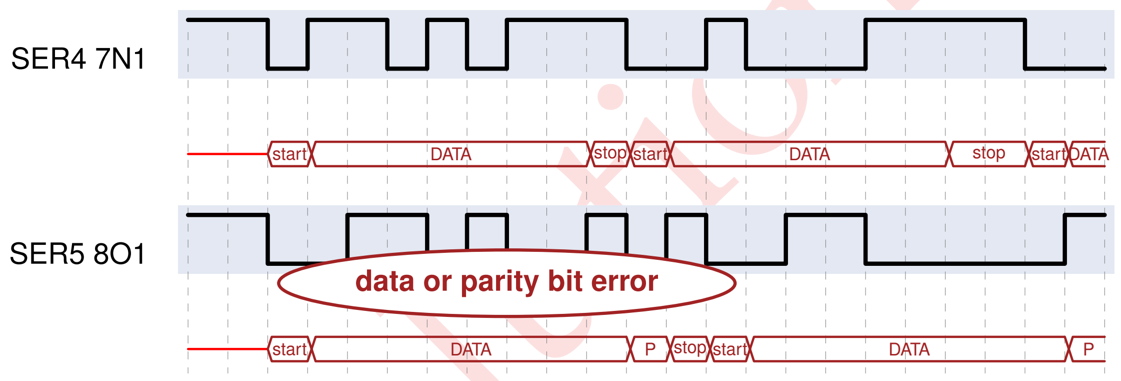


Figure 1.2: RS232 pulse sequences

## Task 2: Flow-Control

6

A communication system is given in Figure 2.1. The sender's clock frequency is 1 MHz, the receiver's is 200 kHz. Both partners work synchronously to their own clock signal and try their best to communicate as fast as possible. They apply a Level-triggered Closed-loop Flow Control corresponding to Figure 3.1 for the high-level synchronization.

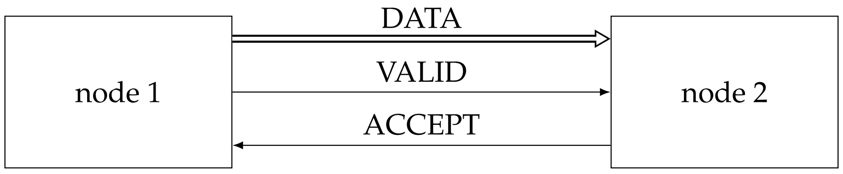


Figure 2.1: Level-triggered Closed-loop Flow Control

2.1 In Figure 2.2 the sensitive clock edges of the sender and the receiver as well as the signal values for the first sender clock period are shown. In order to avoid violations of setup and hold times, the data is put onto the bus and one clock cycle later the valid signal is set to '1' by the sender. The receiver will also set the accept signal one clock cycle after having received the data. Fill in the progression of all signal lines until the end of the time scale.

4

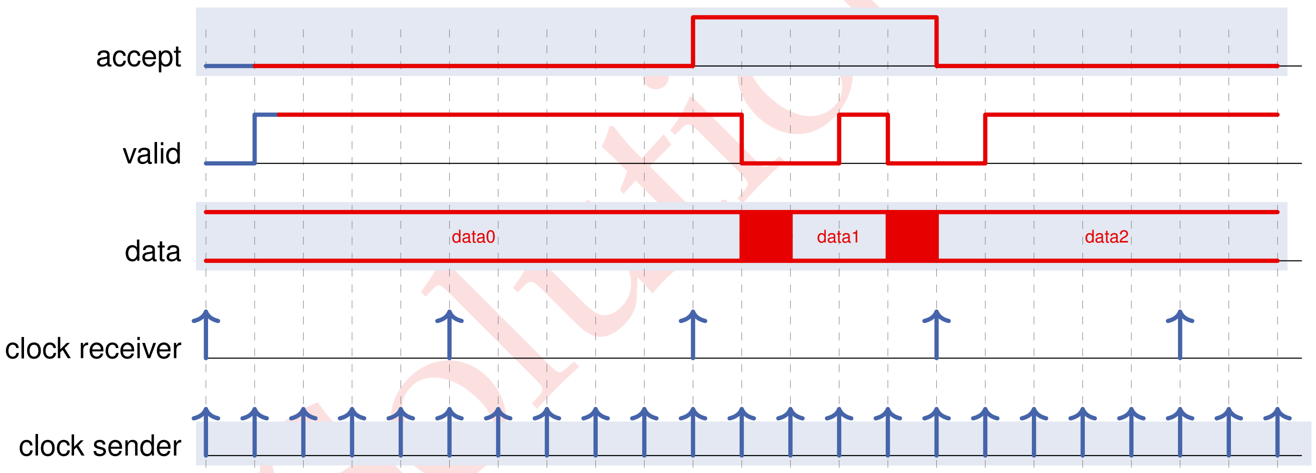


Figure 2.2: Signal progression diagram

2.2 Is this kind of synchronization free from error in this specific case? Justify your answer.

1

No. One can see that data1 got lost because it was put on to the data lines and removed by sender while receiver is still busy processing data0.

2.3 Propose a better solution for this communication scenario.

1

1. Clock divider in sender so that both are only clocked at 200 kHz
2. Apply an Edge-triggered Closed-loop Flow Control
3. Clock down the sender: to avoid this loss of data, the following equation must be satisfied: $T_r \leq 4T_s$

# Task 3: Cyclic Redundancy Check

8

To protect a data transmission, CRC with the generator polynomial $g(x) = x^2 + 1$ is used.

3.1 Determine the bit string that is associated with the generator polynomial. 1

Polynomial Generator: $1 \cdot x^2 + 0 \cdot x^1 + 1 \cdot x^0 \rightarrow$ Bitstring: 1 0 1

3.2 What is the length of the checksum that is to be appended to the data stream? 1

Length of the checksum = Order of polynomial generator. here: order $g(x) = 2$

3.3 Calculate the data stream that will be transmitted if the following bit string is to be protected: 1001010101. 2

Division where
Divisor : 101
Dividend : 100101010100

Computation is provided below. If the leftmost bit is '1', use divisor '101'. When the leftmost bit is '0', instead of using divisor '000' , the next bit of the dividened is pulled down.

$$
\begin{array}{ccccccccccccccc l}
1 & 0 & 1: & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & \\
\hline
 & & & 1 & 0 & 1 & & & & & & & & & & \\
 & & & 0 & 0 & 1 & 1 & 0 & & & & & & & & \\
 & & & & & 1 & 0 & 1 & & & & & & & & \\
 & & & & & 0 & 1 & 1 & 1 & & & & & & & \\
 & & & & & & 1 & 0 & 1 & & & & & & & \\
 & & & & & & 0 & 1 & 0 & 0 & & & & & & \\
 & & & & & & & 1 & 0 & 1 & & & & & & \\
 & & & & & & & 0 & 0 & 1 & 1 & 0 & & & & \\
 & & & & & & & & & 1 & 0 & 1 & & & & \\
 & & & & & & & & & 0 & 1 & 1 & 1 & & & \\
 & & & & & & & & & & 1 & 0 & 1 & & & \\
 & & & & & & & & & & 0 & 1 & 0 & 0 & & \\
 & & & & & & & & & & & 1 & 0 & 1 & & \\
 & & & & & & & & & & & 0 & 0 & 1 & 0 & \text{Remainder}
\end{array}
$$

Bit string to be transmitted: 1001010101 10

## Reception

In a transmission system that uses CRC for error protection, a sender transmits the following bit stream: 100101010110. Due to interferences during transmission the last 4 bits of the bit stream are flipped before reaching the receiving node.

3.4 Denote the bit stream as it arrives at the receiving node. 1

Received bit stream: 100101011001

3.5 Carry out the CRC error detection scheme of the receiver assuming that the generator polynomial $g(x) = x2 + 1$ has been used. 2
What does the receiver conclude from the result? Explain and discuss the reasons for the receiver's conclusion.

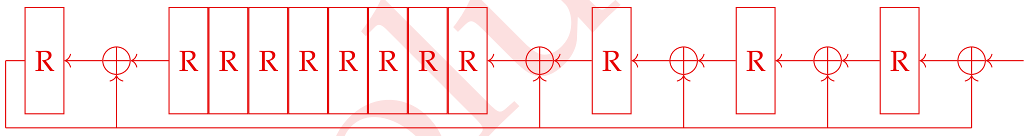
Division where
Divisor : 101
Dividend : 100101011001

```
1  0  1: 1  0  0  1  0  1  0  1  1  0  0  1
---------------------------------------------
         1  0  1
         0  0  1  1  0
               1  0  1
               0  1  1  1
                  1  0  1
                  0  1  0  0
                     1  0  1
                     0  0  1  1  1
                           1  0  1
                           0  1  0  0
                              1  0  1
                              0  0  1  0  1
                                    1  0  1
                                    0  0  0  Remainder
```

## Hardware Implementation

3.6 To protect data transmissions in a mobile device, the CRC scheme is to be implemented using linear feedback registers with XOR operations. Draw the simplified hardware layout for the polynomial CRC-12 ($x^{12} + x^{11} + x^3 + x^2 + x + 1$).

1



# Task 4: Error Protection

19

## Cyclic Redundancy Check (CRC) - Exam Question

4.1 Does the CRC scheme based on the generator polynomial $G(x) = (x + 1)(x^2 + 1)$ allow a receiver to detect all error patterns with exactly three erroneous bits? Justify.

2

Yes. Since $(x + 1)$ is a factor of the generator polynomial, every odd number of bit errors is detectable.

***Correction hints:*** *2 pt. for the correct answer.*

4.2 The receiver of a CRC-protected message performs the CRC error detection procedure and calculates a non-zero remainder. What can it reliably conclude with respect to the occurrence of a transmission error?

2

The receiver can conclude that a transmission error occurred.

***Correction hints:*** *2 pt. for the correct answer.*

4.3 Two nodes use the generator polynomial $G(x) = x^4 + x + 1$ to exchange CRC-protected messages. To transmit a certain message, the sender calculates the corresponding checksum, appends this checksum to the raw message, and finally transmits the bit string

3

0010 1110 1010 0011.

Due to transmission errors, however, the recipient receives the bit string

0010 1000 0010 0011.

Is the receiver, who is aware of $G(x)$, able to detect this error? Justify your answer based on the error pattern and the specific error detection capabilities of $G(x)$.

***Hint:*** *Do not perform the calculation that the receiver has to perform to detect errors!*

Since the degree of $G(x)$ is 4, all burst errors of length 4 can be detected. The error pattern can be interpreted as a burst error of length 4 and is therefore detectable.

***Correction hints:***

- *+2 pt. for describing the relevant error detection capability.*
- *+1 pt. for the correct conclusion (only in addition).*

4.4 Given the generator polynomial $G(x) = x^5 + x^4 + 1$, what calculation does the receiver of the CRC-protected bit string "1111 0110 0001" perform as part of its error detection procedure? Give both the dividend and the divisor of this calculation as a bit string.

3

***Hint:*** *This question does not require you to perform the calculation!*

The receiver calculates the remainder of 1111 0110 0001 : 110001.

***Correction hints:***

- *+2 pt. for the dividend.*
- *+1 pt. for the divisor.*

4.5 To transmit it over a channel, the message “0100 0111 01” shall be protected by a CRC checksum. Using the generator polynomial $G(x) = x^4 + x^3 + x + 1$, calculate this checksum and give the bit string that is sent to the receiver. 6

```
0 1 0 0   0 1 1 1   0 1 0 0   0 0   :   1 1 0 1 1
  1 1 0   1 1
-------------------------------------------------
  0 1 0   1 0 1
    1 1   0 1 1
-------------------------------------------------
    0 1   1 1 0 1
      1   1 0 1 1
-------------------------------------------------
      0   0 1 1 0   0 1
            1 1 0   1 1
-------------------------------------------------
          0 0 0   1 0 0 0   0
                  1 1 0 1   1
-------------------------------------------------
                  0 1 0 1   1 0
                    1 1 0   1 1
-------------------------------------------------
                    0 1 1   0 1
```

The sender will transmit the bit string “0100 0111 0111 01”.

***Correction hints:***

- *Starting from 5 pt. for the correct calculation, -3 pt. for not adding the correct number of zeros to the dividend and -2 pt. for every other error.*
- *In addition, +1 pt. for the transmitted bit string.*

4.6 Give the CRC generator polynomial that is implemented by the linear feedback shift register shown in Figure 4.1. 3

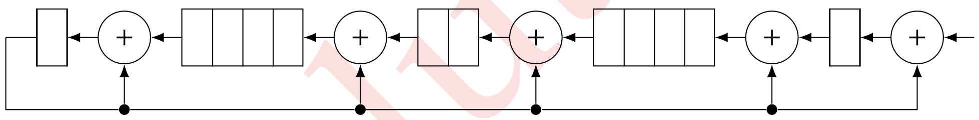


Figure 4.1: Simplified shift register implementation of a CRC scheme

The shift register implements the generator polynomial $G(x) = x^{12} + x^{11} + x^7 + x^5 + x + 1$.

***Correction hints:*** *3 pt. for the correct answer, 1 pt. if there is one mistake.*