

Quellen- und Leitungskodierung



0

1





- Quellencodierung:
 - ➔ Dient der Optimierung des Durchsatzes
 - ➔ Dazu gehört auch Kompression
- Kanalcodierung:
 - ➔ Dient der Fehlersicherung
 - ➔ Paritäts-Bits, Prüfsummen, Redundanz...
- Leitungscodierung
 - ➔ Einem oder mehreren Bits wird ein Symbol zugeordnet, das auf der Leitung übertragen wird.

Forderungen an einen Code



- Daten werden kodiert oder müssen kodiert werden, um
 - eine geeignete Darstellung für die technische Verarbeitung zu haben,
 - eine ökonomische Darstellung und Übertragung zu gewährleisten,
 - Informationen vor Verfälschung oder unberechtigtem Zugriff zu schützen.

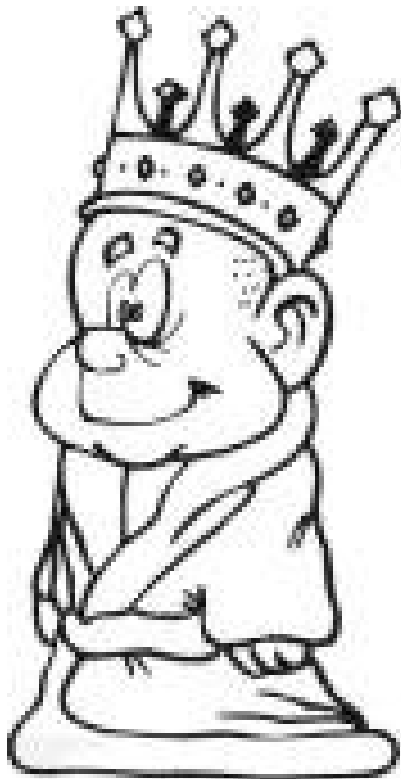
- Aus diesen Bestrebungen resultieren u.a. folgende Anforderungen an Eigenschaften eines Codes:
 - umkehrbar eindeutig
 - geringe Wortlänge
 - Codierung leicht realisierbar
 - etc.

Kleiner Exkurs: Verschlüsselung



- Ein absolut sicheres Verfahren: Der Schlüssel ist ein Unikat und so lang wie die Nachricht (One Time Pad)
- Ein weniger sicheres Verfahren: Der Schlüssel soll ein Unikat sein und ist wesentlich kürzer als die Nachricht
 - ➔ ENIGMA und andere mechanische Verschlüsselungs-Geräte
 - ➔ Nutzerfehler erlaubten das „Knacken“
 - ➔ Bedeutungslos seit der datentechnischen Verschlüsselung
- Ganz unsichere Verfahren sind schon knackbar, wenn man das Verfahren selbst kennt (z.B. IBM -> HAL)
- Grundregel: Nur die Kenntnis des richtigen Schlüssels, aber nicht allein die des Verfahrens ist ausreichend, um eine verschlüsselte Nachricht zu knacken (Kerckhoff).

Der verfressene König*



- Schweinebraten
- Schokoladenpudding
- Essiggurken
- Erdbeertorte

*Nach Walter R. Fuchs, 1968

Des Königs (Quellen-) Kodierung



- Der König beschloss, seine Befehle zu kodieren.
 - rechte Hand heben: Schweinebraten
 - linke Hand heben: Schokoladenpudding
 - erst rechte, dann linke Hand heben: Essiggurken
 - zweimal rechte Hand heben: Erdbeertorte
- Aber es gab Probleme:
- Hob der König dreimal die rechte Hand (RRR) bekam er an einem Tag dreimal Schweinebraten, an einem anderen Tag zuerst Erdbeertorte, dann Schweinebraten und an einem dritten erst Schweinebraten, dann Erdbeertorte.

Des Mathematikus Kodierung



- Benötigt wird eine eindeutige, binäre Kodierung.
- Um vier Worte binär zu kodieren, wir eine Codewortlänge von mindestens 2 Zeichen benötigt.
 - Schweinebraten RR
 - Schokoladenpudding RL
 - Essiggurken LR
 - Erdbeertorte LL
- Nun war die Kodierung unverwechselbar.
- Der König bestellte: LRRLLLRRRRLL und erhielt
Gurken, Pudding, Torte, 2 x Braten und Torte

Des Königs Vorlieben



- Die Kodierung war nun eindeutig und der König erhielt immer, was er auch wollte.
- Er wurde immer dicker und dicker und das Heben der Hände strengte ihn an.
- Ist die Kodierung optimal oder könnte sie noch kürzer sein?
- Der Mathematiker ließ seinen Assistenten eine Statistik über die königlichen Essenwünsche führen.
 - Jeden Tag ißt der König durchschnittlich 18 Gerichte.
 - Die Verteilung der Häufigkeiten liegt bei 9 x Schweinebraten, 6 x Schokopudding, 1 x Essiggurken und 2 x Erdbeertorte.

Kodierung nach Vorlieben I



- Um die Kodierung optimal zu wählen, beschloss der Mathematiker dem Braten einen möglichst kurzen Code zu geben, während der Code für die Gurken ruhig länger sein konnte.
- Der Code soll optimal, eindeutig und zweckmäßig sein.
- Bei der Generierung des Codes muss die Fano-Bedingung berücksichtigt werden:
 - Kein Code (kompletter Code!) darf der Beginn eines anderen verwendeten Codes sein.

Kodierung nach Vorlieben II



- Für den Schweinebraten wird festgelegt:
 - Braten --> R
- Damit ist das R verbraucht und kein Codewort darf mehr mit R beginnen.
- Für den Schokoladenpudding wird festgelegt:
 - Pudding --> LR
- Damit bleibt nur noch ein zweistelliges Wort (LL) übrig, es sind aber noch zwei Worte zu kodieren. Daher werden dreistellige Codes gewählt:
 - Gurken --> LLR
 - Torte --> LLL

Kodierung nach Vorlieben III



- Bei der ersten Kodierung benötigt der König für die Bestellung der 18 Speisen 36 Bit.
- Ausgehend von der vorgegebenen Häufigkeitsverteilung benötigt der König mit dem neuen Code...
 - 9 x Schweinebraten 9 Bit
 - 6 x Schokopudding 12 Bit
 - 1 x Essiggurken 3 Bit
 - 2 x Erdbeertorte 6 Bit
- Es ergeben sich in Summe für diese 18 Speisen 30 Bit.
- Was bestellt der König bei LRRRLLR?

Häufigkeit und Wahrscheinlichkeit I



- Bei der Ermittlung von Häufigkeiten handelt es sich um Erfahrungswerte.
- Führt man ein Ereignis n -mal durch und es führt in m Fällen zu dem zufälligen Ergebnis A , dann liegt die Häufigkeit $h(A) = m/n$ beliebig nahe an der Wahrscheinlichkeit $P(A)$.
- Für unser Beispiel heißt das:
 - A1: Der König bestellt Braten
 - A2: Der König bestellt Pudding
 - A3: Der König bestellt Torte
 - A4: Der König bestellt Gurken

Häufigkeit und Wahrscheinlichkeit II



- Nehmen wir an, dass die Werte auf genügend vielen Beobachtungen beruhen, können wir die Wahrscheinlichkeiten durch Häufigkeiten annähern.
 - $P(A1) = 9/18 = 1/2$
 - $P(A2) = 6/18 = 1/3$
 - $P(A3) = 2/18 = 1/9$
 - $P(A4) = 1/18$
- Die Summe aller Wahrscheinlichkeiten ergibt immer den Wert 1.
- Je kleiner die Wahrscheinlichkeit des Auftretens der Nachricht ist, desto höher ist der *Informationsgehalt*, d.h. die Länge des Codes darf grösser sein.

Häufigkeit und Wahrscheinlichkeit III



- Als Formel bedeutet das:
 - $I(A) = \text{Id} (1/P(A))$ [Bit*]
- Wir wenden nun diese Formel auf die Nachrichten A1 bis A4 an:
 - $I(A1) = \text{Id} (1/(1/2)) = \text{Id}(2) = 1,000$ bit
 - $I(A2) = \text{Id} (1/(1/3)) = \text{Id}(3) = 1,585$ bit (2**1,585=3)
 - $I(A3) = \text{Id} (1/(1/9)) = \text{Id}(9) = 3,170$ bit
 - $I(A4) = \text{Id} (1/(1/18)) = \text{Id}(18) = 4,170$ bit

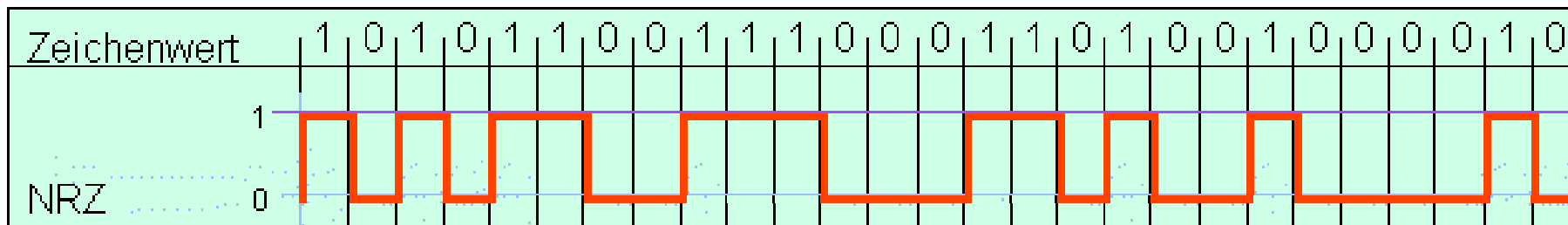
*Die Einheit der Information wird Bit genannt. Sie ist aber nicht identisch mit der Stelle einer Binärzahl.

Leitungskodierungen



- Auch Bitwerte und -kombinationen können auf verschiedene Weise kodiert werden.
 - Non-Return-to-Zero
 - Return-to-Zero
 - Non-Return-to-Zero-Mark
 - Bi-Phase, Bi-Phase-Mark (Manchester)
 - Differential Bi-Phase
 - Delay-Mark
 - Bi-Puls (Dipuls)
- Die verschiedenen Verfahren werden unterschieden in Verfahren ohne Taktrückgewinnung, mit Taktrückgewinnung und Mehrfachkodierungsverfahren.

Non-Return-to-Zero (NRZ) I

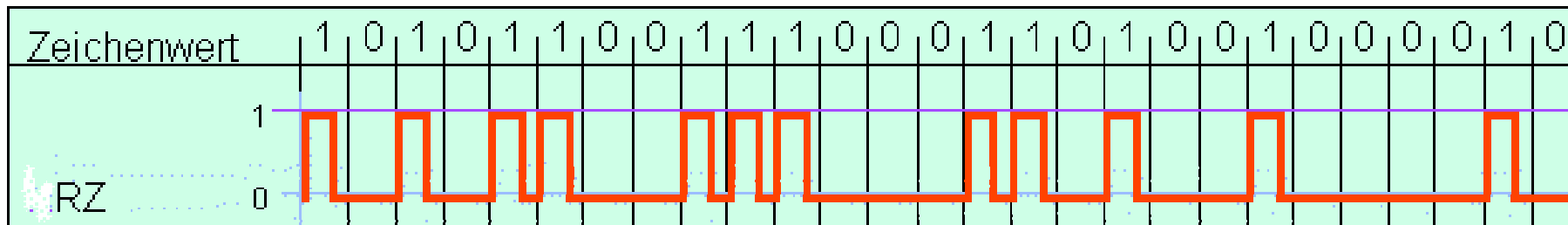


Non-Return-to-Zero (NRZ-L) II



- Der Non-Return-to-Zero Code ist die einfachste Art einer Kodierung (Standard in digitalen Geräten).
- Jedem Bit wird ein Signal zugeordnet (z.B. einem Zeichenwert '0' eine Null-Spannung, '1' eine '+'-Spannung).
- Folgen gleicher Bits werden durch Folgen gleicher Signalwerte übertragen.
- Der Code eignet sich nicht zur Taktrückgewinnung.
- Im Fall von Fehlern sind nur die fehlerhaften Bits falsch.

Return-to-Zero (RZ) I

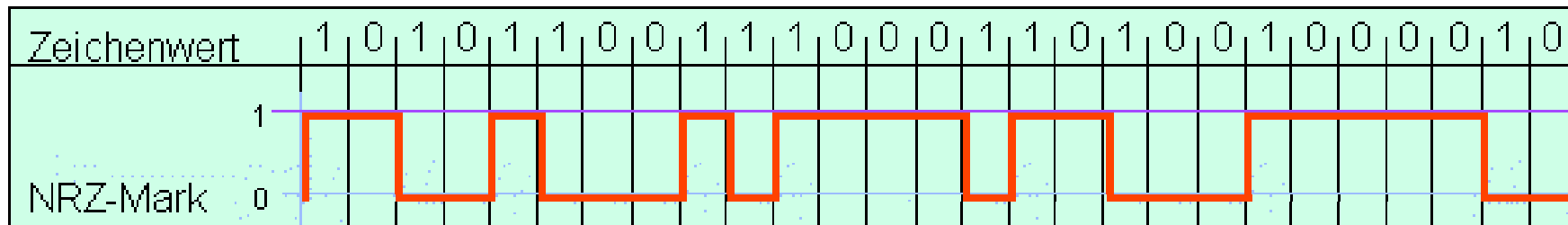


Return-to-Zero (RZ) II



- Bei diesem Code kehrt das Signal immer wieder zur Nulllinie zurück.
- Eine Eins wird durch eine '+'-Spannung von halber Bit-Dauer kodiert, eine Null bleibt auf der Nulllinie für die gesamte Bit-Dauer.
- Dadurch dass es bei längeren Null-Folgen keine Signaländerung gibt, ist auch beim Return-to-Zero-Code keine Taktrückgewinnung möglich.

Non-Return-to-Zero-Mark (NRZ-Mark) I

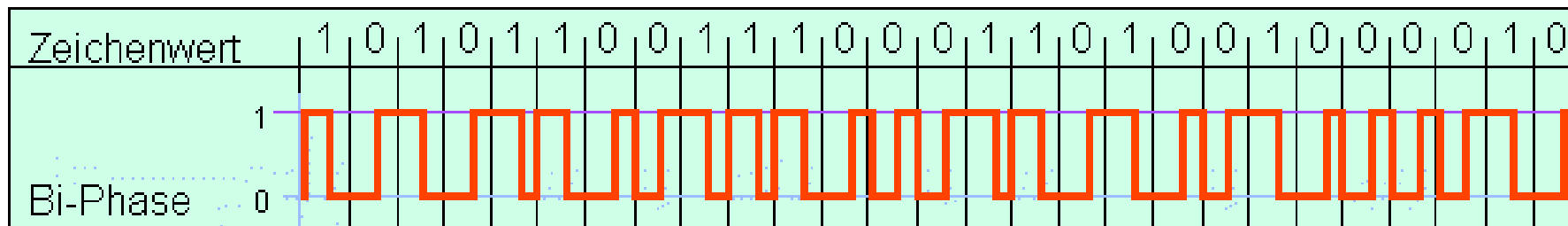


Non-Return-to-Zero-Mark (NRZ-Mark) I



- Zu Beginn jeder 1-Bit-Periode wird ein Flankenwechsel durchgeführt (von + zu Null und zurück zu +).
- Bei einem Null-Bit-Wert ändert sich der Signalzustand nicht.
- Auch bei diesem Verfahren ist eine Taktrückgewinnung nicht möglich.
- Bei den letzten drei Verfahren kann durch eine beliebig lange Null-Folge eine Taktrückgewinnung nicht durchgeführt werden. Wenn der Sender aber nach einer bestimmten Anzahl von Nullen immer eine Eins einfügt (Bitstuffing), ist eine Taktrückgewinnung doch möglich.

Bi-Phase I

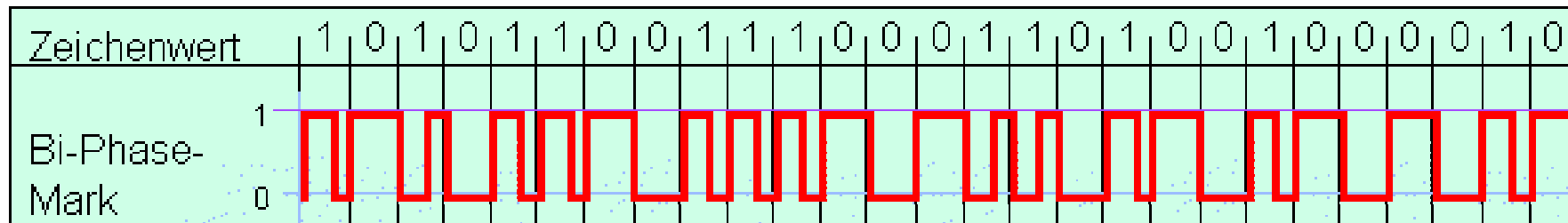


Bi-Phase II



- In der Mitte einer Taktperiode erfolgt ein Übergang von + nach Null bei Übertragung einer Eins bzw. von Null nach + bei Übertragung einer Null.
- Entsprechend ist der Signalwert am Anfang einer Bitperiode so einzustellen, dass der Signalübergang sichergestellt werden kann.
- Die Taktrückgewinnung ist bei diesem Verfahren möglich.
- Die maximale Frequenz ist bei diesem Verfahren sehr viel höher als bei den letztgenannten.

Bi-Phase-Mark (Manchester) I

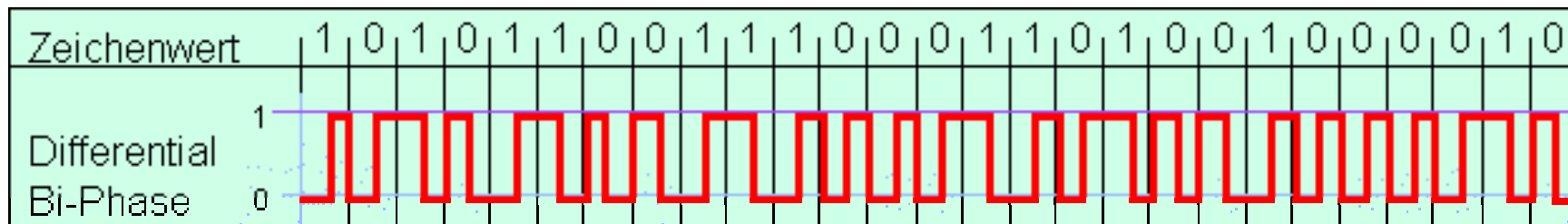


Bi-Phase-Mark (Manchester) II



- Am Anfang eines Bits wird immer ein Flankenwechsel durchgeführt, durch den der Takt zurückgewonnen werden kann.
- In der Mitte der Bitperiode wird immer dann ein Flankenwechsel vorgenommen, wenn eine Eins übertragen wird, sonst nicht.
- Die Eigenschaften gleichen dem Bi-Phase-Code.
- Dieser Code wird im Ethernet-LAN benutzt.

Differential Bi-Phase I

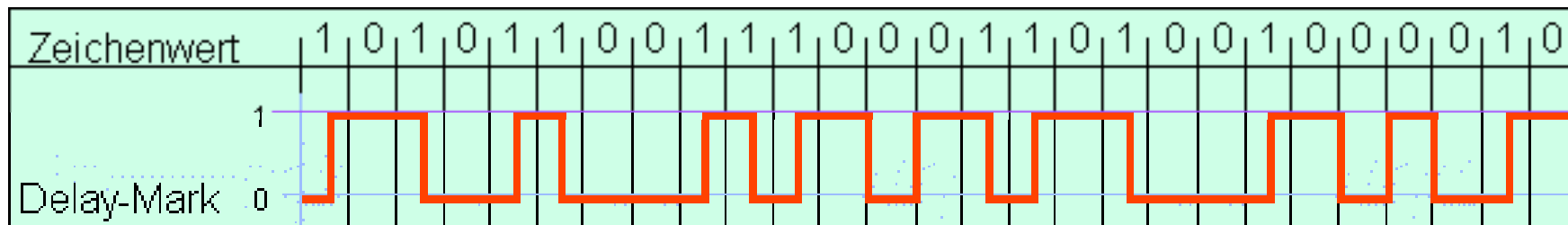


Differential Bi-Phase II



- In der Mitte einer Bitphase wird immer ein Flankenwechsel durchgeführt, durch den der Takt zurückgewonnen werden kann.
- Am Anfang der Bitperiode wird nur dann ein Flankenwechsel vorgenommen, wenn eine Null übertragen wird, sonst nicht.
- Die Eigenschaften gleichen dem Bi-Phase-Mark-Code.

Delay-Mark I

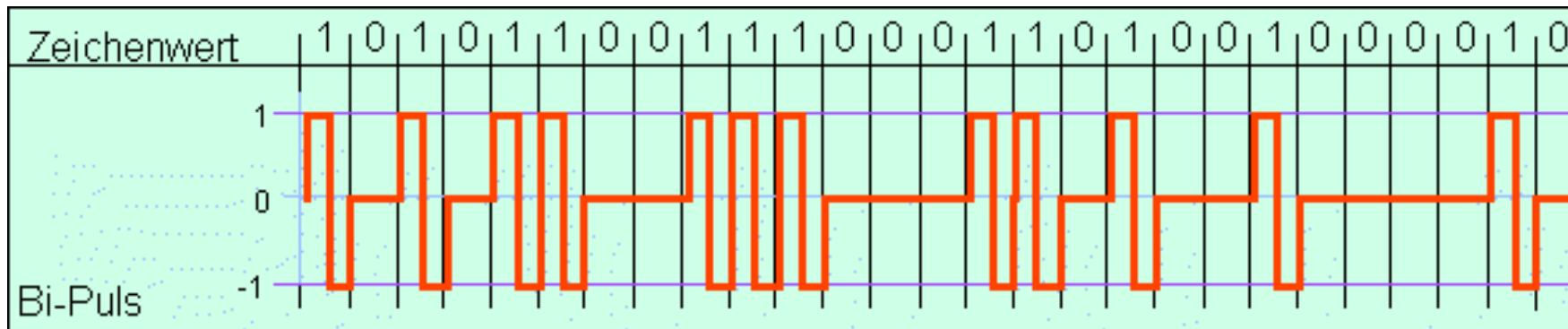


Delay-Mark II



- Wie beim Bi-Phase-Mark-Code wird in der Mitte einer Bit-Phase bei einer Eins ein Zustandswechsel durchgeführt, es gibt aber keinen am Anfang eines Bits.
- Um bei längeren Bitfolgen von Null-Bits sicher den Takt zurückgewinnen zu können, wird z.B. ab drei hintereinander folgenden Null-Bits am Anfang jedes Bits ein Flankenwechsel durchgeführt.
- Bei diesem Verfahren ist die Resynchronisation des Empfängers sehr problematisch.

Bi-Puls (Dipuls) I

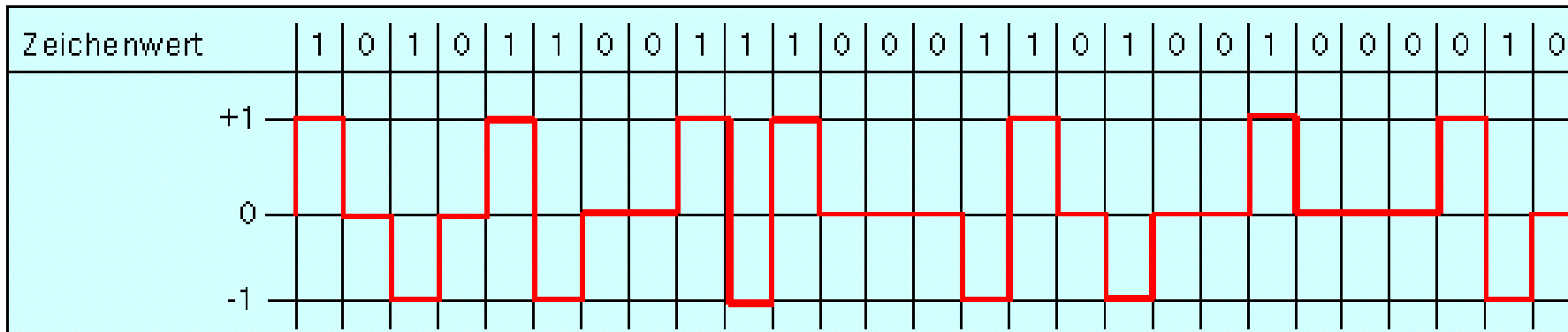


Bi-Puls (Dipuls) II



- Bei der Bi-Puls-Kodierung wird ein ternäres Signal verwendet.
- Wird eine Eins übertragen, so wird in einer halben Bitperiode Plus und in der anderen halben Minus übertragen.
- Für eine Null wird auch immer eine Null übertragen.
- Eine sichere Taktrückgewinnung ist nicht möglich.
- Die Resynchronisation des Empfängers ist aber sehr einfach.

Alternating-Mark-Inversion (AMI) I



Alternating-Mark-Inversion (AMI) II



- Auch bei der AMI-Kodierung wird ein ternäres Signal verwendet.
- Eine Eins wird durch alternierende Plus- und Minus-Signale dargestellt, eine Null durch ein Null-Signal.
- Bei AMI-Kodierung kann der Takt sicher zurückgewonnen werden. Dazu gibt es eine besondere Wechselregel bei langen Nullfolgen.
- Nach einer Anzahl von n Nullbits wird ein zusätzliches Plus- oder Minussignal entgegen der Richtung der Wechselregel gesendet.

Alternating-Mark-Inversion (AMI) III



- Danach erfolgt ein Wechsel in die jeweils andere Richtung und wiederum nach drei Bit ein Signal entgegen der Richtung der Wechselregel.
- Nach einem Plus-Minus-Wechsel (oder umgekehrt) müssen erst drei weitere Signale abgewartet werden, bis erkannt werden kann, ob es sich um die Folge „1XXX“ oder „0000“ handelt.
- Die AMI-Kodierung wird für PCM und ...
- speziell in der CCITT-Empfehlung G.703 in den 2 Mbit/sek Vermittlungssystemen der europäischen Post- und Telekommunikationsverwaltungen eingesetzt und ist daher von besonderer Bedeutung.

Alternating-Mark-Inversion (AMI) IV

