

## 3. Foliensatz Computernetze

Prof. Dr. Christian Baun

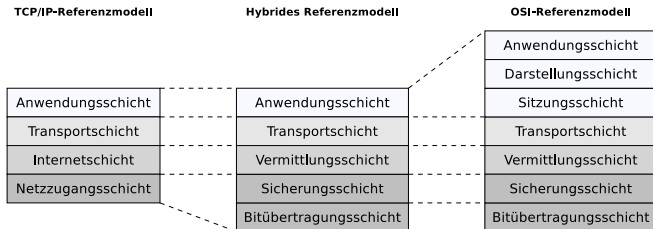
Frankfurt University of Applied Sciences  
(1971–2014: Fachhochschule Frankfurt am Main)  
Fachbereich Informatik und Ingenieurwissenschaften  
[christianbaun@fb2.fra-uas.de](mailto:christianbaun@fb2.fra-uas.de)

# Lernziele dieses Foliensatzes

- Bitübertragungsschicht (Teil 2)
  - Geräte der Bitübertragungsschicht
    - Repeater und Hubs
  - Auswirkungen auf die Kollisionsdomäne
  - Kodierung von Daten mit Leitungscodes
    - Non-Return to Zero (NRZ)
    - Non-Return to Zero Invert (NRZI)
    - Multilevel Transmission Encoding - 3 Levels (MLT-3)
    - Return-to-Zero (RZ)
    - Unipolare RZ-Kodierung
    - Alternate Mark Inversion (AMI-Kodierung)
    - B8ZS
    - Manchesterkodierung
    - Manchester II-Kodierung
    - Differentielle Manchesterkodierung
    - 4B5B
    - 6B6B
    - 8B10B
    - 8B6T

# Bitübertragungsschicht

- Aufgaben der Bitübertragungsschicht (Physical Layer):
  - Bitübertragung auf leitungsgebundenen oder leitungslosen Übertragungsstrecken
  - Bereitstellung von Vernetzungstechnologien
  - Übertragungsmedien
  - Rahmen der Sicherungsschicht werden mit Leitungscode in Signale kodiert



- Geräte: Repeater, Hub (Multiport-Repeater)
- Protokolle: Ethernet, Token Ring, WLAN, Bluetooth,...

Bildquelle: StarTech

- 
- Host A                      Repeater/Hub                      Host B
- |   |   |   |
|---|---|---|
| 7 |   | 7 |
| 6 |   | 6 |
| 5 |   | 5 |
| 4 |   | 4 |
| 3 |   | 3 |
| 2 |   | 2 |
| 1 | 1 | 1 |
- Medium                      Medium



Bildquelle: [www.planet.com.tw](http://www.planet.com.tw)

- 

5/41

# Topologie bei Hubs

- **Physische Topologie:** Stern-Topologie wegen der Verkabelung
- **Logische Topologie:** Bus-Topologie, weil genau wie bei einem langen Kabel, an dem alle Netzwerkgeräte hängen, leitet ein Hub ankommende Signale zu allen anderen Schnittstellen weiter
  - Darum kann jedes Endgerät, das mit einem Hub verbunden ist, den kompletten Datenverkehr, der über den Hub geht, empfangen und analysieren
- Vorteile von Hubs gegenüber der physischen Bus-Topologie:
  - Bessere Ausfallsicherheit, denn der Ausfall einzelner Kabelsegmente legt beim Hub nicht das komplette Netz lahm
  - Beim Hinzufügen oder Entfernen von Netzwerkgeräten wird das Netz nicht unterbrochen
- Alle Knoten, die an einen Hub angeschlossen sind, befinden sich in einer **Kollisionsdomäne**

## Kollisionsdomäne (Kollisionsgemeinschaft)

- Die Kollisionsdomäne ist ein Netzwerk oder Teil eines Netzwerks, in dem mehrere Netzwerkgeräte ein gemeinsames Übertragungsmedium nutzen
  - Sie umfasst alle Netzwerkgeräte, die gemeinsam um den Zugriff auf ein Übertragungsmedium konkurrieren
- Behandlung von Kollisionen:
  - **Carrier Sense Multiple Access/Collision Detection**
    - Kollisionserkennung
    - Ethernet
  - **Carrier Sense Multiple Access/Collision Avoidance**
    - Kollisionsvermeidung
    - WLAN

Die Medienzugriffsverfahren sind Teil der Sicherungsschicht (⇒ Foliensatz 6)

## Kollisionsdomäne – Repeater und Hubs

- Repeater und Hubs vergrößern die Kollisionsdomäne
  - Grund: Diese Geräte können keine Signale analysieren, sondern nur weiterleiten
- **Repeater**
  - In einem CSMA/CD-Netz (Ethernet) gehören alle mit Repeatern verbundenen Segmente zu einer Kollisionsdomäne
- **Hubs**
  - Alle Ports (und damit alle Rechner, die an einen Hub angeschlossenen sind) gehören in einem CSMA/CD-Netz zur gleichen Kollisionsdomäne

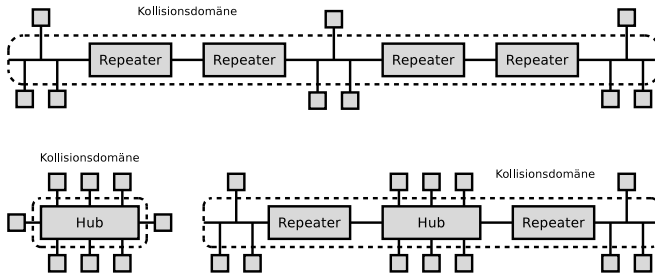
Mit steigender Anzahl an Netzwerkgeräten steigt die Anzahl der Kollisionen

Ab einer bestimmten Anzahl Netzwerkgeräte ist keine Datenübertragung mehr möglich, da alle Sendungen durch Kollisionen zerstört werden



# Kollisionsdomänen

- Damit CSMA/CD funktioniert, müssen Kollisionen innerhalb einer bestimmten Zeit jedes Netzwerkgerät einer Kollisionsdomäne erreichen



- Ist die Kollisionsdomäne zu groß, besteht die Gefahr, dass sendende Netzwerkgeräte Kollisionen nicht erkennen
  - Darum darf es maximal 1023 Geräte pro Kollisionsdomäne geben

Bei Thin Ethernet (10BASE2) und Thick Ethernet (10BASE5) sind max. 2 Repeater-Paare zwischen 2 beliebigen Netzwerkgeräten erlaubt

# Hubs kaskadieren

- Hubs kann man kaskadieren um die Netzausdehnung zu vergrößern
- Hubs können aber nicht beliebig kaskadiert werden!
  - Die **Round-Trip-Time (RTT)** darf nicht überschritten werden
    - RTT = Zeit, die ein Rahmen braucht, um vom Sender zum weitest entfernten Empfänger zu gelangen plus die Zeit, die die Bestätigung für den Empfang des Rahmens zum Sender braucht
    - Die RTT hängt von der Geschwindigkeit des Netzwerks ab
  - Wird das Netz zu groß, wird die RTT zu hoch
    - Dann werden Kollisionen häufiger und unerkannte Kollisionen möglich

## 5-4-3-Repeater-Regel $\Leftarrow$ gilt nur für Repeater und Hubs!

- Maximal 5 Kabelsegmente dürfen verbunden sein
  - Dafür werden maximal 4 Repeater eingesetzt
  - An nur 3 Segmenten dürfen Endstationen angeschlossen werden
- 
- Für Gigabit-Ethernet (und schneller) sind keine Hubs/Repeater mehr spezifiziert

# Kodierung von Daten

Bildquelle: Wikipedia

- Die effiziente Kodierung von Daten ist nicht erst seit dem Aufkommen von Computernetzen wichtig
- Ein Beispiel für eine effiziente Kodierung ist der **Morsekode** (Morsealphabet) nach Samuel Morse von 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	— — — — —



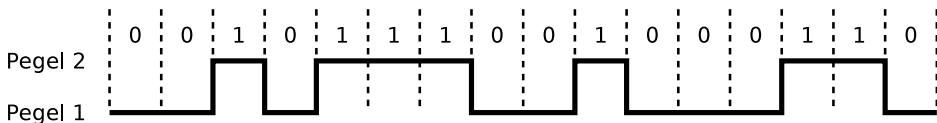
Samuel Morse (1791 – 1872)

# Kodierung von Daten in Netzwerken

- Die Kodierung, die in diesem Kontext **Leitungscode** heißt, legt in Computernetzen fest, wie Signale auf dem verwendeten Medium übertragen werden
- Bestimmte Signalpegelfolgen werden Bitfolgen im Datenstrom zugeordnet
- In Computernetzen sind folgende Aktionen nötig:
  - ① Umwandlung von Binärdaten bzw. Binärzahlen in Signale (Kodierung)
  - ② Übertragung der Signale vom Sender zum Empfänger
  - ③ Rückwandlung der Signale in Bits (Dekodierung)
- Die Kodierung von Bits in Signale ist auf verschiedene Arten möglich
- Die einfachste Form der Darstellung von logischer 0 und 1 ist mit verschiedenen Spannungsniveaus möglich
  - Dieser Leitungscode heißt **Non-Return to Zero (NRZ)**
    - Es kann z.B. eine 0 durch einen Signalpegel (z.B. 0 Volt) und eine 1 durch einen anderen Signalpegel (z.B. 5 Volt) kodiert werden

## Non-Return to Zero (NRZ)

- Heißt auch Non-Return to Zero-Level (NRZ-L), binäre Übertragung oder binäre Kodierung
- Dieser Leitungscode kodiert den Datenwert 0 mit Signalpegel 1 und den Datenwert 1 mit Signalpegel 2



Das serielle Bussystem CAN-Bus (Controller Area Network), das die Firma Bosch in den 1980er Jahren zur Vernetzung von Steuergeräten in Automobilen entwickelt hat, verwendet NRZ

- Beim Übertragen einer längeren Serie von Nullen oder einer Serie von Einsen gibt es keine Pegeländerung
  - Das führt zu 2 Problemen
    - 1 Verschiebung des Durchschnitts (**Baseline Wander**)
    - 2 Taktwiederherstellung (**Clock Recovery**)

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

- Problem: Verschiebung des Durchschnitts (**Baseline Wander**) bei NRZ
- Der Empfänger unterscheidet die Signalpegel anhand des Durchschnitts einer bestimmten Anzahl zuletzt empfangener Signale
  - Signale unter dem Durchschnitt interpretiert der Empfänger als 0
  - Signale über dem Durchschnitt interpretiert der Empfänger als 1
- Werden Serien von Nullen oder Einsen übertragen, kann sich der Durchschnitt soweit verschieben, dass eine Signaländerung schwierig zu erkennen ist

## Quellen

- Steve Zdancewic. 2004. <http://www.cis.upenn.edu/~cse331/Fall104/Lectures/CSE331-3.pdf>
- Charles Spurgeon, Joann Zimmerman. *Ethernet: The Definitive Guide*. O'Reilly (2014)

## Sehr ausführliche Quelle, die auch die elektrotechnischen Grundlagen behandelt

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

## Baseline Wander verhindern

- Zur Verhinderung der Durchschnittsverschiebung, muss im Leitungscode die Belegung der Signalpegel bei 2 Signalpegeln **gleichverteilt** sein
  - Die zu übertragenden Daten müssen also so kodiert werden, dass die Signalpegel gleich häufig vorkommen
    - Die Daten müssen **verwürfelt** (englisch: *scrambled*) werden
- Verwendet eine Netzwerktechnologie 3 oder 5 Signalpegel, muss der Durchschnitt über die Zeit dem mittleren Signalpegel entsprechen

## Non-Return to Zero (NRZ) – Taktwiederherstellung

- Problem: **Taktwiederherstellung (Clock Recovery)** bei NRZ
- Auch wenn die Prozesse für Kodierung und Dekodierung auf unterschiedlichen Rechnern laufen, müssen sie vom gleichen Takt gesteuert werden

Man kann sich den lokalen Takt als internes Signal vorstellen, das von low nach high wechselt. Ein low/high-Paar ist ein Taktzyklus

- In jedem Taktzyklus überträgt der Sender ein Bit und der Empfänger empfängt eins
- Driften die Taktgeber von Sender und Empfänger auseinander, könnte sich der Empfänger bei einer Folge von Nullen oder Einsen verzählen



## Problem der Taktwiederherstellung vermeiden

- Eine Möglichkeit: Eine **getrennte Leitung, die den Takt überträgt**

Eine Vernetzungstechnologie mit einer separaten Signalleitung nur für den Takt, ist das serielle Bussystem I<sup>2</sup>C (Inter-Integrated Circuit)

Dieses ist aber wie auch vergleichbare Bussysteme nur für lokale Anwendungen und nicht zur Überbrückung großer Distanzen geeignet

- In Computernetzen sind separate Signalleitungen nur für den Takt wegen des Verkabelungsaufwands **nicht praktikabel**
  - Stattdessen empfiehlt es sich, die **Anzahl der Signalpegelwechsel zu erhöhen**, um die Taktrückgewinnung aus dem Datenstrom zu ermöglichen

Die folgenden Folien präsentieren verschiedene Leitungscodes, die alle...

- (mehr oder weniger erfolgreich) versuchen die Probleme Baseline Wander und Clock Recovery zu lösen
- die Einschränkungen der jeweiligen Übertragungsmedien berücksichtigen müssen
  - Lichtwellenleiter und drahtlose Übertragungen via Infrarot und Laser ermöglichen nur 2 physische Signalpegel
  - Kupferkabel und drahtlose Übertragungen via Radiowellen ermöglichen  $\geq 2$  physische Signalpegel

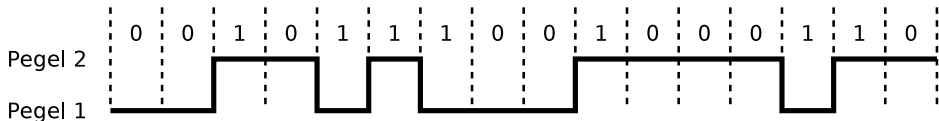
Auf keinen Fall bespreche ich alle Leitungscodes in der Vorlesung!



- Es macht keinen Sinn, alle Leitungscodes in der Vorlesung vorzustellen
  - Das wäre kein geeigneter Weg, damit Sie die wichtigsten Informationen über Leitungscodes lernen
  - Zudem ist es langweilig
- Die optimale Methode, sich mit den Leitungscodes auseinanderzusetzen ist die entsprechenden Übungsaufgaben von Übungsblatt 2 zu machen
  - Ich helfe Ihnen damit während der Übungsstunden

## Non-Return to Zero Invert (NRZI)

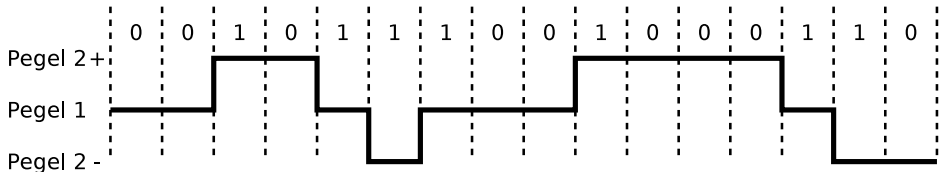
- Datenwert 1 senden  $\implies$  zu Beginn des Takts wechselt der Signalpegel
- Datenwert 0 senden  $\implies$  Signalpegel bleibt einen Takt lang unverändert
- **Keine Taktrückgewinnung** bei Reihen von Nullen
- Die Belegung der Signalpegel ist nicht gleichverteilt
  - Darum sind **Durchschnittsverschiebungen möglich**



Verwendet Ethernet 100BASE-FX (Multimode-Glasfaser) und FDDI

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

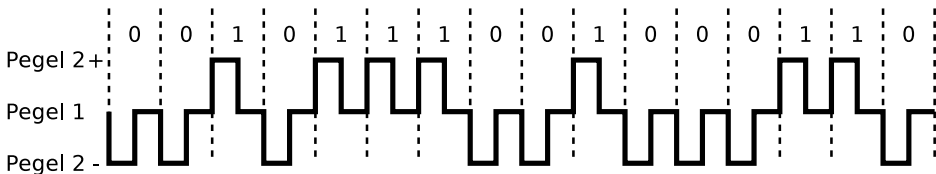
- Dieser Leitungscode verwendet die 3 Signalpegel +, 0 und -
  - Beim Senden des Datenwerts 0 findet kein Pegelwechsel statt
  - Datenwert 1 wird abwechselnd entsprechend der Folge [+ , 0 , - , 0] kodiert
- Genau wie bei NRZI besteht das **Problem der Taktrückgewinnung** bei Reihen von Nullen und **Durchschnittsverschiebungen sind möglich**



Verwendet Ethernet 100BASE-TX

## Return-to-Zero (RZ)

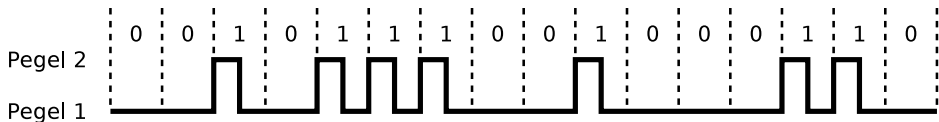
- Auch RZ verwendet 3 Signalpegel
  - Datenwert 1 senden  $\Rightarrow$  der positive Signalpegel wird für einen halben Takt übertragen und danach zum mittleren Signalpegel zurückgekehrt
  - Datenwert 0 senden  $\Rightarrow$  der negative Signalpegel wird für einen halben Takt übertragen und danach zum mittleren Signalpegel zurückgekehrt



- Vorteil: Änderung des Signalpegels bei jedem übertragenen Datenbit
  - Ermöglicht dem Empfänger die **Taktrückgewinnung** (Synchronisierung)
- Nachteile:
  - Benötigt im Vergleich zu NRZ **doppelt so viel Bandbreite**
  - Bei Reihen von Einsen oder Nullen sind **Durchschnittsverschiebungen möglich**

# Unipolare RZ-Kodierung

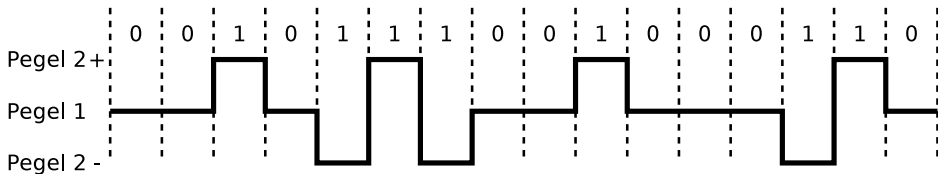
- Sonderform der RZ-Kodierung
  - Verwendet nur 2 Signalpegel
    - Datenwert 0 wird mit Signalpegel 1 kodiert
    - Datenwert 1 senden  $\implies$  der positive Signalpegel wird für einen halben Takt übertragen und danach zu Signalpegel 1 zurückgekehrt
- **Keine Taktrückgewinnung** bei Reihen von Nullen
- Die Belegung der Signalpegel ist nicht gleichverteilt
  - Somit sind **Durchschnittsverschiebungen möglich**



Dieser Leitungscode wird u.a. bei drahtlosen optischen Datenübertragungen via IrDA im Übertragungsmodus SIR verwendet

## AMI-Kodierung (AMI = Alternate Mark Inversion)

- Arbeitet mit 3 Pegeln (+, 0 und -)
  - Datenwert 0 wird als mittlerer Signalpegel (0) übertragen
  - Datenwert 1 wird abwechselnd als positiver (+) oder negativer Signalpegel (-) übertragen
- Vorteil: **Durchschnittsverschiebungen sind ausgeschlossen**
- Nachteil: **Keine Taktrückgewinnung** bei Reihen von Nullen
- Fehlererkennung teilweise möglich, da ++, --, +0+ und -0- nicht erlaubt



# AMI-Kodierung in der Praxis und Scrambler (Verwürfler)

## Der ISDN S<sub>0</sub>-Bus verwendet eine modifizierte Version der AMI-Kodierung

- Dabei wird der Datenwert 1 als mittlerer Signalpegel und der Datenwert 0 abwechselnd als positiver oder negativer Signalpegel übertragen
- Bei der AMI-Kodierung ist bei Serien aufeinanderfolgender Nullen die Taktrückgewinnung für den Empfänger unmöglich
  - Darum wird nach der AMI-Kodierung meistens ein **Scrambler** (Verwürfler) verwendet
    - Scrambler stellen ein Datensignal nach einem einfachen Algorithmus umkehrbar um
  - In diesem Fall kommen Scrambler zum Einsatz, die lange Ketten von Nullen unterbrechen um so die Taktrückgewinnung für den Empfänger zu ermöglichen

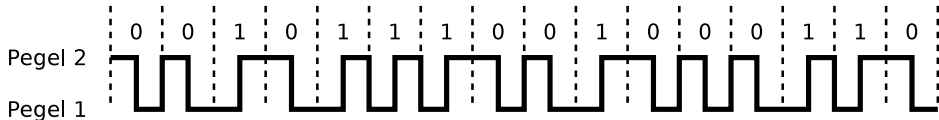


## Bipolar With 8 Zeros Substitution (B8ZS)

- Wegen des Problems bei längeren Serien von Nullen verwendet man in der Praxis eine geringfügig modifizierte Versionen der AMI-Kodierung  
⇒ **B8ZS**
- B8ZS verhindert einen Synchronisationsverlust bei längeren Serien von Nullen durch 2 **Regeln zur Modifikation** von Folgen von 8 Nullbits
  - +00000000 wird kodiert als: +000+-0-+
  - -00000000 wird kodiert als: -000-+0+-
- Eigentlich sind beide Ersetzungsregeln **Coderegelerletzungen**
  - Beide Ersetzungen enthalten jeweils 2 positive und negative Signalpegel nacheinander
    - Das macht die Ersetzungen für den Empfänger **erkennbar**
- Bei B8ZS sind im Gegensatz zu AMI **keine Scrambler nötig**
  - Grund: Längere Serien von Nullen sind bei B8ZS kein Problem
- **Durchschnittsverschiebungen sind** wie bei AMI **ausgeschlossen**

# Manchesterkodierung (1/2)

- Arbeitet mit 2 Pegeln
  - Datenwert 1 wird als steigende Flanke kodiert
    - Wechsel von Signalpegel 1 zu Signalpegel 2
  - Datenwert 0 wird als fallende Flanke kodiert
    - Wechsel von Signalpegel 2 zu Signalpegel 1
- Folgen 2 identische Bits aufeinander, wird am Ende der Bitzelle auf das Anfangsniveau zurückgesprungen
  - Eine Bitzelle ist der für die Übertragung eines Bits reservierte Zeitraum



10 MBit/s Ethernet (z.B. 10BASE2 und 10BASE-T) verwendet diesen Leitungscode

## Manchesterkodierung (2/2)

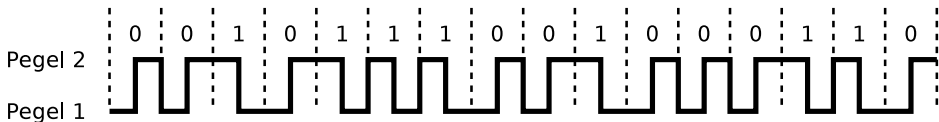
- Vorteile:
  - Es gibt stets Pegelwechsel zur Taktrückgewinnung  
⇒ **Taktrückgewinnung ist** für den Empfänger **möglich**
  - Die Belegung der Signalpegel ist gleichverteilt  
⇒ Der **Durchschnitt kann sich nicht verschieben**
- Nachteil: 1 Bit übertragen erfordert im Schnitt 1,5 Wechsel des Signalpegels

Da die Anzahl der Pegelwechsel ein limitierender Faktor des Übertragungsmediums ist, ziehen moderne Netzwerktechnologien andere Leitungscodes der Manchesterkodierung vor

- Bei diesem Leitungscode entspricht die Bitrate der halben Baudrate
  - Also ist die Effizienz des Leitungscodes nur 50% im Vergleich zu NRZ
- **Bitrate:** Anzahl der Nutzdaten (in Bits) pro Zeit
- **Baudrate:** Rate, in der sich Signale ändern pro Zeit

# Manchester II-Kodierung

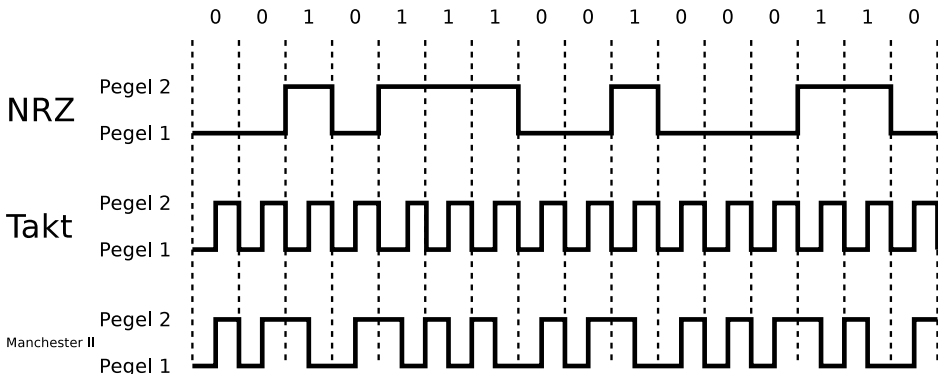
- Dieser Leitungscode ist das Gegenteil der Manchesterkodierung
  - Manchesterkodierung:
    - Sprung von 1 nach 0 entspricht einer 0
    - Sprung von 0 nach 1 entspricht einer 1
  - Manchester II-Kodierung:
    - Sprung von 0 nach 1 entspricht einer 0
    - Sprung von 1 nach 0 entspricht einer 1
- Wie bei Manchester ist die **Taktrückgewinnung** für den Empfänger **möglich** und die **Belegung der Signalpegel gleichverteilt**



# Manchester II-Kodierung

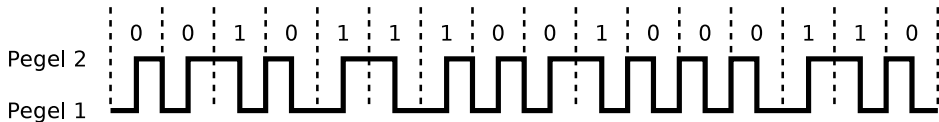
A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0

- Die Manchester II-Kodierung ist das exklusive Oder (XOR) der NRZ-kodierten Daten und des Takts



# Differentielle Manchesterkodierung

- Heißt auch **Conditional DePhase Encoding (CDP)**
  - Datenwert 1 kodieren  $\implies$  nur in der Mitte der Bitzelle wechselt der Signalpegel
  - Datenwert 0 kodieren  $\implies$  am Anfang und in der Mitte der Bitzelle wechselt der Signalpegel
- Auch bei dieser Variante der Manchesterkodierung...
  - ist die **Taktrückgewinnung** für den Empfänger **möglich** und
  - der **Durchschnitt kann sich nicht verschieben**
- Abhängig vom Anfangspegel ergeben sich **2 mögliche, zueinander inverse Signalfolgen**



Diesen Leitungscode verwendet Token Ring (IEEE 802.5)

# Zusammenfassung

- Alle bislang vorgestellten Kodierungen haben Nachteile

## ① Verschiebung des Durchschnitts (Baseline Wander)

- Problem aufeinanderfolgender Nullen und Einsen bei NRZ
- Problem aufeinanderfolgender Nullen bei NRZI, MLT-3, Unipolarem RZ und AMI

## ② Taktwiederherstellung (Clock Recovery)

- Kann bei NRZ, NRZI, MLT-3, Unipolarem RZ und AMI nicht garantiert werden

## ③ Mangelhafte Effizienz

- Bei den Varianten der Manchesterkodierung

## Lösungsmöglichkeit: Blockcodes

- Moderne Netzwerktechnologien kodieren die Nutzdaten zuerst mit einer Kodierung, die...
  - **effizient ist**,
  - **Taktrückgewinnung garantiert** und
  - die **Durchschnittsverschiebungen vermeidet**
- Diese Kodierungen **verbessern** die Nutzdaten dahingehend, dass eine weitere Kodierung mit den Leitungscodes NRZ, NRZI oder MLT-3 ohne Probleme möglich ist
  - Danach ist eine Kodierung mit NRZ oder NRZI problemlos
- Beispiele für Kodierungen, die die Nutzdaten zuerst aufbereiten, sind 4B5B, 5B6B und 8B10B
  - Diese Kodierungen gehören zur Gruppe der **Blockcodes**, weil diese Eingabeblocks fester Größe in Ausgabeblocks fester Größe kodieren
- Das Ziel ist also, die positiven Eigenschaften der Manchesterkodierung und eine möglichst hohe Effizienz zu erreichen



## 4B5B-Kodierung

- 4 Nutzdatenbits werden auf 5 Codebits abgebildet
    - Mit 5 Bits sind 32 Kodierungen möglich
      - Nur 16 Kodierungen werden für Daten verwendet (0–9 und A–F)
      - Die Übrigen 16 Kodierungen werden teilweise für Steuerzwecke verwendet
    - Wegen des zusätzlichen Bits zur Kodierung wird die kodierte Bitrate um den Faktor  $5/4$  gegenüber der Nutzdatenbitrate gesteigert
      - Effizienz der 4B5B-Kodierung: 80%
    - Jede 5-Bit-Kodierung hat maximal eine führende Null und im Ausgabedatenstrom gibt es maximal drei Nullen in Folge
      - Die **Taktrückgewinnung** für den Empfänger ist somit **möglich**
  - Nach der Kodierung mit 4B5B erfolgt eine **weitere Kodierung** z.B. mittels NRZI oder MLT-3
    - Bei Kombination von 4B5B und NRZI (für 2 Signalpegel) oder MLT-3 (für 3 Signalpegel) kann **keine Durchschnittsverschiebung** auftreten
- Ethernet 100BASE-TX: Nach 4B5B erfolgt eine weitere Kodierung mit MLT-3
  - FDDI und Ethernet 100BASE-FX: Nach 4B5B erfolgt eine weitere Kodierung mit NRZI

# 4B5B-Kodierung (Tabelle)

Bezeichnung	4B	5B	Funktion
0	0000	11110	0 Hexadezimal (Nutzdaten)
1	0001	01001	1 Hexadezimal (Nutzdaten)
2	0010	10100	2 Hexadezimal (Nutzdaten)
3	0011	10101	3 Hexadezimal (Nutzdaten)
4	0100	01010	4 Hexadezimal (Nutzdaten)
5	0101	01011	5 Hexadezimal (Nutzdaten)
6	0110	01110	6 Hexadezimal (Nutzdaten)
7	0111	01111	7 Hexadezimal (Nutzdaten)
8	1000	10010	8 Hexadezimal (Nutzdaten)
9	1001	10011	9 Hexadezimal (Nutzdaten)
A	1010	10110	A Hexadezimal (Nutzdaten)
B	1011	10111	B Hexadezimal (Nutzdaten)
C	1100	11010	C Hexadezimal (Nutzdaten)
D	1101	11011	D Hexadezimal (Nutzdaten)
E	1110	11100	E Hexadezimal (Nutzdaten)
F	1111	11101	F Hexadezimal (Nutzdaten)
Q	—	00000	Quiet (Leitung ist tot) $\Rightarrow$ Signalverlust
I	—	11111	Idle (Leitung ist untätig) $\Rightarrow$ Pause
J	—	11000	Start (Teil 1)
K	—	10001	Start (Teil 2)
T	—	01101	Ende (Teil 1)
R	—	00111	Ende (Teil 2) $\Rightarrow$ Reset
S	—	11001	Set
H	—	00100	Halt (Übertragungsfehler)

- Die in der Tabelle fehlenden 5-Bit-Kombinationen sind ungültig, da sie mehr als 1 führende oder 2 aufeinanderfolgende Nullen besitzen

Bei Fast Ethernet 100BASE-TX beginnen Rahmen mit einem JK und enden mit einem TR

## 5B6B-Kodierung (1/2)

- 5 Nutzdatenbits werden auf 6 Codebits abgebildet
  - Von den 32 möglichen 5-Bit-Wörtern werden 20 auf 6-Bit-Wörter mit einer identischen Anzahl Einsen und Nullen abgebildet  
⇒ **neutrale Ungleichheit** (*balanced*)
  - Für die verbleibenden zwölf 5-Bit-Wörter existiert je eine Variante mit 2 Einsen und 4 Nullen und eine mit 4 Einsen und 2 Nullen  
⇒ **positive oder negative Ungleichheit** (*unbalanced*)
- Sobald das erste 5-Bit-Wort ohne Abbildung mit neutraler Ungleichheit verarbeitet werden soll, wird auf die Variante mit der positiven Ungleichheit zurückgegriffen
  - Beim nächsten 5-Bit-Wort ohne neutrale Ungleichheit wird die Variante mit negativer Ungleichheit verwendet
    - Die Varianten mit positiver oder negativer Ungleichheit wechseln sich ab

## 5B6B-Kodierung (2/2)

- Nach der Kodierung mit 5B6B erfolgt eine weitere Kodierung mit NRZ
  - Das ist möglich, da bei 5B6B die **Taktrückgewinnung** für den Empfänger **möglich** ist und **keine Durchschnittsverschiebungen** auftreten können
- Vorteil gegenüber Manchesterkodierung: Höhere Baudrate
  - Effizienz:  $5/6 = 83, \bar{3}\%$

5B6B wird bei Fast-Ethernet 100Base-VG verwendet

# 5B6B-Kodierung (Tabelle)

5B	6B neutral	6B positiv	6B negativ	5B	6B neutral	6B positiv	6B negativ
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-Kodierung

- Bei 8B10B werden 8 Nutzdatenbits auf 10 Codebits kodiert
  - Die Effizienz ist somit 80%
- Jede 8B10B-Kodierung ist so aufgebaut, dass in den 10 Bits entweder...
  - 5x Null und 5x Eins vorkommt  $\Rightarrow$  neutrale Ungleichheit
  - 6x Null und 4x Eins vorkommt  $\Rightarrow$  positive Ungleichheit
  - 4x Null und 6x Eins vorkommt  $\Rightarrow$  negative Ungleichheit
- Nach der Kodierung mit 8B10B erfolgt eine weitere Kodierung via NRZ
  - **Durchschnittsverschiebungen sind nicht möglich**, weil einige der  $2^8 = 256$  möglichen 8-Bit-Wörter auf 2 verschiedene Arten kodiert werden können
    - So werden Ungleichheiten ausgeglichen
- Jede 10-Bit-Kodierung enthält mindestens 3 Pegelsprünge und nach spätestens 5 Takten wechselt der Signalpegel
  - Das **ermöglicht** dem Empfänger **die Taktrückgewinnung**

Wird u.a. bei Gigabit-Ethernet 1000Base-CX, -SX, -LX, FibreChannel, InfiniBand, DisplayPort, FireWire 800 (IEEE 1394b) und USB 3.0 verwendet

## 8B6T-Kodierung

- 8B6T = 8 Binary 6 Ternary
  - Hilfreich für Netzwerktechnologien, die **> 2 Signalpegel** verwenden
- Kodiert 8-Bit-Blöcke in je 6 Symbole, von denen jedes den Zustand -, 0 oder + repräsentieren kann
  - Die Symbole der Zustände sind stellvertretend für elektrische Signalpegel
- Die Kodierung wird anhand einer Tabelle durchgeführt, die alle  $2^8 = 256$  möglichen 8-Bit-Kombinationen enthält
  - Wie in der Tabelle zu sehen ist, macht die Ausgabe von 8B6T **Durchschnittsverschiebungen unmöglich** und die häufigen Wechsel des Signalpegels **ermöglichen die Taktrückgewinnung** für den Empfänger
- Im Gegensatz zu den Blockcodes 4B5B, 5B6B und 8B10B, die die Nutzdaten nur *verbessern* und anschließend eine Kodierung mit NRZ(I) oder MLT-3 erfordern, **kann eine 8B6T-Kodierung unmittelbar für die Übertragung genutzt werden**

Fast-Ethernet 100BASE-T4 verwendet diesen Leitungscode

# 8B6T-Kodierung (Tabelle)

8-Bitfolge	8B6T-Code	8-Bitfolge	8B6T-Code	8-Bitfolge	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++

USW.



# Zusammenfassung

Leitungscode	Signal- pegel	Durchschnitts- verschiebungen	Pegel- wechsel	Selbstsynchron- nisierend <sup>1</sup>	Effizienz <sup>2</sup>	Direkt übertragbar	Weitere Kodierung
NRZ	2	ja	bei Wechseln	nein	100%	nein	—
NRZI	2	ja	bei 1	nein	75%	nein	—
MLT-3	3	ja	bei 1	nein	100%	nein	—
RZ	3	ja	immer	ja	50%	nein	—
Unip. RZ	2	ja	bei 1	nein	75%	nein	—
AMI	3	nein	bei 1	nein	100%	nein	Scrambler
B8ZS	3	nein	bei 1	ja	100%	ja	—
Manchester	2	nein	immer	ja	50%	ja	—
Manchester II	2	nein	immer	ja	50%	ja	—
Diff. Manch.	2	ja	immer	ja	50%	ja	—
4B5B	2	ja	—	ja	80%	nein	NRZI oder MLT-3
5B6B	2	nein	—	ja	83,3%	nein	NRZ
8B10B	2	nein	—	ja	80%	nein	NRZ
8B6T	3	nein	—	ja	100%	ja	—

<sup>1</sup> Gibt an, ob die Taktrückgewinnung mit diesem Leitungscode möglich ist.

<sup>2</sup> Verhältnis von Bitrate (Nutzdaten in Bits pro Zeit) und Baudrate (Signaländerungen pro Sekunde).