



Silesian  
University  
of Technology



RESEARCH  
UNIVERSITY  
EXCELLENCE INITIATIVE  
Ministry of Science  
and Higher Education

# COMMUNICATION METHODS IN DIGITAL SYSTEMS

Dr hab. inż. Adam Milik, Prof. PŚ,  
Katedra Systemów Cyfrowych i Mikroprocesorowych



**Curiculum sponsored by INTEL**



# Outline

1. Shared bus system
  1. Concepts, properties, implementation
  2. Bus Functional Model – Reduced complexity modeling
2. Point-to-point bus systems
  1. AXI-3 and AXI-4
  2. AXI -Stream
  3. AMBA APB
  4. Wishbone – Open Bus System
3. Serial communication
  1. Principles of data transmission and reception
  2. Asynchronous and synchronous communication

# Outline

1. Serial bus systems
  1. Serial Peripheral Bus - SPI
  2. I<sup>2</sup>C advanced peripheral bus system
  3. 1-Wire network bus system
  4. JTAG (IEEE1149.1) diagnostic and programming interface
2. Ethernet interface – a wide band and long distance communication
  1. Interfacing, PHY concept
  2. Data transmission basics, Encapsulations
3. PCIe – a wideband serial interface

Communication Methods in Digital Systems

# Serial transmission

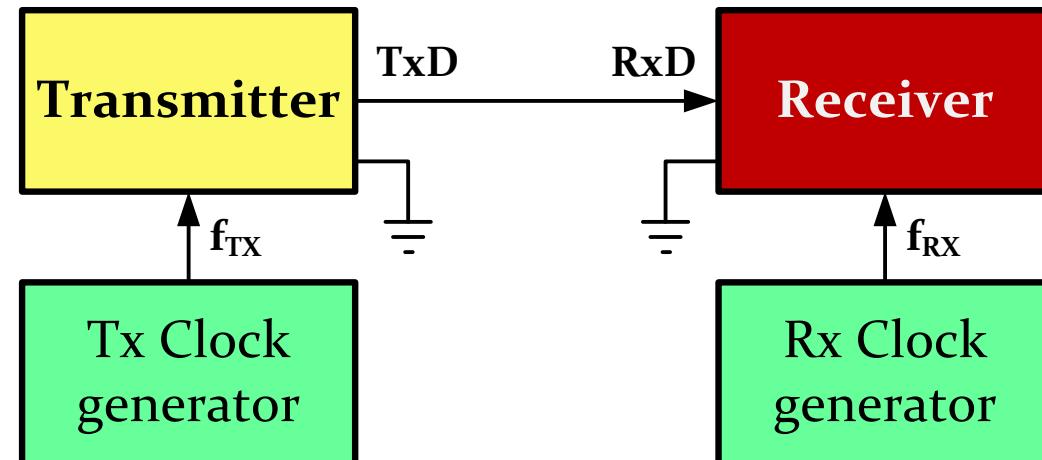
Asynchronous data transmission

# Outline

- Principles of serial asynchronous data transmission
  - Synchronization concept
  - Clock Generator requirements
  - Data recovery
- Clock generators tolerance
- Synchronous data transmission
  - Combining the clock signal into data
  - Deterministic synchronization events

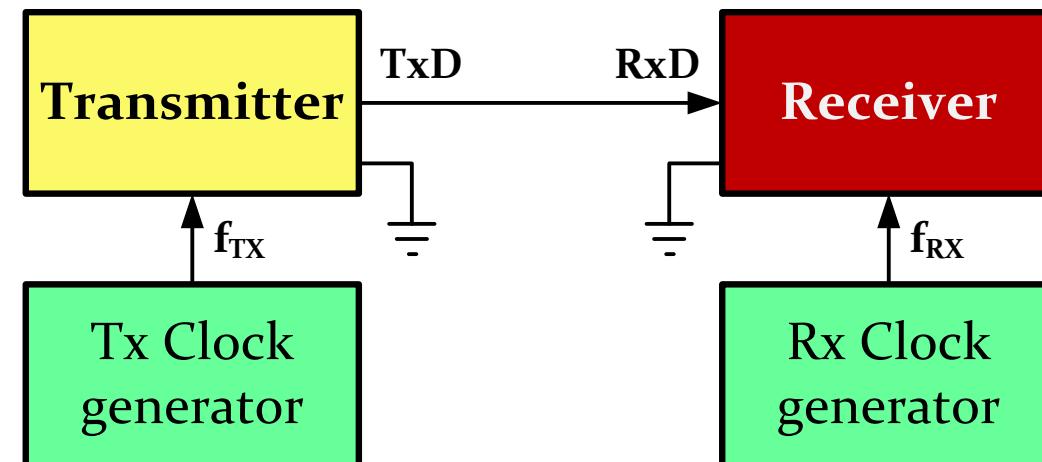
# Introduction

- How to transfer a data between transmitter and receiver?
  - Identical data format
- How to synchronize receiver?
  - Is it possible to achieve  $f_{TX} = f_{RX}$  in physical circuits

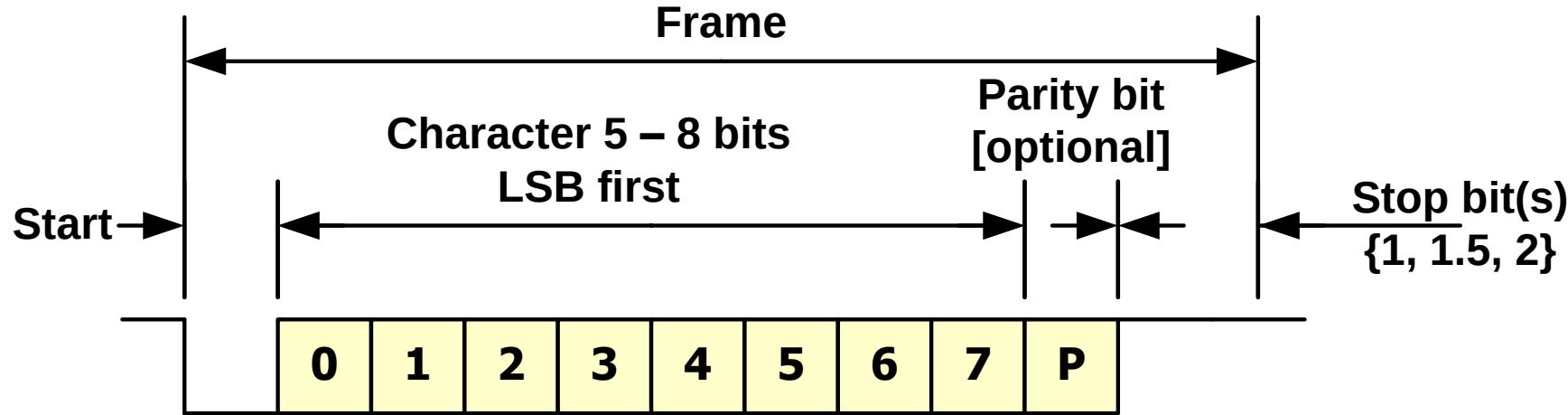


# Clock signal relation

- Clock generator relation
  - Typically  $f_{TX} \neq f_{RX}$  – asynchronous relation
  - **Plezjochronous relation**  $f_{TX} = f_{RX} \pm \delta$
  - Frequency error  $\delta$  must be small enough to assure correct reception



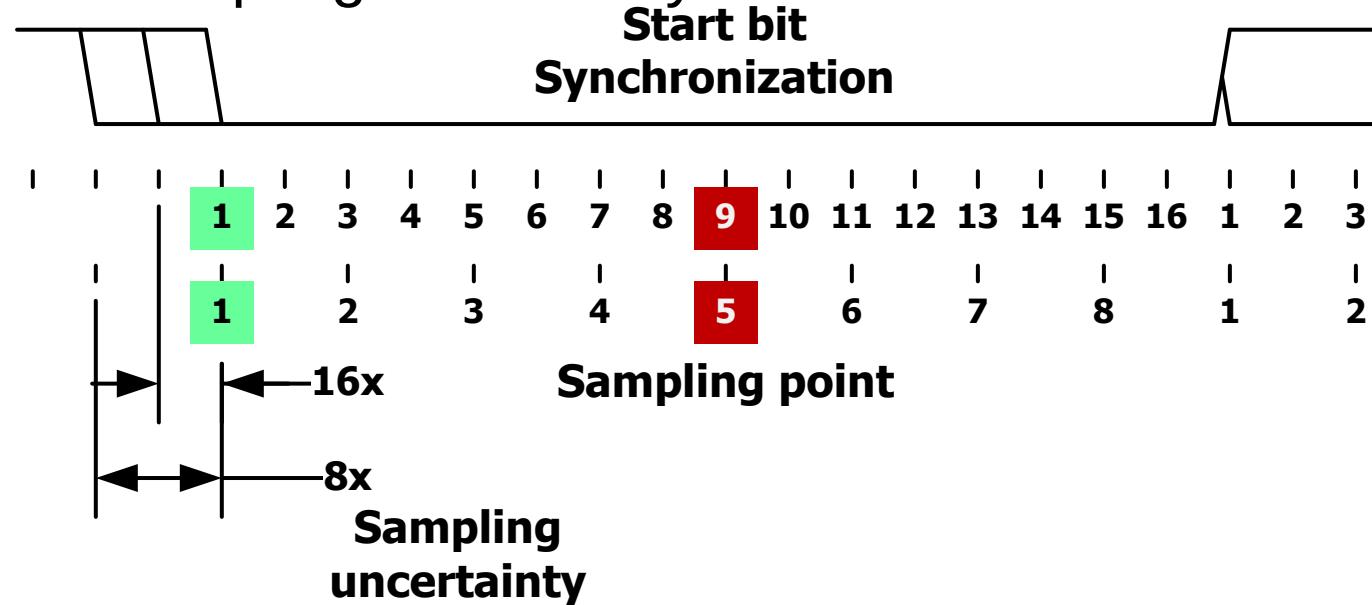
# Asynchronous transmission



- Data format
  - Start bit – exactly 1 - synchronization
  - Character 5 – 8 bits
  - Integrity control – 1 bit (parity even/odd or additional data)
  - Stop bit(s): 1, 1.5, 2 – time gap before next frame

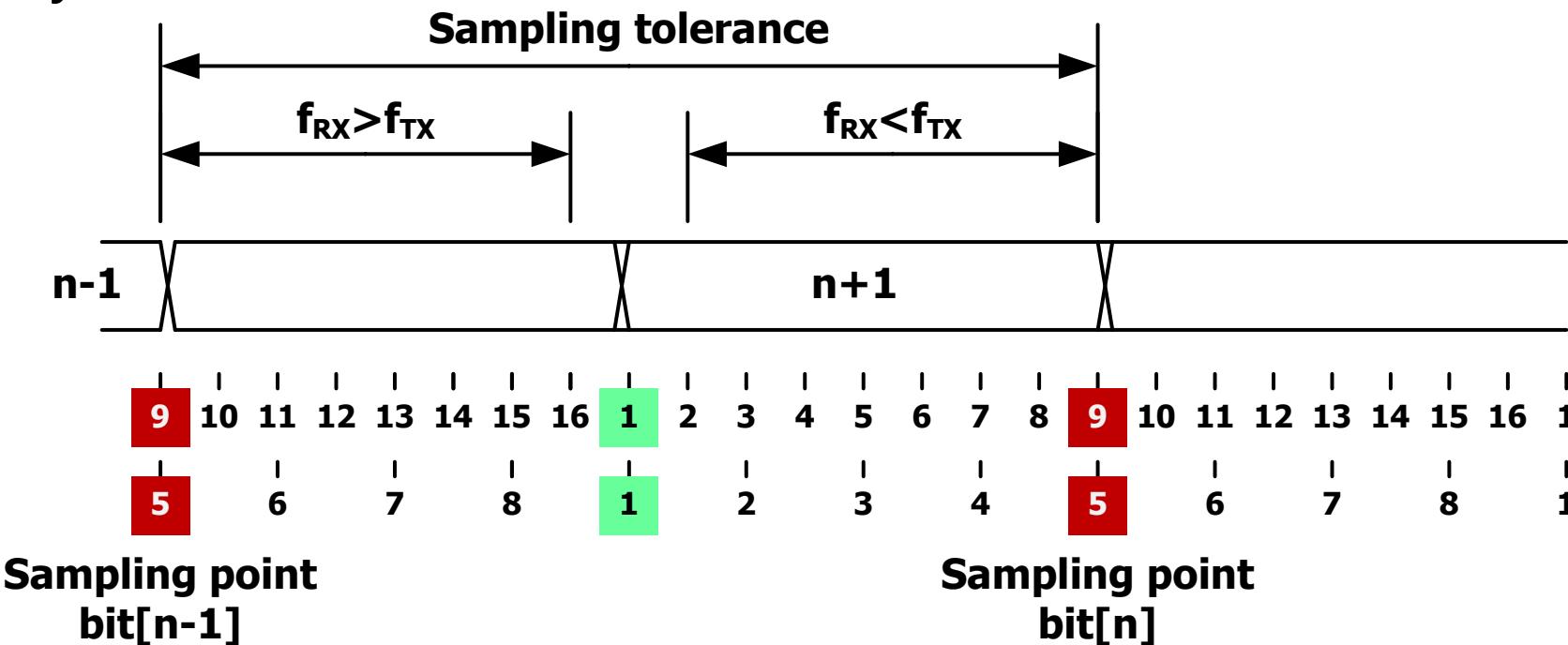
# Asynchronous transmission

- Idea of reception - synchronization
    - An event – change of the state on the RxD line
    - An event based concept
    - Synchronous sampling uncertainty



# Asynchronous transmission

- Reception stability - frequency tolerance
  - sampling uncertainty
  - Frequency tolerance



# Correct reception range

- Relative frequency error

$$\delta_f = \frac{\frac{N}{2} - 1}{b \cdot N} = \frac{\frac{1}{2} - \frac{1}{N}}{b}$$

$$\lim_{N \rightarrow \infty} \delta_f = \frac{1}{2 \cdot b}$$

- Maximal distance from synchronization event in bits -  $b$
- Oversampling frequency -  $N$

# Correct reception range

- Frame length
  - Minimal – start + character 5 + stop = 7 bit
  - Maximal – start + character 8 + parity + 2 stop = 12 bit

	Oversampling		
Frame length	8	16	INF
7	5.35%	6.25%	7.14%
12	3.125%	3.64%	4.16%

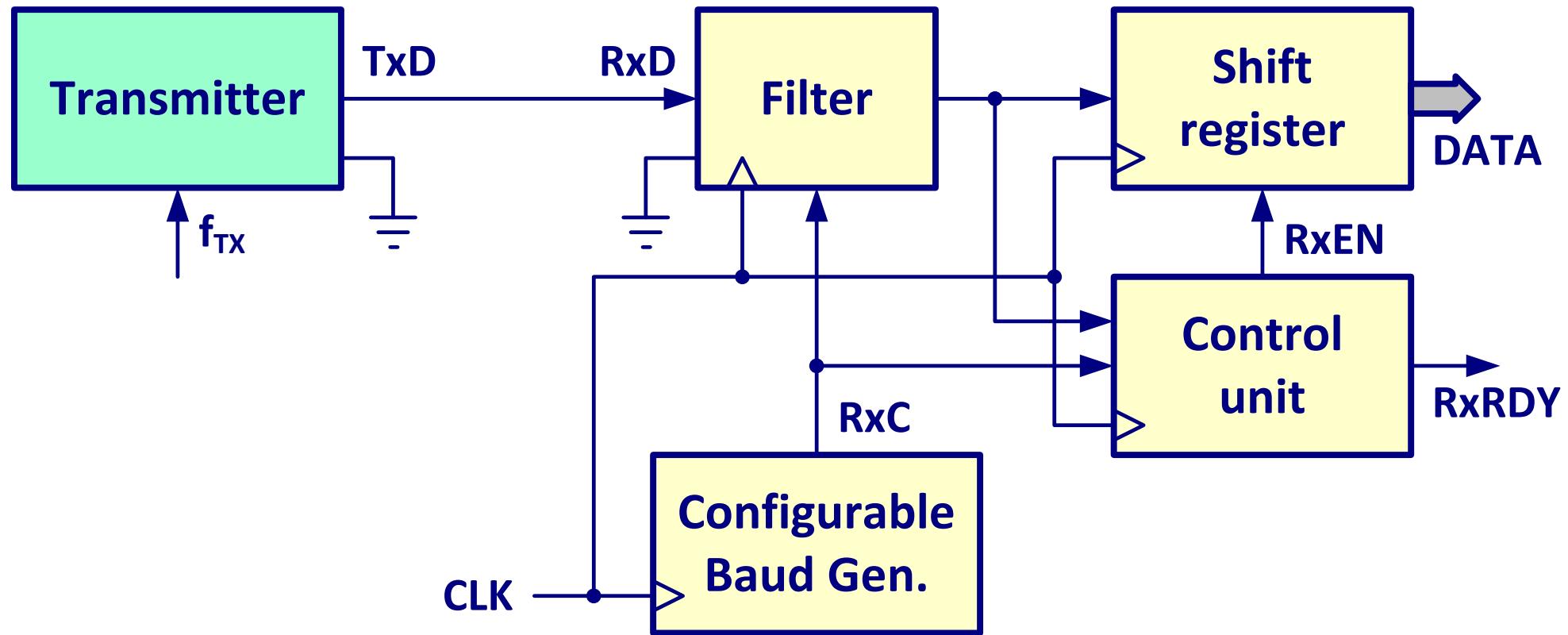
# Transmission overhead

- Frame
  - Useful data – Character
  - Data protection - Parity
  - Organization data Start and Stop bits

	Character	Parity	Stop	Total	Ratio
Worst	5	1	2	9	55.56%
Best (byte)	8	0	1	10	80.00%
Best (9 bit)	8	1	1	11	81.82%
Text	7	0	1	9	77.78%

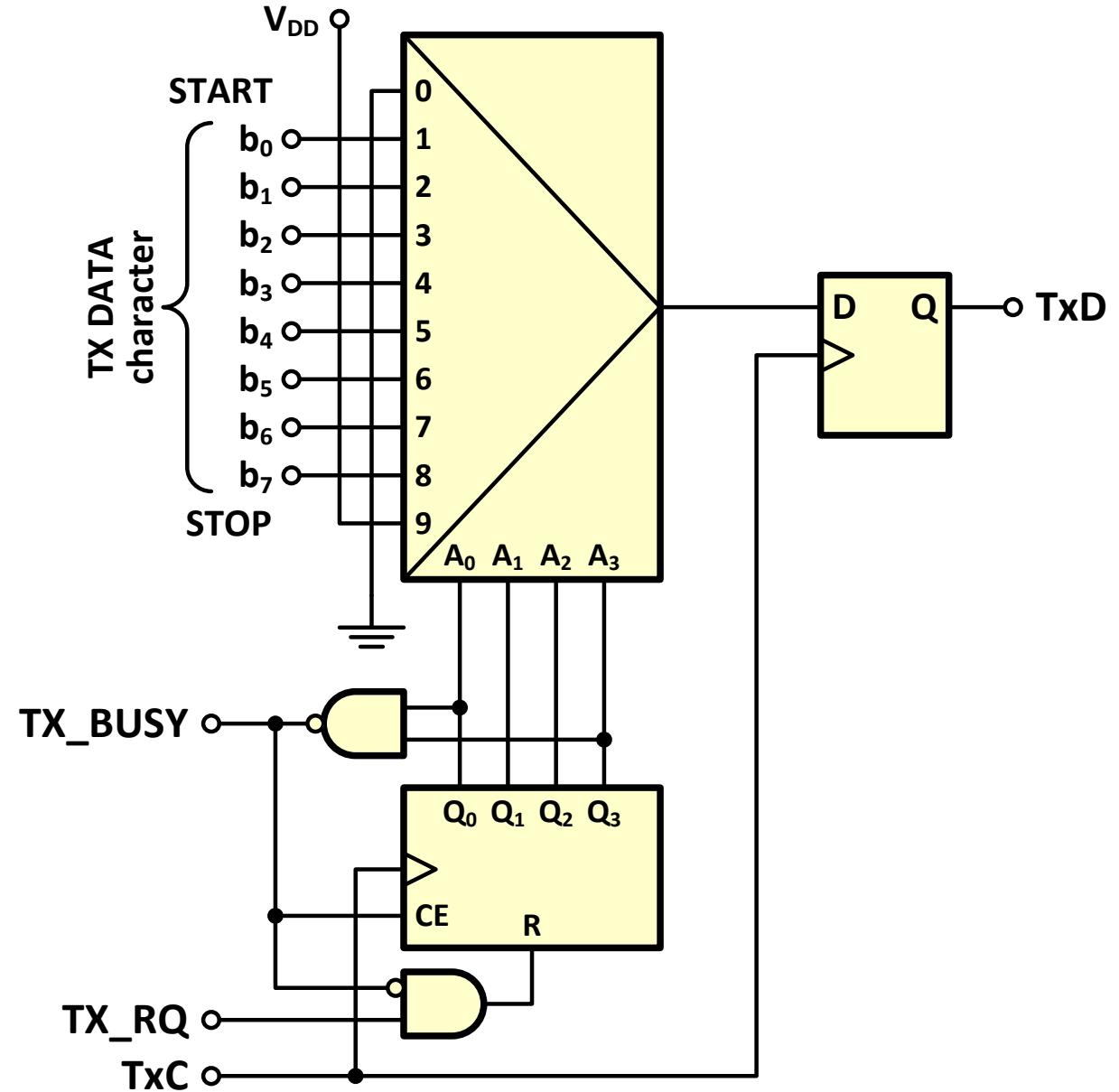
# UART

- General block diagram



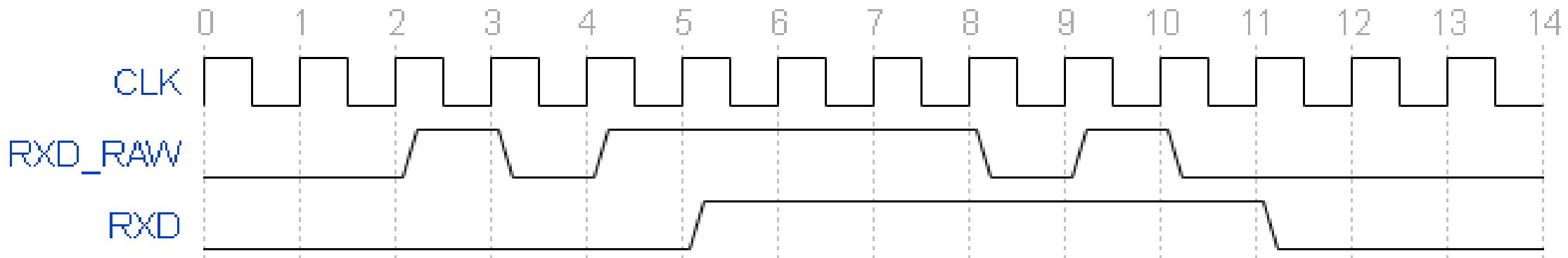
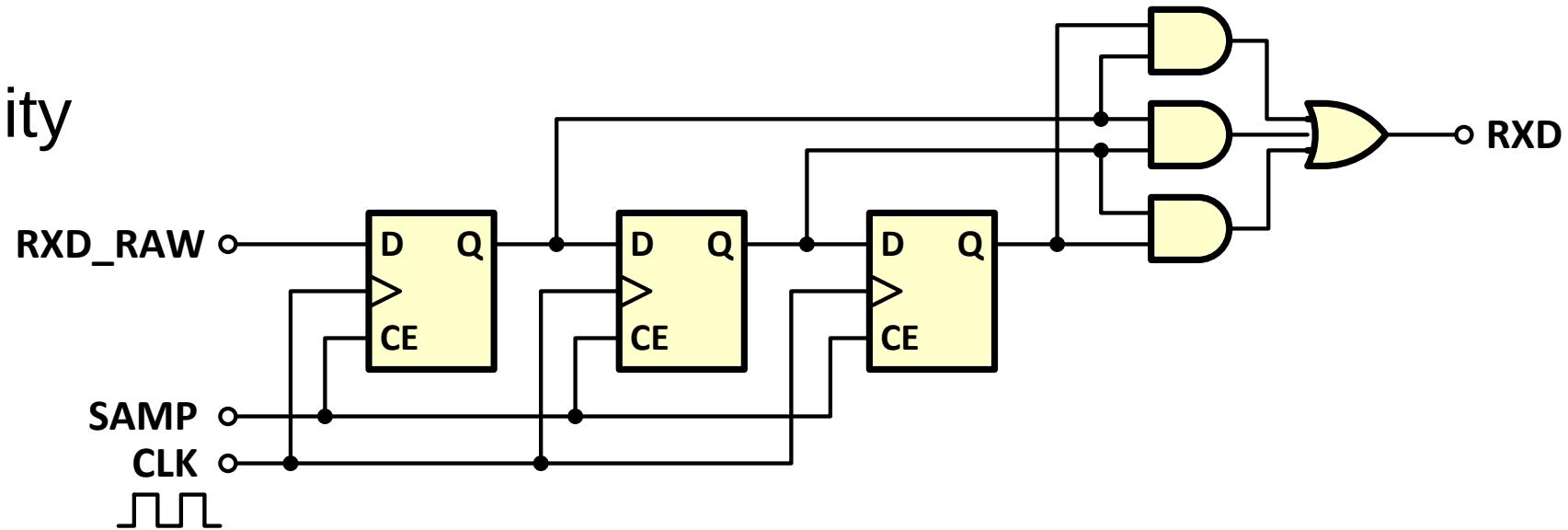
# UART Transmitter

- Parallel to serial converter
- Counter based controller
- Registered output
  - Glitch free output



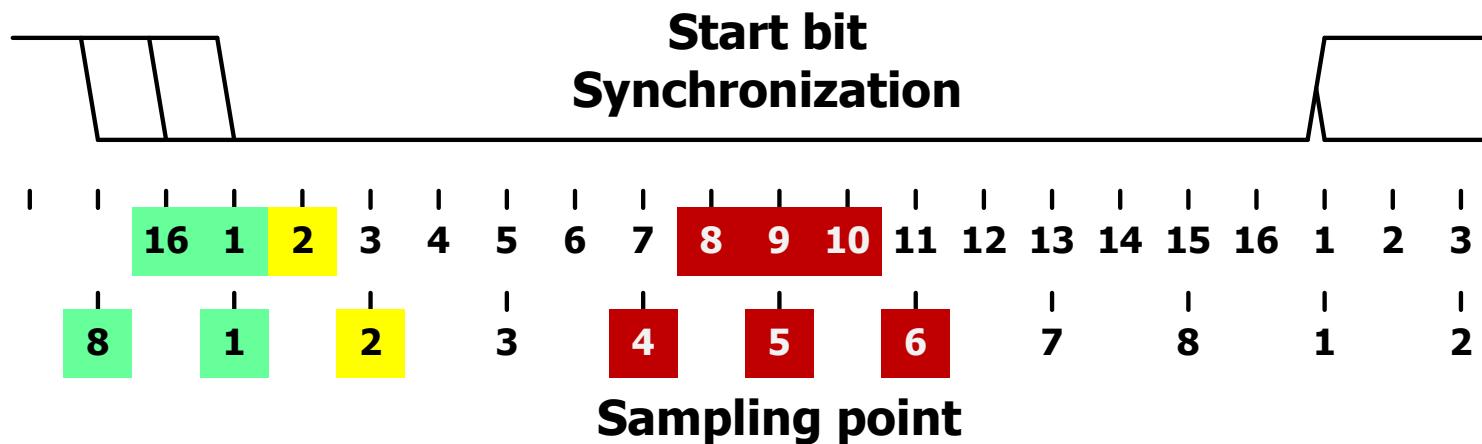
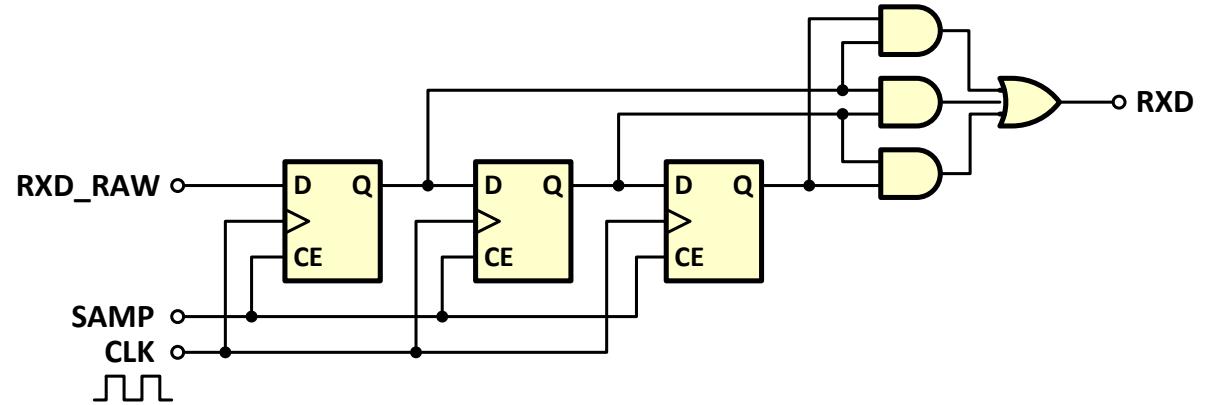
# UART Median filter

- Signal noise immunity
  - Median filter



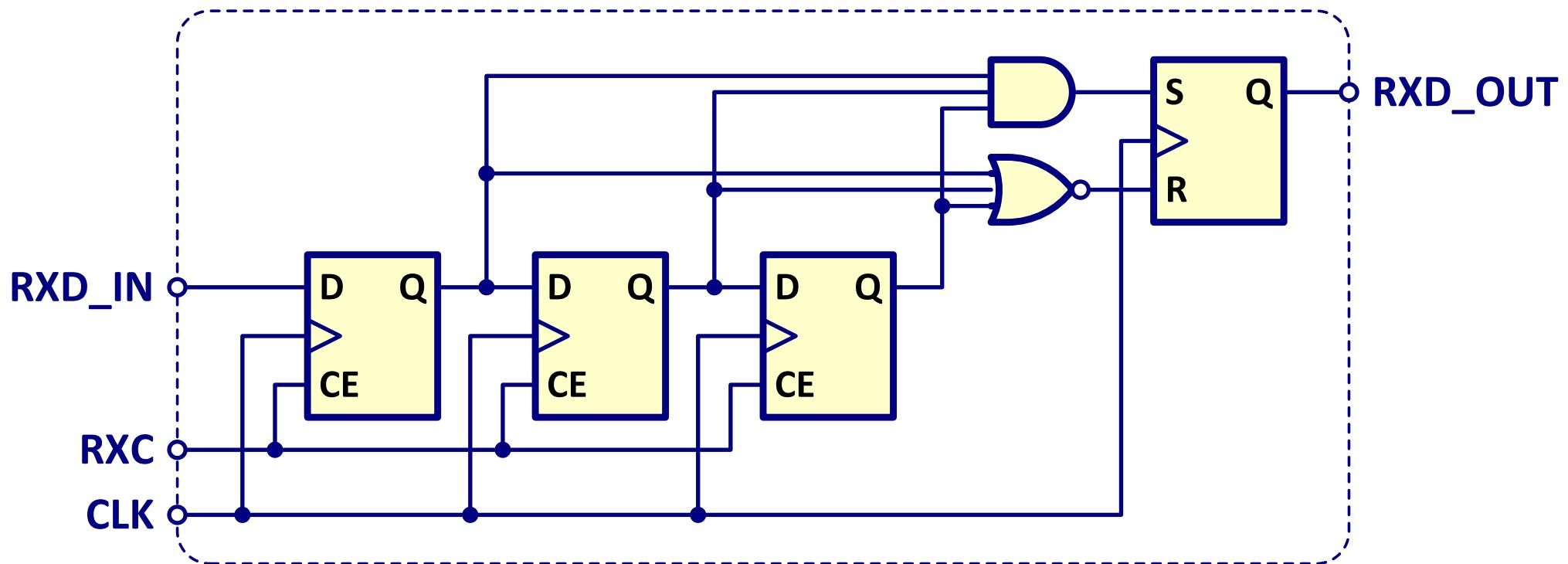
# Median filter – noise suppression

- Signal noise immunity
  - Filtering of received signal



# UART Input filter

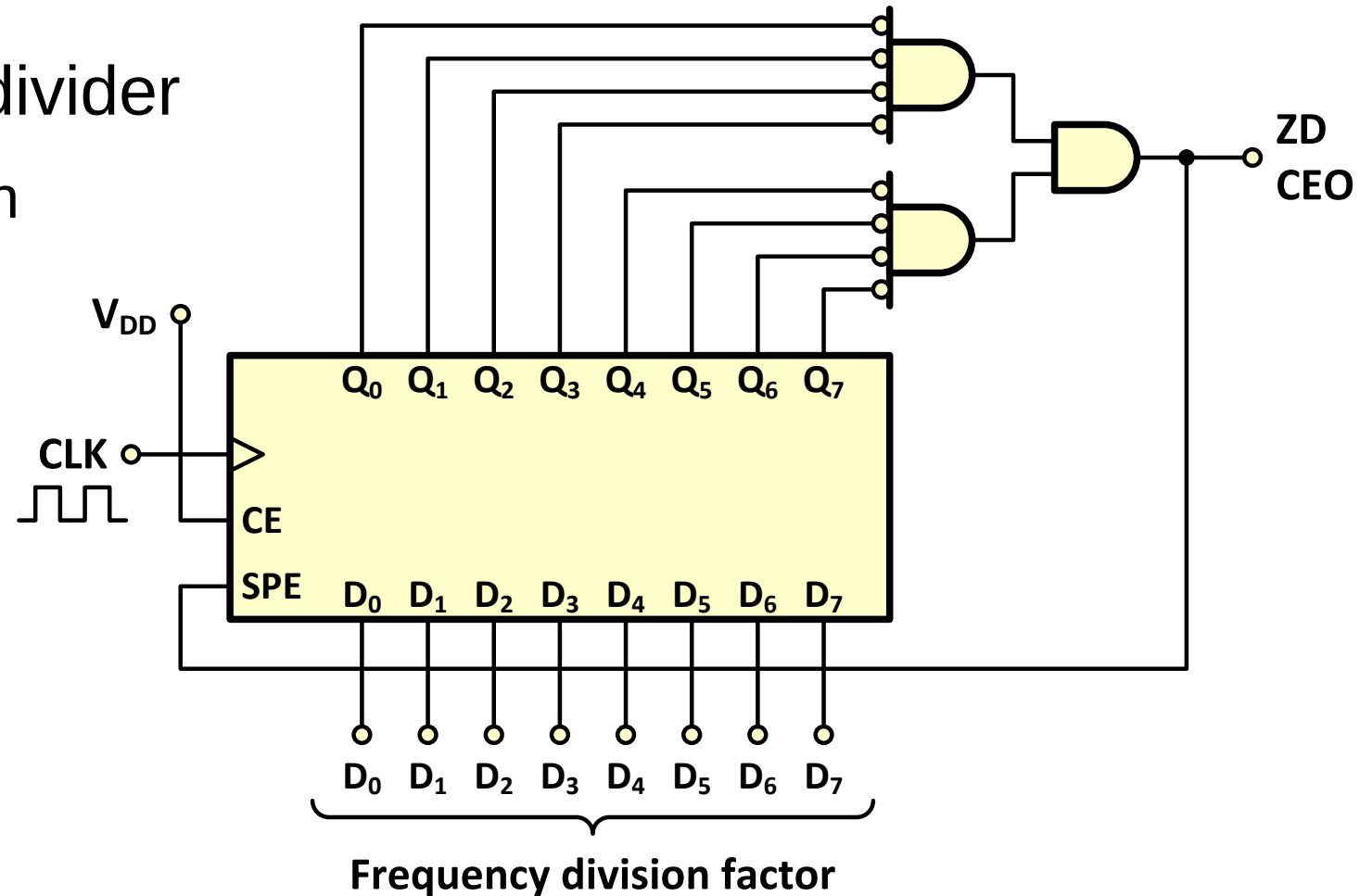
- Input filter – suppressing input glitches



# UART Baud generator

- Programmable frequency divider
  - Programmable integer division
- Down counter with automatic reload at 0

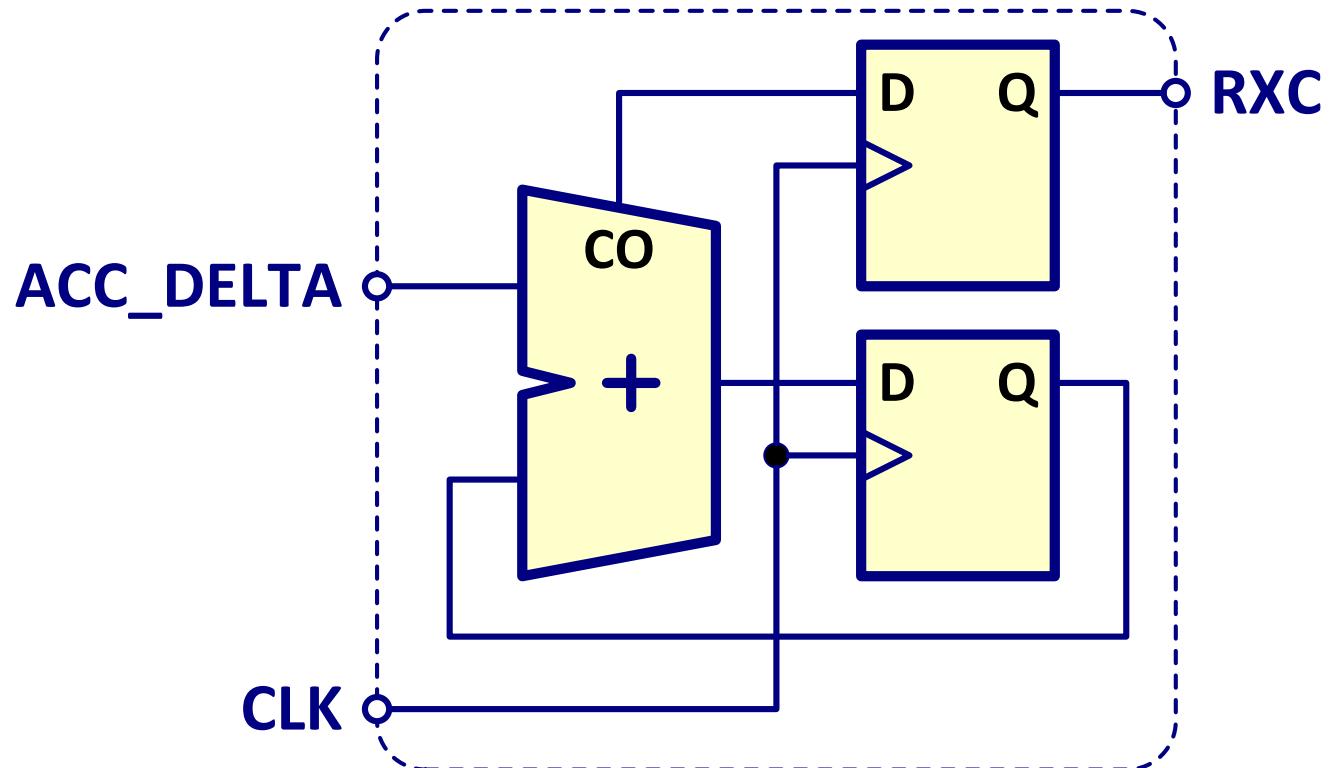
$$f_{CE} = \frac{f_{CLK}}{n + 1}$$



# UART Baud synthesizer

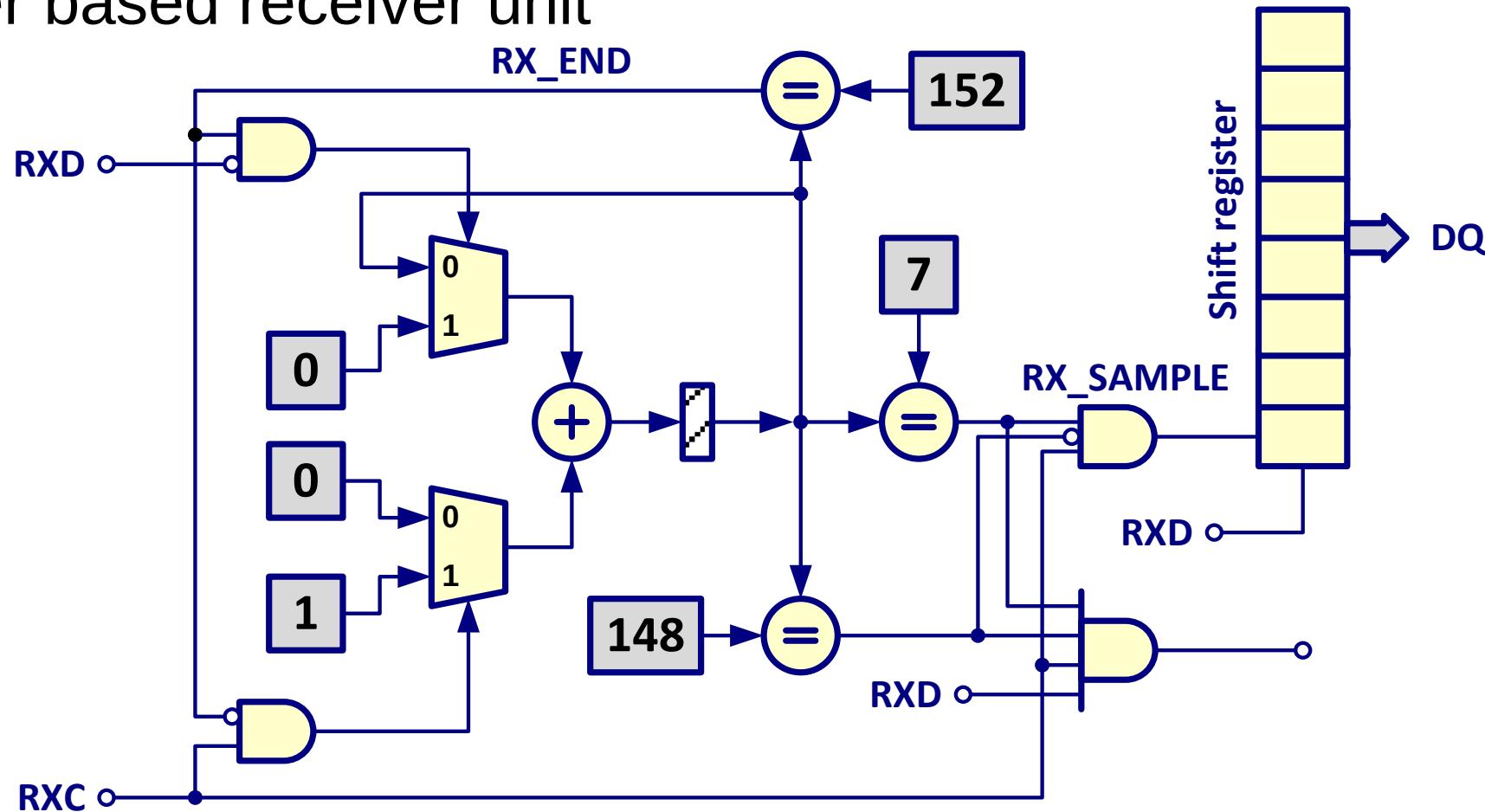
- Direct Digital Synthesis
  - Phase accumulator
  - Carry Out output
- Rational division factor

$$\overline{f_{RXC}} = f_{CLK} \frac{ACC\_DELTA}{2^{ACC\_W}}$$



# UART Receiver

- Counter based receiver unit



# Transmitter model

- Modelling data transmission with a given baud

```
task SEND;
  input [7:0] data;
  integer i;
begin
  //Event correct
  @(posedge CLK);
  //Start bit
  RXD = 1'b0;
  repeat(BAUD) @(posedge CLK);
  //Data frame
  for(i=0; i < 8; i = i + 1) begin
    RXD = data[i];
    repeat(BAUD) @(posedge CLK);
  end
  //Stop bit
  RXD = 1'b1;
  repeat(BAUD) @(posedge CLK);
end
endtask
```

# Receiver model

- Model of the receiver

```
parameter CLOCK_RATIO = 8;
parameter CLOCK_RATIO_2 = 4;
integer rx_cnt;
reg [7:0] DATA;
event EV_RCV;
```

```
initial begin
    DATA = 8'h00;
    rx_cnt = 0;
    repeat(10) @(posedge CLK);
    forever begin
        while(TXD) @(posedge CLK);
        repeat(CLOCK_RATIO_2) @(posedge CLK);
        repeat(8) begin
            repeat(CLOCK_RATIO) @(posedge CLK);
            DATA = {TXD,DATA[7:1]};
        end
        repeat(CLOCK_RATIO) @(posedge CLK);
        rx_cnt = rx_cnt + 1;
        $display("RXD(%d) -> %h", rx_cnt, DATA);
        ->EV_RCV;
    end
end
```

# Transmission errors

- Frame overrun error
  - Overwriting data in the buffer
- Frame error
  - Incorrect value of stop bit found **0** expected **1**
- Parity error
  - Mismatch of calculated parity field with received value
- Timeout error
  - Timeout encountered before the arrival of the next character

Communication Methods in Digital Systems

# Serial transmission

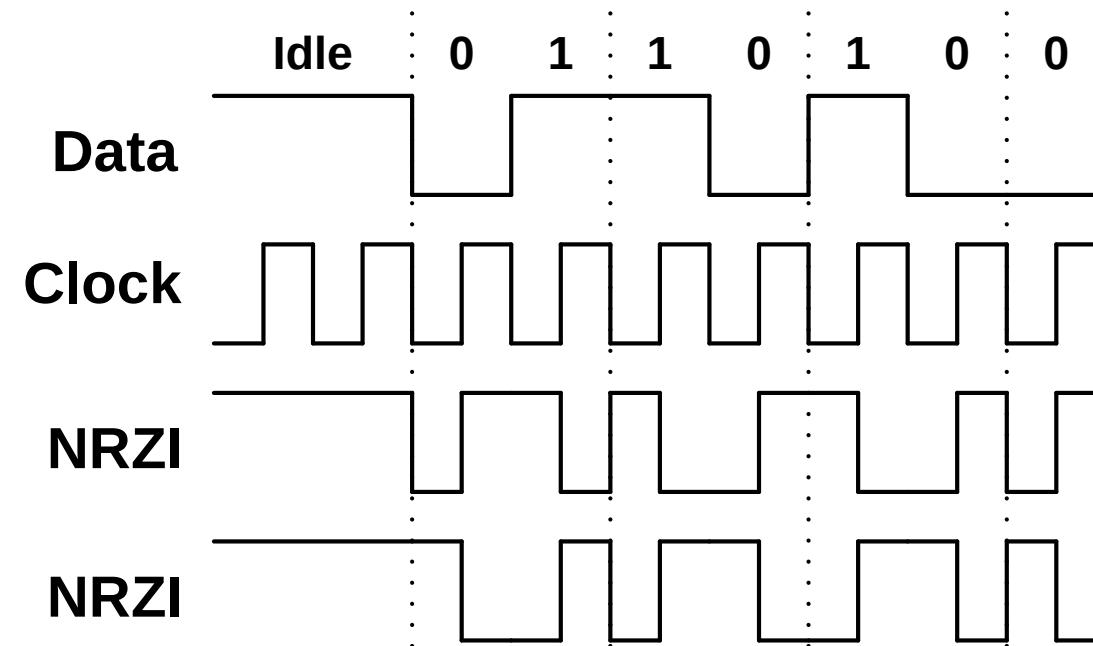
Synchronous data transmission

# Synchronous transmission

- Reducing transmission overhead
  - Eliminating start and stop bits
- How to synchronize receiver
  - Self synchronization - Manchester code
    - Composition of data and clock signal
    - Each bit contains synchronization information
    - Receiver is resynchronized with each bit
    - theoretically 25% tolerance of  $f_{RX}$  is allowed
    - Average signal value for 0 and 1 is the same
    - Eliminates DC component

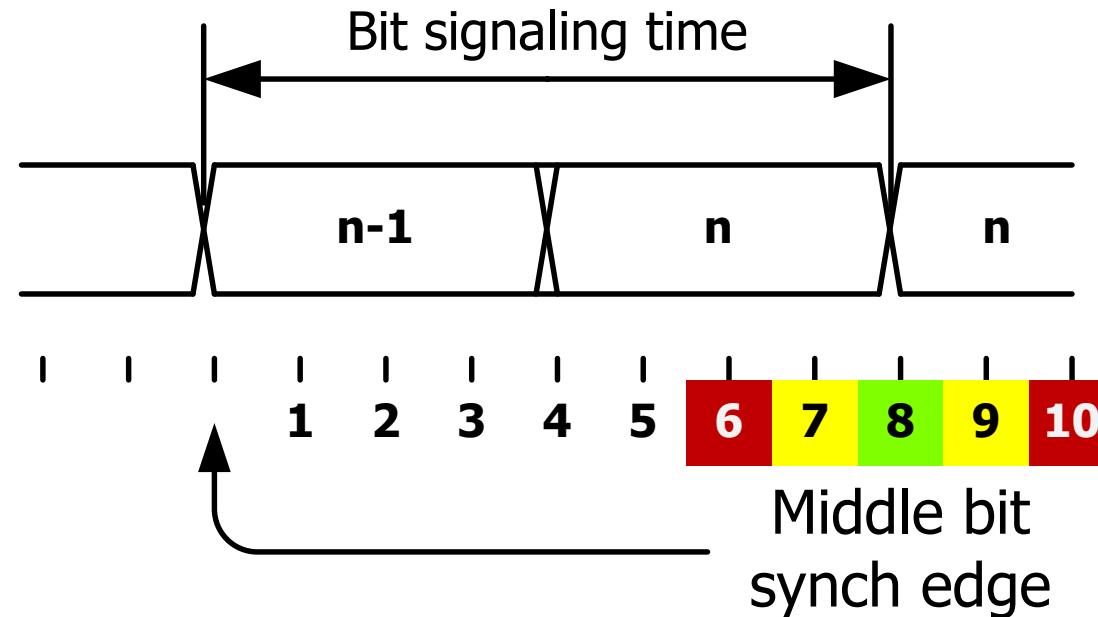
# Synchronous transmission

- Self-synchronization - Manchester code
  - Composition of data and clock signal
  - Eliminates DC component in signal



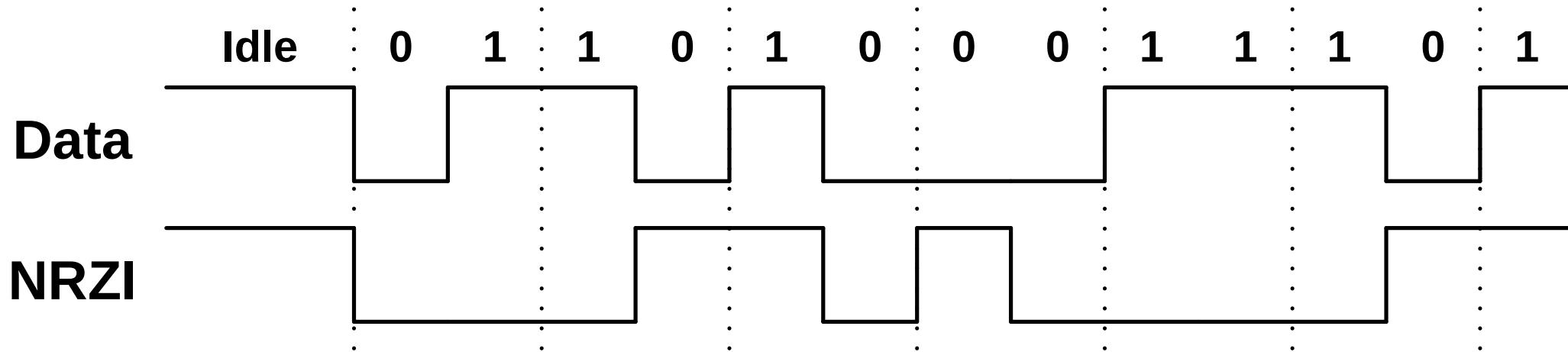
# Synchronization recovery

- Manchester code synchronization recovery
  - Ignore possible changes at the bit boundary
  - Wait for change in the middle of the signaling period



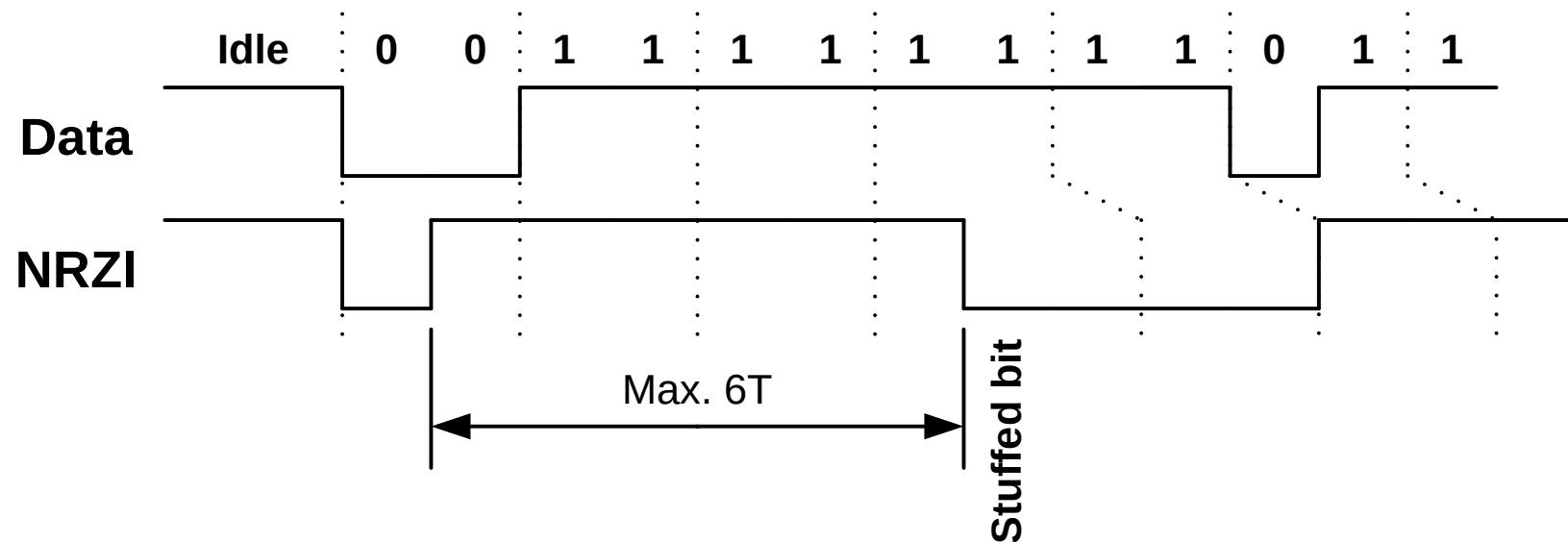
# USB signaling concept

- Reducing frequency band
- Modified encoding
  - 0 – changes state of transmission line
  - 1 – no changes



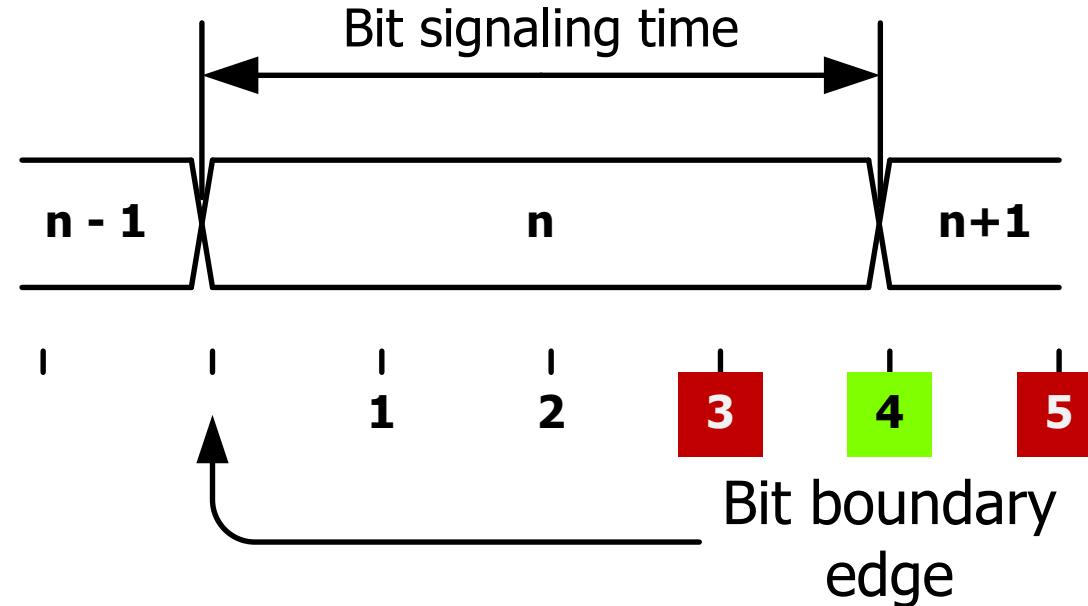
# USB signaling concept

- Problem with synchronization
  - When long sequence of 1 is transmitted
- Idea: force synchronization event by bit stuffing



# Synchronization recovery

- Digital synchronization recovery
- Wait for change at bit boundary
  - Accept changes near ideal synchronization point



# Encoding: 4 bit to 5 bit

- Basic symbols transcoding

Input	4B	5B	Symbol
0	0000	11110	0
1	0001	01001	1
2	0010	10100	2
3	0011	10101	3
4	0100	01010	4
5	0101	01011	5
6	0110	01110	6
7	0111	01111	7

Input	4B	5B	Symbol
8	1000	10010	8
9	1001	10011	9
A	1010	10110	A
B	1011	10111	B
C	1100	11010	C
D	1101	11011	D
E	1110	11100	E
F	1111	11101	F

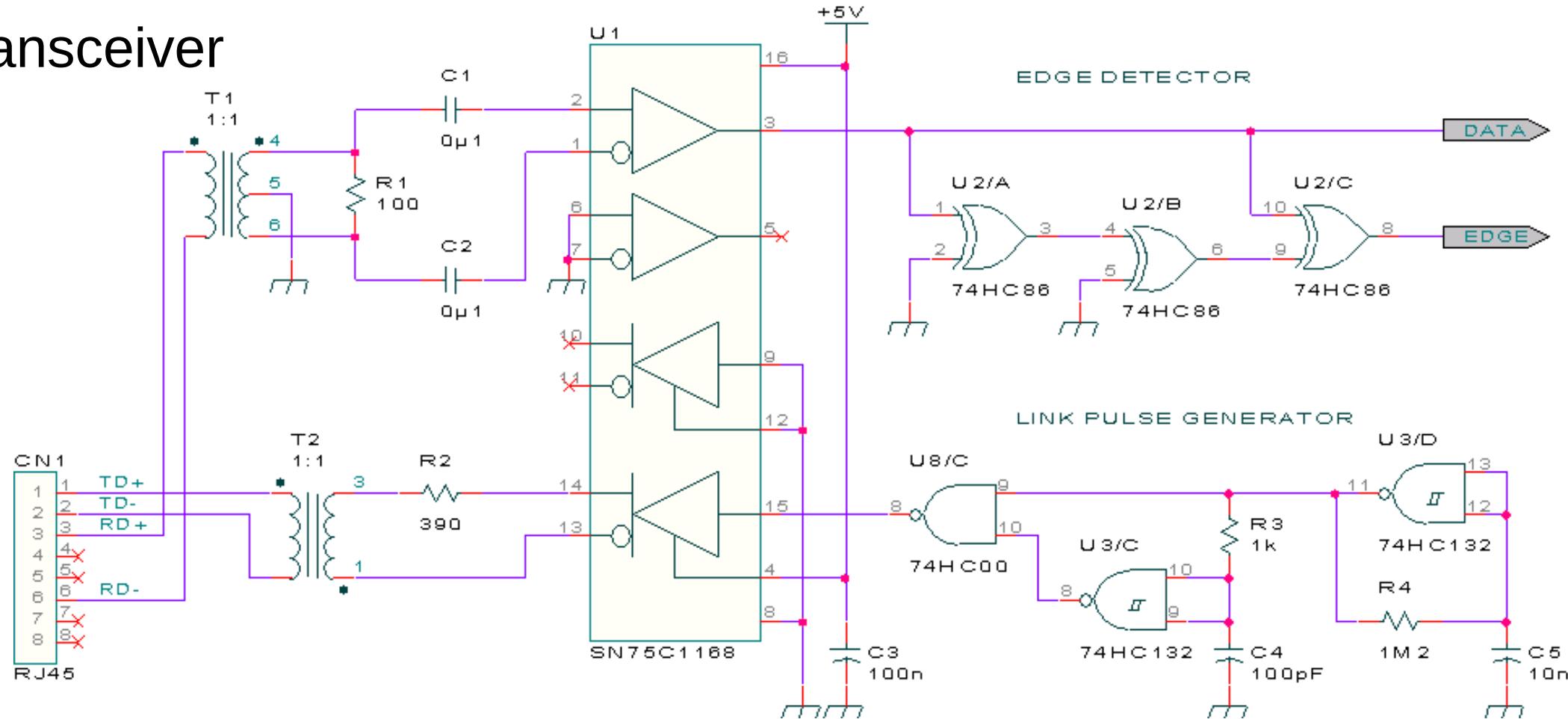
# Encoding: 4 bit to 5 bit

- Extended symbols encoding

Symbol	4B	5B	Description
Q	-NONE-	00000	Quiet (signal lost)
I	-NONE-	11111	Idle
J	-NONE-	11000	Start #1
K	-NONE-	10001	Start #2
T	-NONE-	01101	End
R	-NONE-	00111	Reset
S	-NONE-	11001	Set
H	-NONE-	00100	Halt

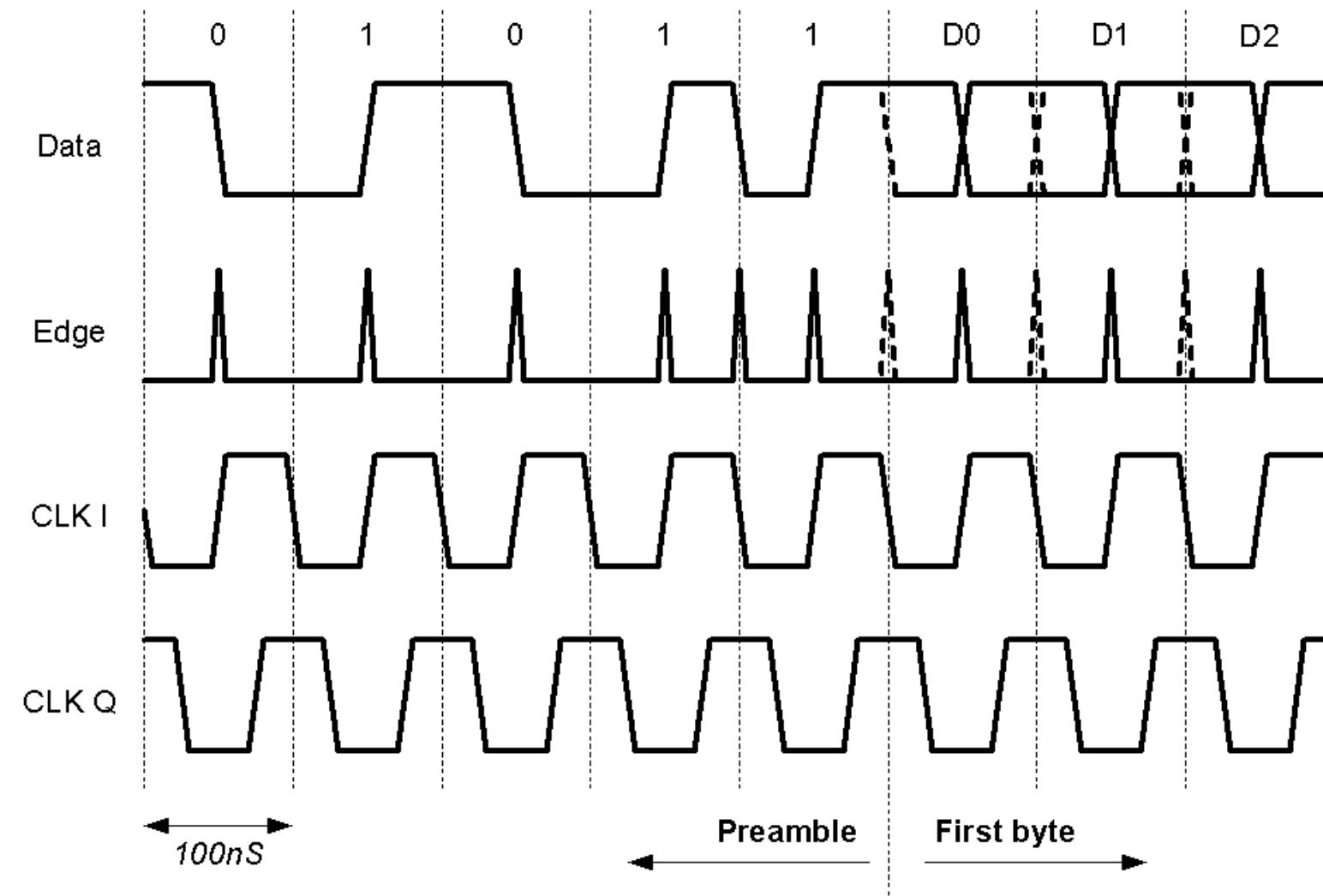
# Manchester receiver

## ■ Transceiver



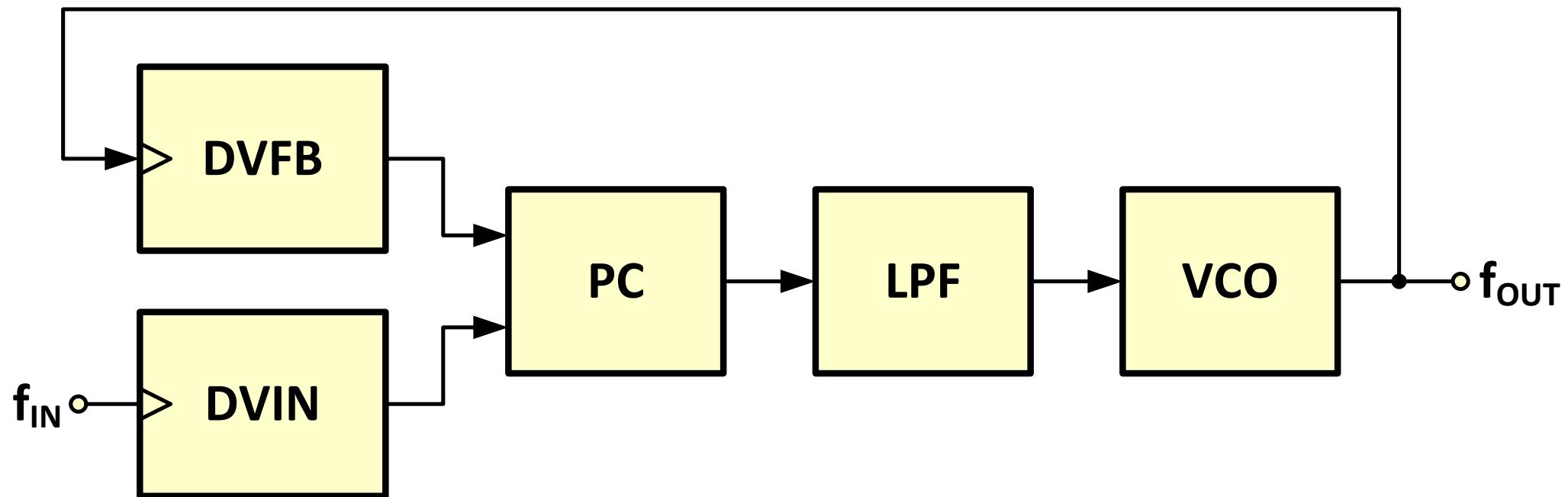
# Manchester receiver

- Signals timing



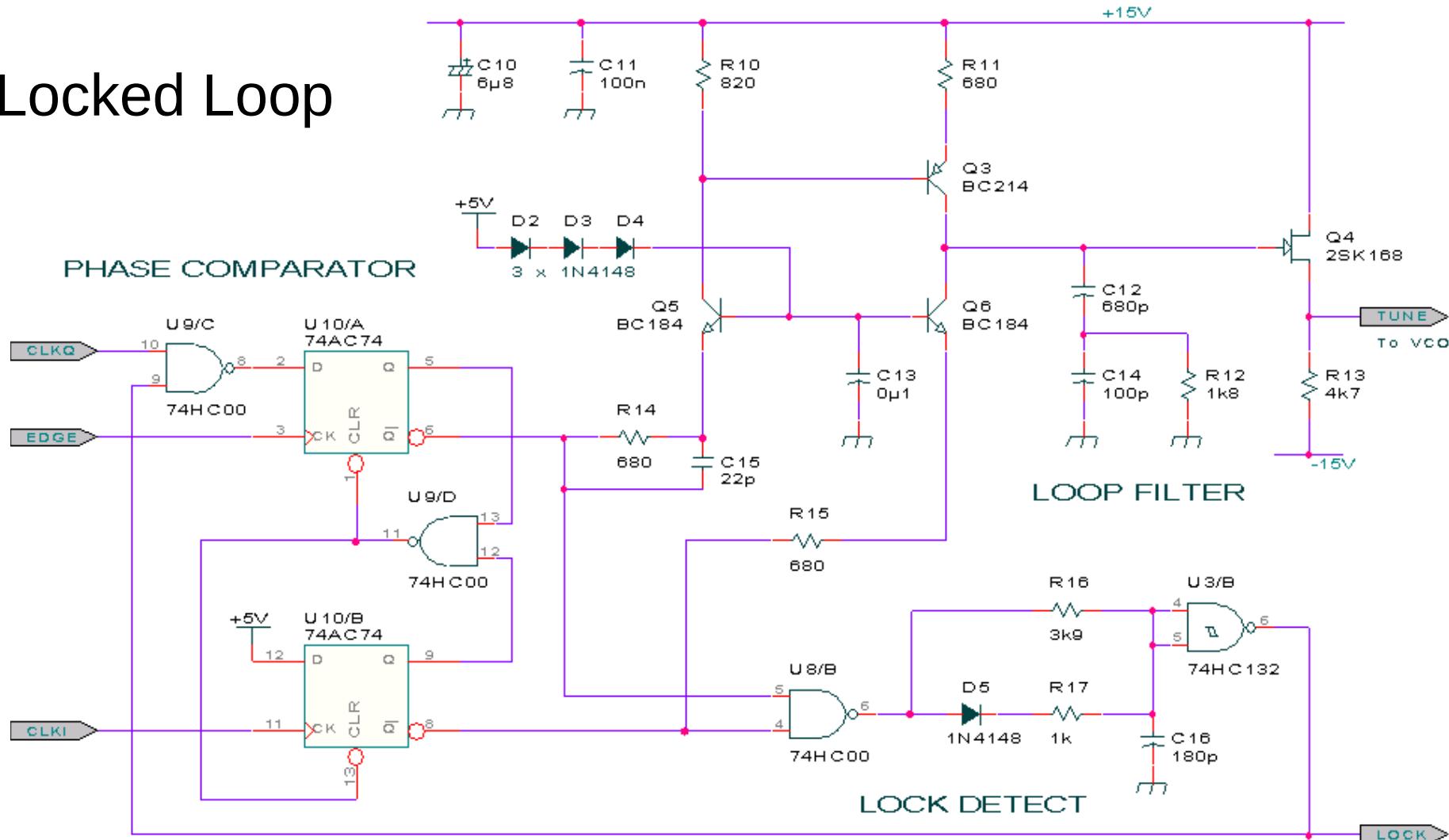
# Phase Locked Loop

- Operation principles of PLL



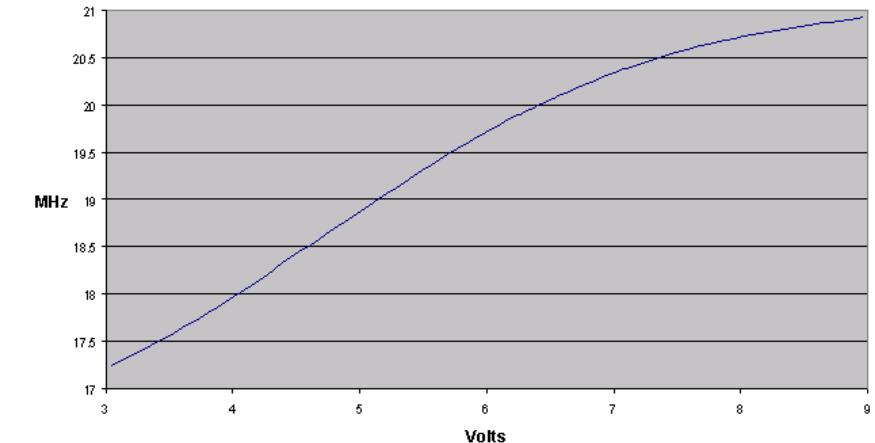
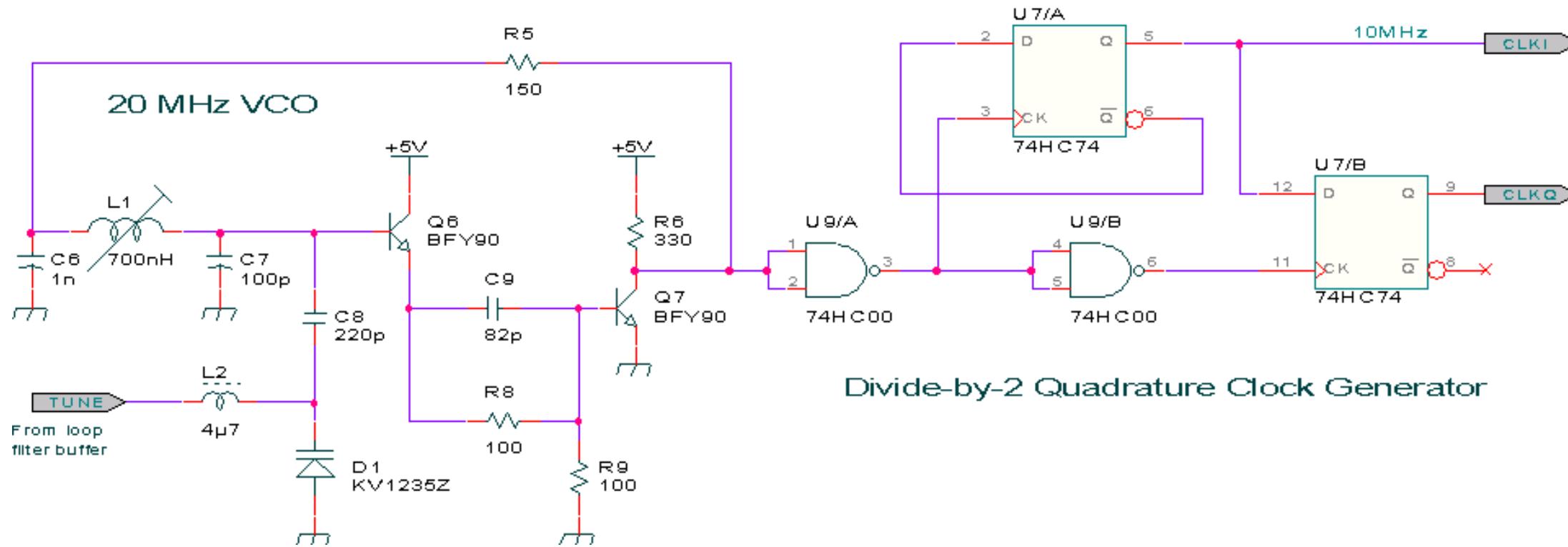
# Manchester receiver

- Phase Locked Loop



# Manchester receiver

- VCO generator



# Receiver Synchronization

- Oscilograms of FDP output after Phase Comparator and Charge Pump
- Clock signal synchronization by SYNC preamble

