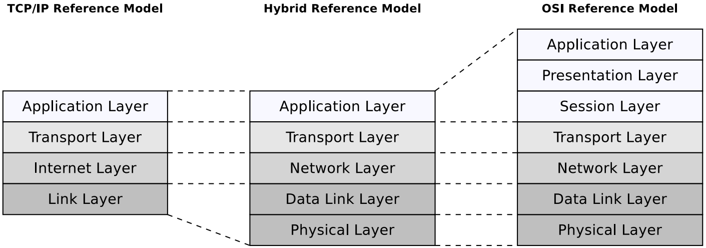
# 3rd Slide Set
# Computer Networks

Prof. Dr. Christian Baun

Frankfurt University of Applied Sciences
(1971–2014: Fachhochschule Frankfurt am Main)
Faculty of Computer Science and Engineering
christianbaun@fb2.fra-uas.de

# Physical Layer

- Functions of the Physical Layer
  - Bit transmission on wired or wireless transmission paths
  - Provides network technologies (e.g. Ethernet) and transmission media
  - Frames from the Data Link Layer are encoded with line codes into signals




Exercise sheet 2 repeats the contents of this slide set which are relevant for these learning objectives

- Devices: Repeater, Hub (Multiport Repeater)
- Protocols: Ethernet, Token Ring, WLAN, Bluetooth,. . .

# Learning Objectives of this Slide Set

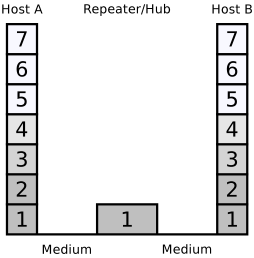
- Physical layer (part 2)
  - Devices of the Physical Layer
    - Repeaters and Hubs
  - Impact on the collision domain
  - Encoding data with line codes
    - Non-Return-to-Zero (NRZ)
    - Non-Return-to-Zero, Inverted (NRZI)
    - Multilevel Transmission Encoding - 3 Levels (MLT-3)
    - Return-to-Zero (RZ)
    - Unipolar RZ encoding
    - Alternate Mark Inversion (AMI code) = Bipolar encoding
    - B8ZS
    - Manchester code
    - Manchester II code
    - Differential Manchester encoding
    - 4B5B
    - 6B6B
    - 8B10B
    - 8B6T

# Repeater

Image Source: ASSMANN Electronic GmbH

- Because for all transmission media, the problem of **attenuation** (signal weakening) exists, the maximum range is limited
- **Repeaters** increase the range of a LAN by amplifying received electrical or optical signals and cleaning them from the noise and from Jitter
  - Jitter = deviation of the transmission timing
- Repeaters just forward signals
  - They do not analyze their meaning or correctness
- Repeaters have only 2 interfaces (*ports*)

Repeaters with just 2 interfaces are often called **Media Converter**




# Hub (Multiport-Repeater)

Image Source (Repeater): Perle Systems

- **Hubs** are Repeaters with $>$ 2 interfaces
- Forwards all incoming signals to all its output ports
- Repeaters and Hubs have no physical or logical network addresses
  - Reason: They just forward the received signals
    - They operate transparent and communicate only on Physical Layer


(Repeater)


(Hub)

# Topology of Hubs

- **Physical topology**: Star network because of the cabling
- **Logical topology**: Bus network, because equal to a long cable, where all network devices are connected with, a Hub forwards incoming signals to all other interfaces
  - For this reason, each terminal device, which is connected to a Hub, can receive and analyze the entire traffic, passing the Hub
- Advantages of Hubs over the physical bus network topology:
  - Better reliability, because the failure of individual cable segments does not result in a complete network failure
  - Adding or removing network devices does not cause network interruptions
- All nodes in the network that are connected to a Hub, are located in the same **collision domain**

# Collision Domain

- The collision domain is a network or a section of a network where multiple network devices use a shared transmission medium
  - It includes all network devices which compete for accessing a shared transmission medium
- Procedures for handling collisions:
  - **C**arrier **S**ense **M**ultiple **A**ccess/**C**ollision **D**etection
    - Collision detection
    - Ethernet
  - **C**arrier **S**ense **M**ultiple **A**ccess/**C**ollision **A**voidance
    - Collision avoidance
    - WLAN

The media access protocols are part of the Data Link Layer ($\Longrightarrow$ slide set 6)

# Collision Domain – Repeater and Hubs

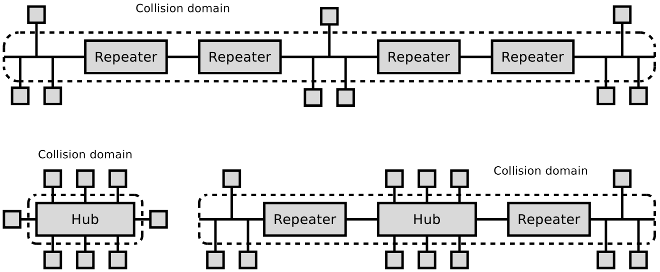
- Repeaters and Hubs increase the collision domain
  - Reason: These devices can not analyze signals
    - They only forward signals
- **Repeater**
  - In a network with CSMA/CD, all segments connected with Repeaters belong to the same collision domain
- **Hubs**
  - All ports (and thus all computers that are connected to a Hub) belong in a network with CSMA/CD to the same collision domain

**With a growing number of network devices, the number of collisions rises**

Beyond a certain number of network devices, no data transmissions are possible any more, because all transmissions are destroyed by collisions

# Collision Domains

- To make CSMA/CD work, collisions inside a collision domain must reach each network devices within a certain time.




- If the collision domain is too large, there is a risk that sending network devices do not detect collisions
  - Therefore, a maximum of 1023 devices per collision domain is allowed

For Thin (10BASE2) and Thick Ethernet (10BASE5), a maximum of 2 pairs of Repeaters are allowed between any 2 network devices

# Cascading Hubs

- Hubs can be cascaded to allow a greater network expansion
- But Hubs cannot be cascaded infinitely
  - The **round-trip time (RTT)** must not be exceeded
    - This is the period of time it takes for a Frame to be sent to the most distant point of the network plus the period of time it takes for an acknowledgment of that Frame to be received
    - The RTT depends on the data rate of the network
  - If the network becomes too large, the RTT will become too high
    - Then collisions occur more frequent and undetected collisions are possible

**5-4-3 rule** ⟸ applies only for Repeaters and Hubs!

- In a collision domain, 5 segments maximum can be connected
- For this, a maximum of 4 Repeaters are used
- Only at 3 segments, active senders (terminal devices) can be connected

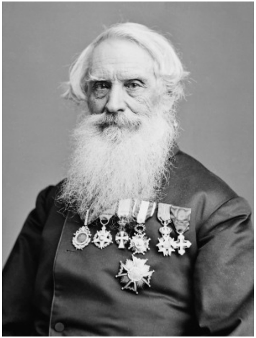
- For Gigabit Ethernet (and faster standards), no more Hubs/Repeaters are specified

# Encoding Data

Image source: Wikipedia (CC0)

- Efficient data encoding is important not only since the rise of computer networks
- An example for an efficient encoding is the **Morse Code**, invented by Samuel Morse from 1838

| | | | | | |
|---|---|---|---|---|---|
| A | · — | M | — — | Y | — · — — |
| B | — · · · | N | — · | Z | — — · · |
| C | — · — · | O | — — — | 1 | · — — — — |
| D | — · · | P | · — — · | 2 | · · — — — |
| E | · | Q | — — · — | 3 | · · · — — |
| F | · · — · | R | · — · | 4 | · · · · — |
| G | — — · | S | · · · | 5 | · · · · · |
| H | · · · · | T | — | 6 | — · · · · |
| I | · · | U | · · — | 7 | — — · · · |
| J | · — — — | V | · · · — | 8 | — — — · · |
| K | — · — | W | · — — | 9 | — — — — · |
| L | · — · · | X | — · · — | 0 | — — — — — |



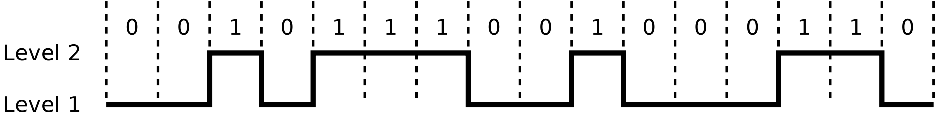
Samuel Morse (1791 – 1872)

# Encoding Data in Computer Networks

- The encoding is called **line code** in this context, and specifies how signals are transmitted on the transmission medium
- Specific signal sequences correspond with bit sequences in the data stream
- Computer networks must implement these operations:
  1. Conversion of binary data ($\Longrightarrow$ binary numbers) into signals (encoding)
  2. Transmission of signals from sender to receiver
  3. Conversion of the signals back into the binary data (decoding)
- Different ways exist to encode bits into signals
- The most simple way of representing logical 0 and 1 is by using different voltage levels
  - This line code is called **Non-Return-to-Zero** (NRZ)
    - Example: A logical 0 can be encoded by one signal level (e.g. 0 V) and a logical 1 by a different one (e.g. 5 V)

# Non-Return-to-Zero (NRZ)

- This line code encodes...
  - a logical 0 bit with physical signal level 1 (low value)
  - a logical 1 bit with physical signal level 2 (high value)




Implemented by the serial CAN (Controller Area Network) bus system, which was developed by Bosch in the 1980s for connecting control devices in cars

- When transmitting a long series of logical 0 bits or logical 1 bits, the physical signal level does not change
  - This results in 2 problems:
    1. **Baseline Wander**
    2. **Clock Recovery**

# Non-Return-to-Zero (NRZ) – Baseline Wander

- Problem: Shift of the average signal level (**Baseline Wander**) when using NRZ
- The receiver distinguishes the physical signal levels by using the average signal level of a certain number of received signals
  - Signals below the average signal level, interprets the receiver as logical 0 bit
  - Signals above the average signal level, interprets the receiver as logical 1 bit
- When transmitting long sequences of logical 0 or 1 bits, the average signal level may shift so much, making it difficult to detect a change of the physical signal

Sources

- Steve Zdancewic (2004). `http://www.cis.upenn.edu/~cse331/Fall04/Lectures/CSE331-3.pdf`
- Charles Spurgeon, Joann Zimmerman. *Ethernet: The Definitive Guide*. O'Reilly (2014)

Detailed source, which explains baseline wander from the electrical engineering perspective

- Maxim Integrated (2008). *NRZ Bandwidth – LF Cutoff and Baseline Wander*. `http://pdfserv.maximintegrated.com/en/an/AN1738.pdf`

# Avoid Baseline Wander

- In order to prevent **Baseline Wander**, when using a line code with 2 physical signal levels, the usage of both signal levels must be **distributed equally**
  - Therefore, the data to be transmitted must be encoded in a way, that the signal levels occur equally often
    - The data must be **scrambled**
- If a network technology uses 3 or 5 physical signal levels, the average signal level must match the middle signal level over the time

# Non-Return-to-Zero – Clock Recovery

- Problem: **Clock Recovery** when using NRZ
- Even if the processes for encoding and decoding run on different computers, they need to be controlled by the same clock

You can imagine the local clock as an internal signal, switching from low to high. A low/high pair is a clock cycle

- In each clock cycle, the sender transmits a bit and the receiver receives a bit
- If the clocks of sender and receiver drift apart, the receiver may lose count during a sequence of logical 0 bits or 1 bits

# Avoid the Problem of Clock Recovery

- One option: Using a **separate line, which transmits just the clock**

**A network technology with a separate signal line just for the clock is the serial bus system $I^2C$ (Inter-Integrated Circuit)**

But like comparable systems this bus system is only suited for local application and cannot be used to span large distances

- In computer networks, a separate signal line just for the clock is **not practical** because of the cabling effort
  - Instead, it is recommended to **increase the number of signal level changes** to enable the clock recovery from the data stream

**The next slides present several line codes, which all. . .**

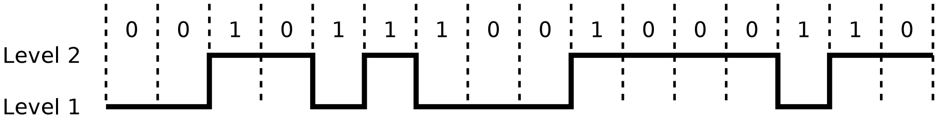
- (more or less successful) try to solve the challenges of baseline wander and/or clock recovery
- must consider the limitations of the transmission medium used
  - Fiber-optic cables and wireless transmissions via infrared and laser provide just 2 physical signal levels
  - Copper cables and wireless transmissions via radio waves provide $\geq 2$ physical signal levels

# I will not discuss all Lines Codes during Class!


- It makes no sense to show you all these line codes during class
  - It is not the best way to memorize the most important information about line codes
  - And it is boring
- Best practice is to do the line code related exercises from exercise sheet 2 by using the course material
  - I will assist you during the exercise sessions

# Non-Return-to-Zero, Inverted (NRZI)

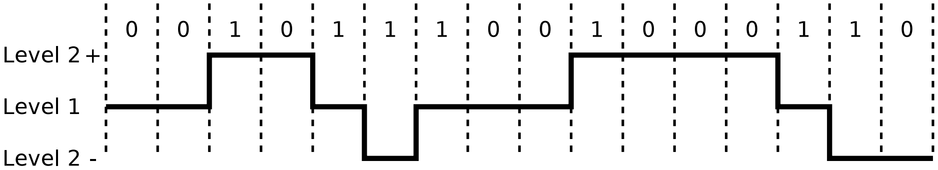
- Transmit a logical 1 bit $\Longrightarrow$ signal level change at the beginning of the clock
- Transmit a logical 0 bit $\Longrightarrow$ signal level remains unchanged for an entire clock
- **Clock recovery is impossible** for series of logical 0 bits
- The usage of the signal levels is not equally distributed
  - Therefore, **baseline wander can occur**




Implemented by Ethernet 100BASE-FX (Multi-mode fiber) and FDDI

# Multilevel Transmission Encoding - 3 Levels (MLT-3)

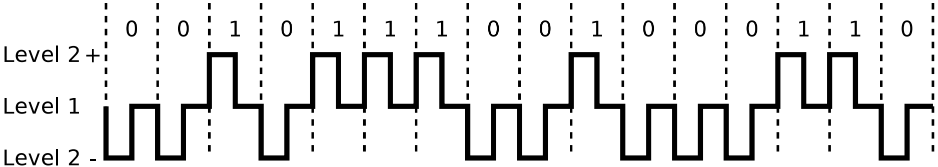
- This line code uses 3 signal levels +, 0 and −
  - If a logical 0 bit is transmitted, no signal level change takes place
  - A logical 1 bit is alternating encoded, according to the sequence [+, 0, −, 0]
- Just as for NRZI, the **clock recovery problem exists** with series of logical 0 bits and **baseline wander can occur**




Implemented by Ethernet 100BASE-TX

# Return-to-Zero (RZ)

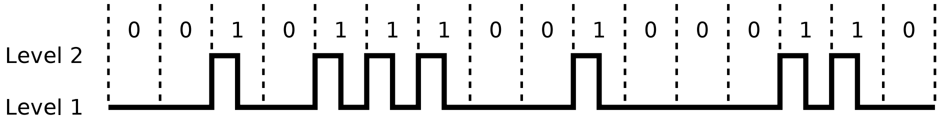
- RZ uses 3 signal levels too
  - Transmit a logical 1 bit ⟹ high signal level is transmitted for a half clock and then the signal level returns to the middle signal level
  - Transmit a logical 0 bit ⟹ low signal level is transmitted for a half clock and then the signal level returns to the middle signal level




- Advantage: Each transmitted bit causes a signal level change
  - Enables the receiver to do the **clock recovery** (synchronization)
- Drawbacks:
  - Requires **double as much bandwidth** compared with NRZ
  - **Baseline wander can occur** for series of logical 0 bits or 1 bits

# Unipolar RZ Encoding

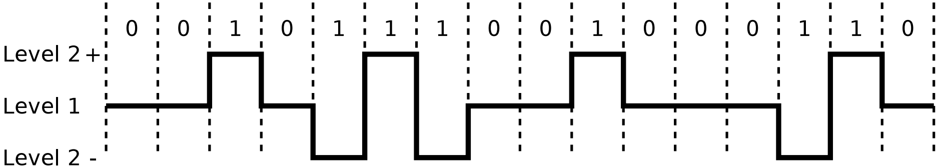
- Special form of Return-to-Zero (RZ)
  - Uses only 2 signal levels
    - Logical 0 bits are encoded as low signal level
    - Transmit a logical 1 bit $\Longrightarrow$ high signal level is transmitted for a half clock and then the signal level returns to the low signal level
- **Clock recovery is impossible** for series of logical 0 bits
- The usage of the different signal level is not equally distributed
  - Therefore **baseline wander can occur**




This line code is used for optical wireless data transmission via IrDA in the transmission mode SIR

# Alternate Mark Inversion (AMI code) = Bipolar Encoding

- Uses 3 signal levels (+, 0 und −)
  - Logical 0 bits are encoded as middle signal level (0)
  - Logical 1 bits are alternating encoded as high (+) or low signal level (−)
- Benefit: **Baseline wander cannot occur**
- Drawback: **Clock recovery is impossible** for series of logical 0 bits
- Error detection is partly possible because the signal sequences ++, −−, +0+ and −0− are illegal



# AMI Line Code in Practice and Scramblers

**The ISDN $S_0$ bus uses a modified version of the AMI line code**

- With this variant, logical 1 bits are encoded as middle signal level and logical 0 bits are alternating encoded as high signal level or low signal level

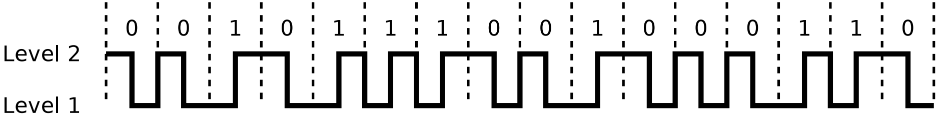
- When the AMI line code is used, clock recovery is impossible for the receiver, if series of logical 0 bits are transmitted
  - For this reason, a **scrambler** is often used, after AMI line code encoding
    - A scrambler is a device, which modifies a bit stream according to a simple algorithm in a way, that it is simple to reverse back to the original bit stream
  - In this case, scramblers are used, to interrupt long series of logical 0 bits
    - This makes the clock recovery for the receiver possible

# Bipolar With 8 Zeros Substitution (B8ZS)

- To avoid problems with long series of logic 0 bits, in practice, a slightly modified version of the AMI line code is used
  $\Longrightarrow$ **B8ZS**
- B8ZS prevents a loss of synchronization for longer series logical 0 bits by implementing 2 **modification rules** for sequences of 8 logical 0 bits
  - +00000000 is encoded as: +000+−0−+
  - −00000000 is encoded as: −000−+0+−
- In fact, both substitution rules are **code violations**
  - In both substitution rules, 2 positive and negative signal levels occur, one after another
    - This makes the substitutions for the receiver **recognizable**
- In contrast to AMI, **no scramblers are required**, when B8ZS is used
  - Reason: longer series of logical 0 bits are not a problem with B8ZS
- Just as with the AMI line code, **baseline wander cannot occur**

# Manchester Encoding (1/2)

- Uses 2 signal levels
  - A logical 1 bit is encoded with a rising edge
    - Change from signal level 1 (low value) to signal level 2 (high value)
  - A logical 0 bit is encoded with a falling edge
    - Change from signal level 2 (high value) to signal level 1 (low value)
- If 2 identical bits follow each other, at the end of the bit cell, the signal level changes to the initial level
  - Bit cell = time period, that is reserved for the transmission of a single bit




10 Mbps Ethernet (e.g. 10BASE2 and 10BASE-T) uses this line code

# Manchester Encoding (2/2)

- Advantages:
  - Signal level changes happen all the time to allow clock recovery
    $\Longrightarrow$ **clock recovery is no problem** for the receiver
  - The usage of the signal levels is equally distributed
    $\Longrightarrow$ **baseline wander cannot occur**
- Drawback: The transmission of a single bit requires on average 1.5 signal level changes

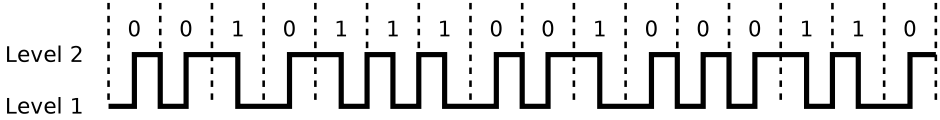
Because the number of level changes is a limiting factor of the transmission medium, modern network technologies don't use the Manchester encoding as line code

- For this line code, the bit rate is half the baud rate
  - Therefore, the efficiency of the line code is only 50 % compared to NRZ

- **Bit rate**: Transferred payload bits per time unit
- **Baud rate**: Transferred symbols per second

# Manchester II Encoding

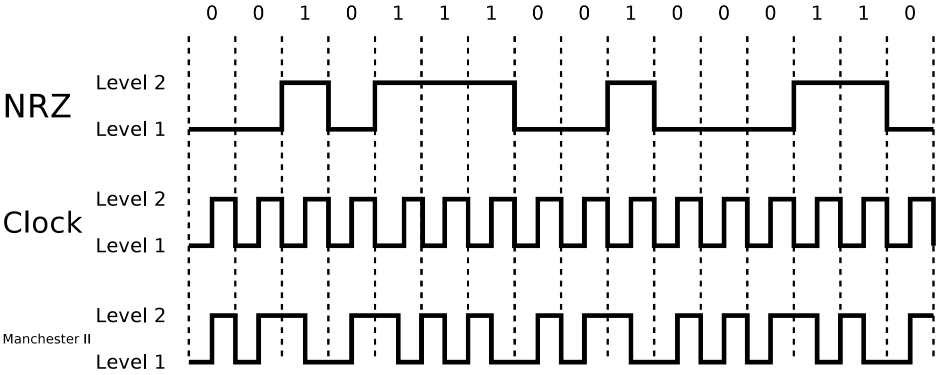
- This line code is the opposite of the Manchester encoding
  - Manchester encoding:
    - Transition from high to low signal corresponds to a logical 0 bit
    - Transition from low to high signal corresponds to a logical 1 bit
  - Manchester II encoding:
    - Transition from low to high signal corresponds to a logical 0 bit
    - Transition from high to low signal corresponds to a logical 1 bit
- Just as for the Manchester encoding, **clock recovery is possible** for the receiver and **baseline wander cannot occur** because the usage of the signal levels is distributed equally



# Manchester II Code

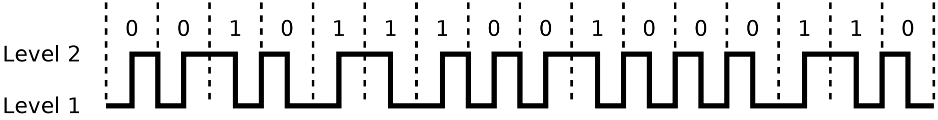
| A | B | A XOR B |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

- The Manchester II encoding is calculated via exclusive or (XOR) of the NRZ encoded data and the clock



# Differential Manchester Encoding

- Also called **Conditional DePhase encoding (CDP)**
  - Transmit a logical 1 bit $\Longrightarrow$ only in the middle of the bit cell changes the signal level
  - Transmit a logical 0 bit $\Longrightarrow$ a change of the signal level will take place at the beginning and in the middle of the bit cell
- In this variant of the Manchester encoding too,. . .
  - is **clock recovery possible** for the receiver and
  - **baseline wander cannot occur**
- Depending on the initial signal level, **2 signal sequences, inverse to each other, are possible**




Token Ring (IEEE 802.5) uses this line code

# Summary

- All line codes presented so far have drawbacks

1. **Baseline wander**
   - Problem with series of logical 0 bits and 1 bits when NRZ is used
   - Problem with series of logical 0 bits when NRZI, MLT-3 and Unipolar RZ are used
2. **Clock recovery**
   - Not guaranteed when NRZ, NRZI, MLT-3, Unipolar RZ and AMI are used
3. **Lack of efficiency**
   - With the variants of the Manchester encoding

# Possible Solution: Line Codes that encode Groups of Bits

- Modern network technologies encode the bit stream first with a line code that. . .
  - works **efficient**,
  - **ensures clock recovery** and
  - **avoids baseline wander**
- These encodings **improve** the bit stream in a way, that an encoding afterwards with the line codes NRZ, NRZI and MLT-3 does not result in any problems
- Examples of line codes, which improve the bit stream first, are 4B5B, 5B6B and 8B10B
  - These line codes encode fixed-size input blocks into fixed-size output blocks
- The objective is to achieve the positive characteristics of the Manchester encoding and a high efficiency at the same time

# 4B5B Encoding

- Maps groups of 4 payload bits onto groups of 5 code bits
  - With 5 bits, 32 different encodings are possible
    - Only 16 encodings are used for data (0–9 and A–F)
    - Some of the remaining 16 encodings are used for connection control
  - Because of the additional bit, added to each group of 4 bits payload, the output is increased by factor 5/4
    - Efficiency of the 4B5B encoding: 80%
  - Each 5-bit encoding has a maximum of a single leading 0 bit and in the output data stream, a maximum of three 0 bits in a row
    - Therefore, **clock recovery** for the receiver is **possible**
- After the encoding with 4B5B, **another encoding** e.g. with NRZI or MLT-3 takes place
  - If 4B5B is combined with NRZI (for 2 signal levels) or with MLT-3 (for 3 signal levels), **baseline wander cannot occur**

- Ethernet 100BASE-TX: After 4B5B, a further encoding with MLT-3 takes place
- FDDI and Ethernet 100BASE-FX: After 4B5B, a further encoding with NRZI takes place

# 4B5B Encoding (Table)

| Label | 4B | 5B | Function |
|---|---|---|---|
| 0 | 0000 | 11110 | 0 hexadecimal (Payload) |
| 1 | 0001 | 01001 | 1 hexadecimal (Payload) |
| 2 | 0010 | 10100 | 2 hexadecimal (Payload) |
| 3 | 0011 | 10101 | 3 hexadecimal (Payload) |
| 4 | 0100 | 01010 | 4 hexadecimal (Payload) |
| 5 | 0101 | 01011 | 5 hexadecimal (Payload) |
| 6 | 0110 | 01110 | 6 hexadecimal (Payload) |
| 7 | 0111 | 01111 | 7 hexadecimal (Payload) |
| 8 | 1000 | 10010 | 8 hexadecimal (Payload) |
| 9 | 1001 | 10011 | 9 hexadecimal (Payload) |
| A | 1010 | 10110 | A hexadecimal (Payload) |
| B | 1011 | 10111 | B hexadecimal (Payload) |
| C | 1100 | 11010 | C hexadecimal (Payload) |
| D | 1101 | 11011 | D hexadecimal (Payload) |
| E | 1110 | 11100 | E hexadecimal (Payload) |
| F | 1111 | 11101 | F hexadecimal (Payload) |
| Q | — | 00000 | Quiet (the line is gone dead) $\Longrightarrow$ Signal loss |
| I | — | 11111 | Idle (the line is idle) $\Longrightarrow$ Pause |
| J | — | 11000 | Start (Part 1) |
| K | — | 10001 | Start (Part 2) |
| T | — | 01101 | Stop (Part 1) |
| R | — | 00111 | Stop (Part 2) $\Longrightarrow$ Reset |
| S | — | 11001 | Set |
| H | — | 00100 | Halt (transmission failure) |

- The missing 5-bit combinations are invalid because they contain more than a single leading 0 bits or more than two 0 bits in a row

If Fast Ethernet 100BASE-TX is used, frames begin with JK and end with TR

# 5B6B Encoding (1/2)

- Maps groups of 5 payload bits onto groups of 6 code bits
  - Of the 32 possible 5-bit words, 20 are mapped to 6-bit words that contain an equal number of 1 bits and 0 bits
    $\Longrightarrow$ **neutral inequality** (*balanced*)
  - For the remaining twelve 5-bit words, a variant with two 1 bits and four 0 bits and a variant with four 1 bits and two 0 bits exist
    $\Longrightarrow$ **positive or negative inequality** (*unbalanced*)
- As soon as the first 5-bit word without neutral inequality needs to be encoded, the variant with the positive inequality is used
  - For encoding the next 5-bit word without neutral inequality, the variant with the negative inequality is used
    - The variants with positive or negative inequality alternate

# 5B6B Encoding (2/2)

- After the encoding with 5B6B, another encoding with NRZ takes place
  - This is possible, because if 5B6B is used, **clock recovery is possible** for the receiver and **baseline wander cannot occur**
- Advantage compared to the Manchester encoding: higher baud rate
  - Efficiency: $5/6 = 83.\overline{3}\%$

5B6B is used by Fast Ethernet 100Base-VG

# 5B6B Encoding (Table)

| 5B | 6B neutral | 6B positive | 6B negative | 5B | 6B neutral | 6B positive | 6B negative |
|---|---|---|---|---|---|---|---|
| 00000 | | 001100 | 110011 | 10000 | | 000101 | 111010 |
| 00001 | 101100 | | | 10001 | 100101 | | |
| 00010 | | 100010 | 101110 | 10010 | | 001001 | 110110 |
| 00011 | 001101 | | | 10011 | 010110 | | |
| 00100 | | 001010 | 110101 | 10100 | 111000 | | |
| 00101 | 010101 | | | 10101 | | 011000 | 100111 |
| 00110 | 001110 | | | 10110 | 011001 | | |
| 00111 | 001011 | | | 10111 | | 100001 | 011110 |
| 01000 | 000111 | | | 11000 | 110001 | | |
| 01001 | 100011 | | | 11001 | 101010 | | |
| 01010 | 100110 | | | 11010 | | 010100 | 101011 |
| 01011 | | 000110 | 111001 | 11011 | 110100 | | |
| 01100 | | 101000 | 010111 | 11100 | 011100 | | |
| 01101 | 011010 | | | 11101 | 010011 | | |
| 01110 | | 100100 | 011011 | 11110 | | 010010 | 101101 |
| 01111 | 101001 | | | 11111 | 110010 | | |

# 8B10B Encoding

- Maps groups of 8 payload bits onto groups of 10 code bits
  - Thus, the efficiency is 80%
- Each 8B10B encoding is composed in a way, that in the groups of 10 code bits either. . .
  - Five 0 bits and five 1 bits occur $\Longrightarrow$ neutral inequality
  - Six 0 bits and four 1 bits occur $\Longrightarrow$ positive inequality
  - Four 0 bits and six 1 bits occur $\Longrightarrow$ negative inequality
- After the encoding with 8B10B, another encoding via NRZ takes place
  - **Baseline wander cannot occur**, because some of the $2^8 = 256$ possible 8-bit words can be encoded in 2 different ways
    - This way, inequalities are compensated
- Each 10-bit encoding contains at least 3 signal level changes and at the latest after 5 clock cycles, the signal level changes
  - This **enables** the receiver **to do clock recovery**

Used by Gigabit-Ethernet 1000Base-CX, -SX, -LX, FibreChannel, InfiniBand, DisplayPort, FireWire 800 (IEEE 1394b) and USB 3.0

# 8B6T Encoding

- 8B6T = 8 Binary 6 Ternary
  - Useful for network technologies, that use > **2 signal levels**
- This line code encodes 8-bit blocks as groups of 6 symbols, where each one can represent the state −, 0 or +
  - The symbols of the states represent electrical signal levels
- The encoding is carried out by using a table, which contains all $2^8 = 256$ possible 8-bit combinations
  - The table shows, that the output of 8B6T makes **baseline wander impossible**, and the frequent signal level changes make **clock recovery possible** for the receiver
- In contrast to 4B5B, 5B6B and 8B10B, which only *improve* the payload and require an encoding with NRZ(I) or MLT-3 afterwards, 8B6T encoded data **can be used directly for transmission**

Fast-Ethernet 100BASE-T4 uses this line code

# 8B6T Encoding (Table)

| 8-bit sequence | 8B6T code | 8-bit sequence | 8B6T code | 8-bit sequence | 8B6T code |
|---|---|---|---|---|---|
| 00 | +−00+− | 10 | +0+−−0 | 20 | 00−++− |
| 01 | 0+−+−0 | 11 | ++0−0− | 21 | −−+00+ |
| 02 | +−0+−0 | 12 | +0+−0− | 22 | ++−0+− |
| 03 | −0++−0 | 13 | 0++−0− | 23 | ++−0−+ |
| 04 | −0+0+− | 14 | 0++−−0 | 24 | 00+0−+ |
| 05 | 0+−−0+ | 15 | ++00−− | 25 | 00+0+− |
| 06 | +−0−0+ | 16 | +0+0−− | 26 | 00−00+ |
| 07 | −0+−0+ | 17 | 0++0−− | 27 | −−+++− |
| 08 | −+00+− | 18 | 0+−0+− | 28 | −0−++0 |
| 09 | 0−++−0 | 19 | 0+−0−+ | 29 | −−0+0+ |
| 0A | −+0+−0 | 1A | 0+−++− | 2A | −0−+0+ |
| 0B | +0−+−0 | 1B | 0+−00+ | 2B | 0−−+0+ |
| 0C | +0−0+− | 1C | 0−+00+ | 2C | 0−−++0 |
| 0D | 0−+−0+ | 1D | 0−+++− | 2D | −−00++ |
| 0E | −+0−0+ | 1E | 0−+0−+ | 2E | −0−0++ |
| 0F | +0−−0+ | 1F | 0−+0+− | 2F | 0−−0++ |

etc.

# Summary

| Line code | Signal levels | Baseline wander possible | Signal level change | Self-synchro-nizing[1] | Efficiency[2] | Directly transfer-able | Additional encoding |
|---|---|---|---|---|---|---|---|
| NRZ | 2 | yes | at changes | no | 100% | no | — |
| NRZI | 2 | yes | for 1-bits | no | 75% | no | — |
| MLT-3 | 3 | yes | for 1-bits | no | 100% | no | — |
| RZ | 3 | yes | always | yes | 50% | no | — |
| Unip. RZ | 2 | yes | for 1-bits | no | 75% | no | — |
| AMI | 3 | no | for 1-bits | no | 100% | no | Scrambler |
| B8ZS | 3 | no | for 1-bits | yes | 100% | yes | — |
| Manchester | 2 | no | always | yes | 50% | yes | — |
| Manchester II | 2 | no | always | yes | 50% | yes | — |
| Diff. Manch. | 2 | yes | always | yes | 50% | yes | — |
| 4B5B | 2 | yes | — | yes | 80% | no | NRZI or MLT-3 |
| 5B6B | 2 | no | — | yes | $83.\overline{3}\%$ | no | NRZ |
| 8B10B | 2 | no | — | yes | 80% | no | NRZ |
| 8B6T | 3 | no | — | yes | 100% | yes | — |

[1] Specifies if the clock recovery is possible with this line code.
[2] Ratio of bit rate (payload in bits per time) and baud rate (signal changes per second).