

# Leitungscode für die Digitalsignalübertragung

**Codierung** = Umsetzung von (Binären) Quellsymbolen (Bitfolge) in (mehrstufige) Symbole

**Ziel:** Anpassung an den Übertragungskanal  
Formung des Signalspektrums (insbesondere: Gleichanteil reduzieren)

**Problem:** Code und Bitrate bestimmen das Spektrum des Leitungssignals  
Übertragungskanal bestimmt den optimalen Leitungscode  
Leitungscode hat Einfluss auf die Komplexität des Systems (Preis!)  
Anforderungen widersprechen sich teilweise

**Lösung:** verschiedene Leitungscode **simulieren** und Ergebnisse analysieren

**Beispiele:** ISDN-Ba                      **MMS43**-Code (Deutschland)  
ISDN-Ba                      **2B1Q**-Code (USA)  
ISDN-PMxA                  **HDB3**-Code

## Beschreibung der Leitungssignale

Binäre Quellensymbolfolge mit Bitrate  $v_b = \frac{1}{T_b}$

Codierte Bitfolge (Leitungscode) mit Symbolrate  $v_c = \frac{1}{T_c}$

Symbol wird Signalelement  $a_i$   $g(t)$  zugeordnet ( $a_i$  = Amplitudenkoeffizient)  
( $g(t)$  = Zeitfunktion einer Spannung)

Bei N-stufigem Code:  $i = 1 \dots N$

$$a_i = \frac{2i - N - 1}{N - 1}$$

Beispiele:

ternär  $\{a_i\} = \{-1; 0, +1\}$

quaternär  $\{a_i\} = \{-1; -\frac{1}{3}; +\frac{1}{3}; +1\}$

quinär  $\{a_i\} = \{-1; -\frac{1}{2}; 0; +\frac{1}{2}; +1\}$

Bild-Quelle:  
Morgenstern  
Wellhausen  
Leitungs-  
codes

# Blockcodes

**Prinzip:** Block von **m** Bits wird umgewandelt in **n** mehrstufige Symbole

**Bezeichnung:** **m B n N** (B für Binär, N für N-stufig)

z.B.: **2 B 1 Q** (2 Bits in 1 Quaternärsymbol, 4-stufig)  
**4 B 3 T** (4 Bits in 3 Ternärsymbole, 3-stufig)

**Begriffe:** **Alphabet** = geordneter Vorrat an Codeworten

**Blocklänge** = Anzahl der Bits bzw. Symbole pro Block

**Digitale Summe** = Summe der Amplitudenkoeffizienten  $a_i$

**Digitale Wortsumme** = Summe der  $a_i$  eines Blocks

**Laufende digitale Summe** = Momentanwert der digitalen Summe

**Maximale Schwankung der digitalen Summe** = Differenz zwischen

Maximalem und minimalem Wert der laufenden digitalen Summe

**Schlusszustand** = Wert der laufenden digitalen Summe am Ende des Blocks

**Wichtig:** Für Blockcodes wird eine Blocksynchronisation benötigt.

## Redundanzfreie Codes

Redundanz: Alphabet hat mehr Symbole als binäre Zustände des codierten Blocks

Bedingung für Redundanzfreiheit:  $N = 2^m$ ,  $m = 2, 3, 4, \dots$

Beispiele: **2B1Q**  $m = 2$ ,  $N = 2^2 = 4$  (quaternär)  $\Rightarrow$  redundanzfrei

**4B3T**  $m = 4$ ,  $2^4 = 16$  binäre Zustände, aber  $3^3 = 27$  Symbole  
 $\Rightarrow$  nicht redundanzfrei

**Eigenschaften** redundanzfreier Codes:

- Leistungsdichtespektrum aller Codes hat die gleiche Form
- Leistungsdichtespektrum ist um Faktor **m** komprimiert
- Leitungscode hat nur **ein** Alphabet
- Codetabelle ist zweckmäßigerweise ein **Gray-Code**

d.h.: benachbarte Codewerte unterscheiden sich nur durch ein Bit  
fehlerhaftes Symbol erzeugt meist nur einen Bitfehler

## Quaternärcode 2B1Q

**Eigenschaften:** Symbolrate  $v_c = 0,5 v_b$  Bitrate der Quellensymbolfolge  
 Symboldauer  $T_c = 2 T_b$  Symboldauer Binärsymbol  
 Anzahl Amplitudenstufen = 4

Codetabelle eines quaternären Codes.

Binärwort	Quaternäre Amplitudenstufe
0 0	- 1
0 1	- 1/3
1 1	1/3
1 0	1

Bild-Quelle:  
Morgenstern  
Wellhausen  
Leitungs-  
codes

**Hinweis:** Gray-Codierung  $\Rightarrow$  Binärwort der 3. Amplitudenstufe ( 1 1 ) ist nicht die Binäre Zahl 2 im System der Dualzahlen ( 1 0 )

# Oktonärcode 3B1O

## Eigenschaften:

Symbolrate  $v_c = 0,33 v_b$

Symboldauer  $T_c = 3 T_b$

Anzahl Amplitudenstufen = 8

## Beispiel: Wirkung der Gray-Codierung

Sender sendet Pegel  $-1/7 = 010$

Empfänger empfängt  $1/7 = 110$

⇒ 1 Bit ist fehlerhaft

bei Dualzahlen: Stufe 4 = 011, Stufe 5 = 100

⇒ bei Amplitudenfehler: 3 Bit sind fehlerhaft

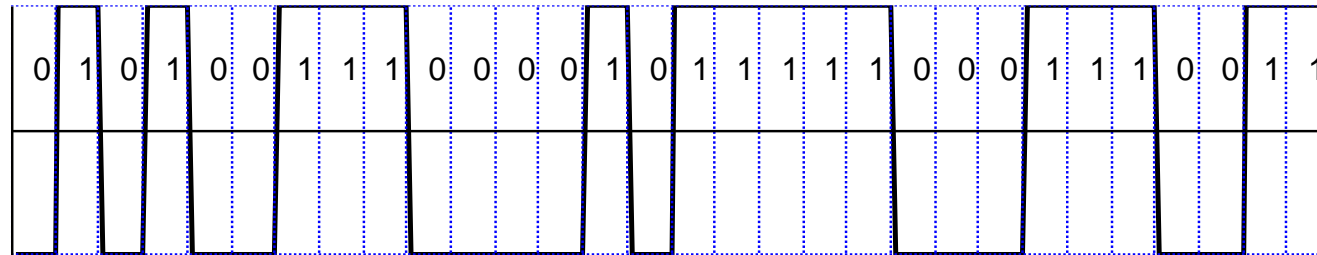
Codetabelle eines oktonären Codes.

Binärwort	Oktonäre Amplitudenstufe
0 0 0	-1
0 0 1	-5/7
0 1 1	-3/7
0 1 0	-1/7
1 1 0	1/7
1 1 1	3/7
1 0 1	5/7
1 0 0	1

## Vergleich: binär – quaternär - oktonär

Amplituden-  
stufen: 2

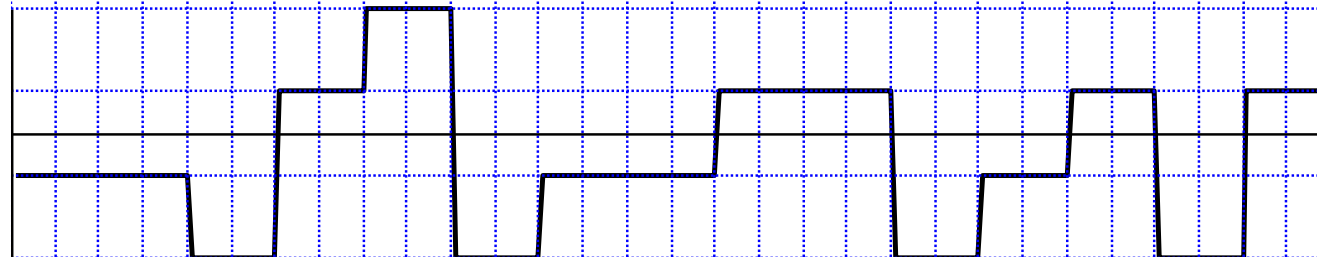
Symboldauer  
 $T_S = 1$



Binär

Amplituden-  
stufen: 4

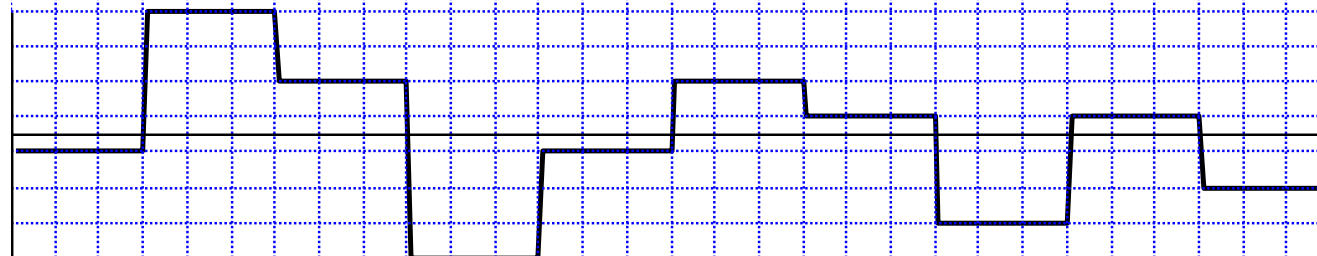
Symboldauer  
 $T_S = 2$



**2B1Q**  
Quaternär

Amplituden-  
stufen: 8

Symboldauer  
 $T_S = 3$



**3B1O**  
Oktonär

## Redundanzbehaftete Codes

Redundanz: Alphabet hat mehr Symbole als binäre Zustände des codierten Blocks

Ziel: Spektrum des Leitungssignals beeinflussen, speziell: Gleichstromfreies Spektrum wegen Übertragern im Übertragungsweg → vereinfacht Signalverarbeitung

Beispiele: **AMI** Alternate **M**ark **I**nversion

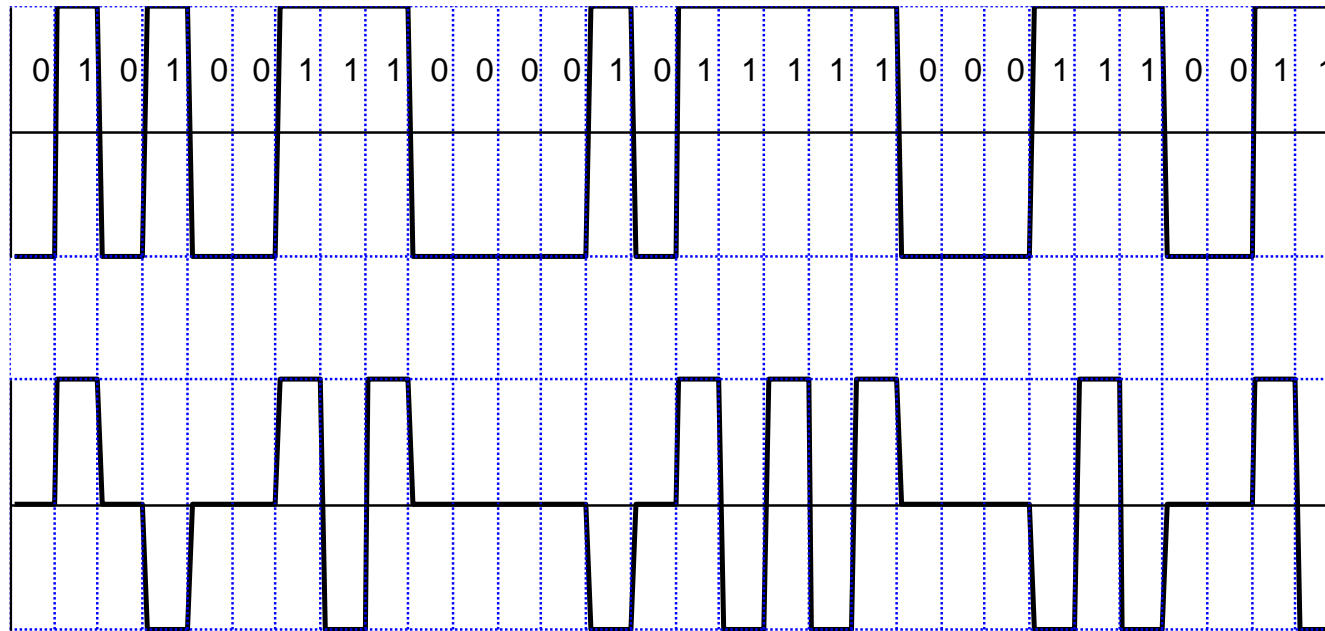
### Eigenschaften:

- pseudoternärer Code (drei Amplitudenstufen)
- Binäre 1 wird abwechselnd durch +1 und -1 dargestellt
- Binäre 0 wird immer durch 0 dargestellt
- Symbolrate verringert sich nicht

### Problem:

- Lange Folgen von 0-Bits verhindern die **Taktrückgewinnung** im Empfänger
- ⇒ Code wird in der Praxis **modifiziert** (z.B.: HDB3)  
oder: ein „**Scrambler**“ beseitigt die langen 0-Folgen

## Beispiel AMI



Binäre Folge	0	1	0	1	0	0	1	1
AMI-codiert	0	+1	0	-1	0	0	+1	-1

## HDB3

HDB3 = High Density Bipolar 3-Code = modifizierter AMI-Code

Ziel: nicht mehr als 3 Symbole „0“ nacheinander im Leitungssignal

Regel: 4 aufeinander Folgende „0“-Symbole ersetzen durch:

**A 0 0 V** falls die Anzahl der vorangehenden binären „1“-Symbole nach der letzten Ersetzung **null oder gerade** ist

**0 0 0 V** falls die Anzahl der vorangehenden binären „1“-Symbole nach der letzten Ersetzung **ungerade** ist

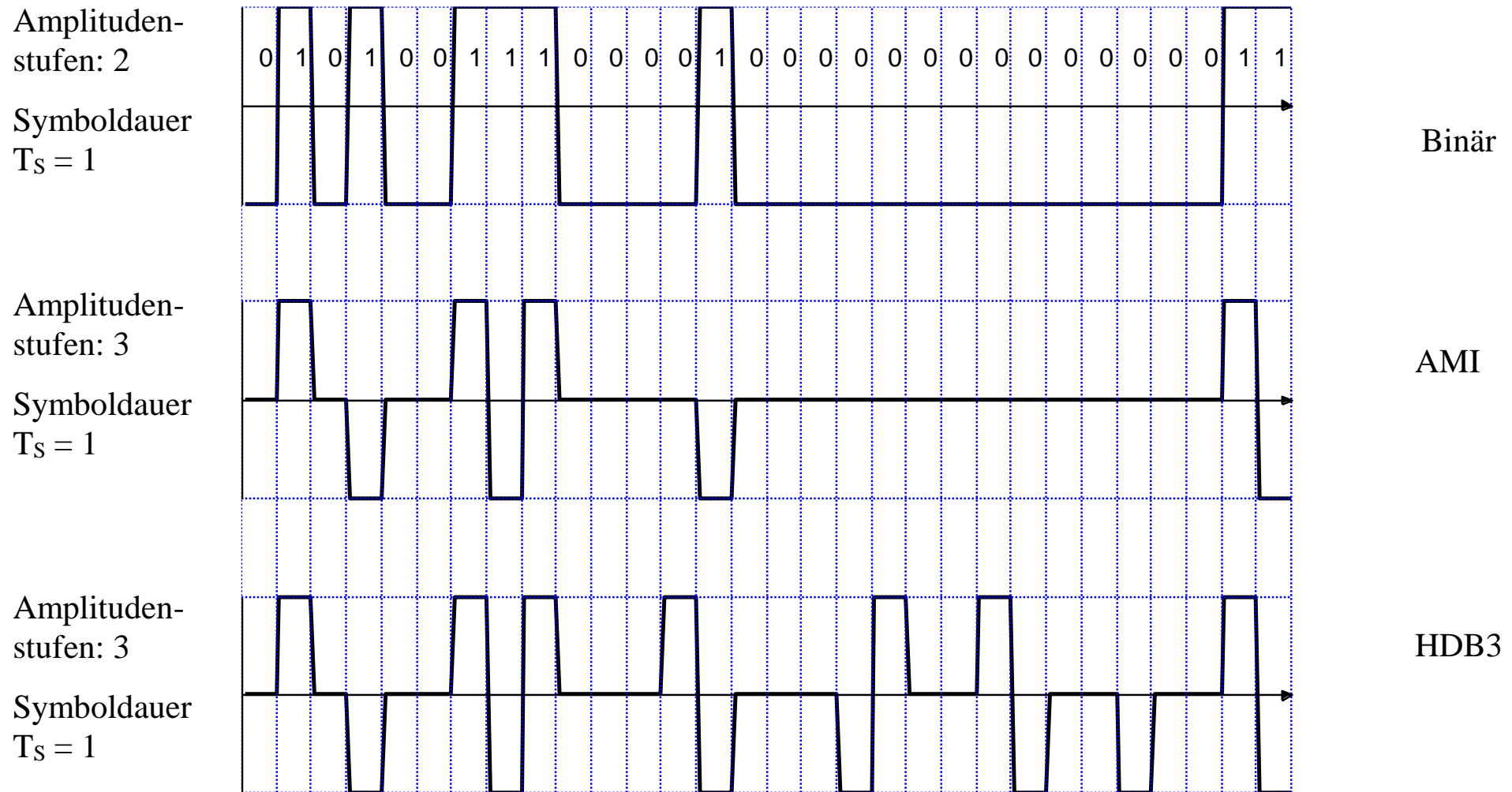
**A** = ein „1“-Symbol mit alternierendem Vorzeichen, d.h. wenn zuletzt +1 gesendet, dann

nun eine -1, und umgekehrt

**V** = eine Verletzung des AMI-Codes, d.h. ein „1“-Symbol mit dem gleichen Vorzeichen wie das zuletzt gesendete „1“-Symbol

→ Nullfolgen werden an den „**Code-Verletzungen**“ erkannt.

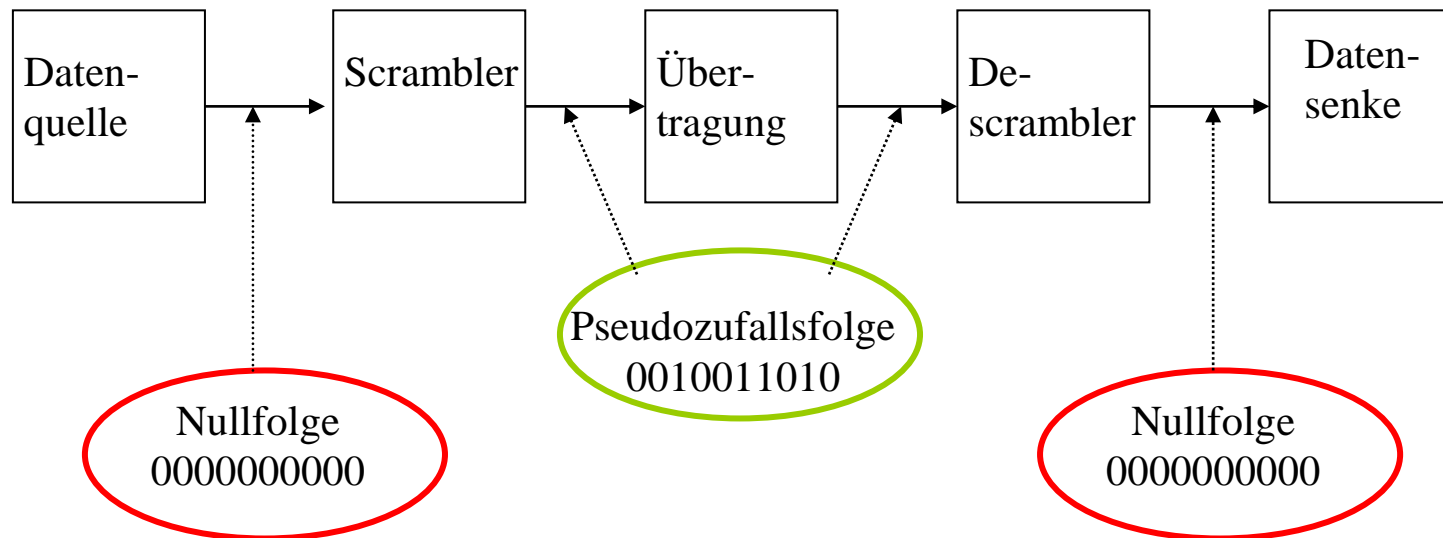
## Vergleich: binär – AMI – HDB3



# Scrambler

**Problem:** Lange Nullfolgen erschweren die Taktrückgewinnung

**Lösung:** Scrambler („Verwürfelung“) erzeugt eine „Pseudozufallsfolge“



# Scrambler-Realisierung

Realisierung: rückgekoppelte Schieberegister mit modulo-2-Addition

Beispiel: 5-stufiger Scrambler

gesendete Bitfolge  $b(n)$ :

$$b(n) = a(n) + b(n-4) + b(n-5) \bmod 2$$

empfangene Bitfolge  $c(n)$ :

$$c(n) = b(n) + b(n-4) + b(n-5) \bmod 2$$

Eingangsbitfolge  $a(n)$ :

$$a(n) = b(n) + b(n-4) + b(n-5) \bmod 2$$

$$\Rightarrow c(n) = a(n)$$

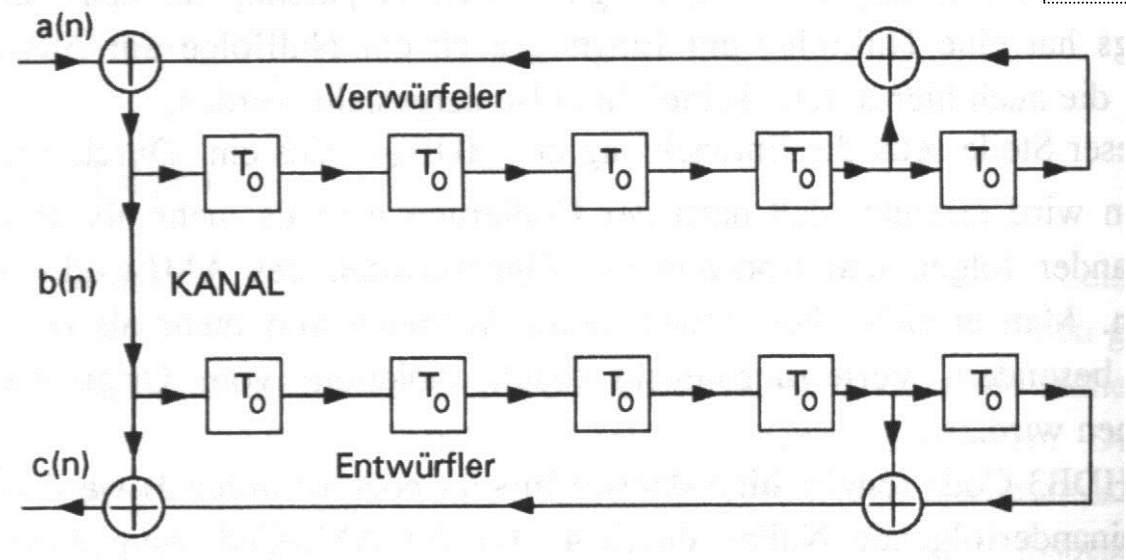


Bild-Quelle:  
Peter Gerdson  
Digitale  
Nachrichten-  
übertragung

## Beispiel Scrambler

Eingangsbitfolge

→ enthält nur 0-Symbole

a(n)      0

interne Zustände des Scramblers

→ Startwert 1 1 1 1 1

b(n-1)	1	0	0	0	0	1	0	0	0	1	1	0	0	1	0	1	1	1	1	1	0	0	0	0	1	0	0	0		
b(n-2)	1	1	0	0	0	0	1	0	0	0	1	1	0	0	1	0	1	0	1	1	1	1	0	0	0	0	1	0	0	
b(n-3)	1	1	1	0	0	0	0	1	0	0	0	1	1	0	0	1	0	1	0	1	1	1	1	0	0	0	0	1	0	
b(n-4)	1	1	1	1	0	0	0	0	1	0	0	0	1	1	0	0	1	0	1	0	1	1	1	1	0	0	0	0	1	
b(n-5)	1	1	1	1	1	0	0	0	0	1	0	0	0	1	1	0	0	1	0	1	0	1	1	1	1	1	0	0	0	0

Sendebitfolge

→ regelmäßige 0-1-Wechsel

b(n)      0 0 0 0 1 0 0 0 1 1 0 0 1 0 1 0 1 1 1 1 1 0 0 0 0 1 0 0 0 1

interne Zustände des Descramblers

→ Startwert 1 1 0 1 1

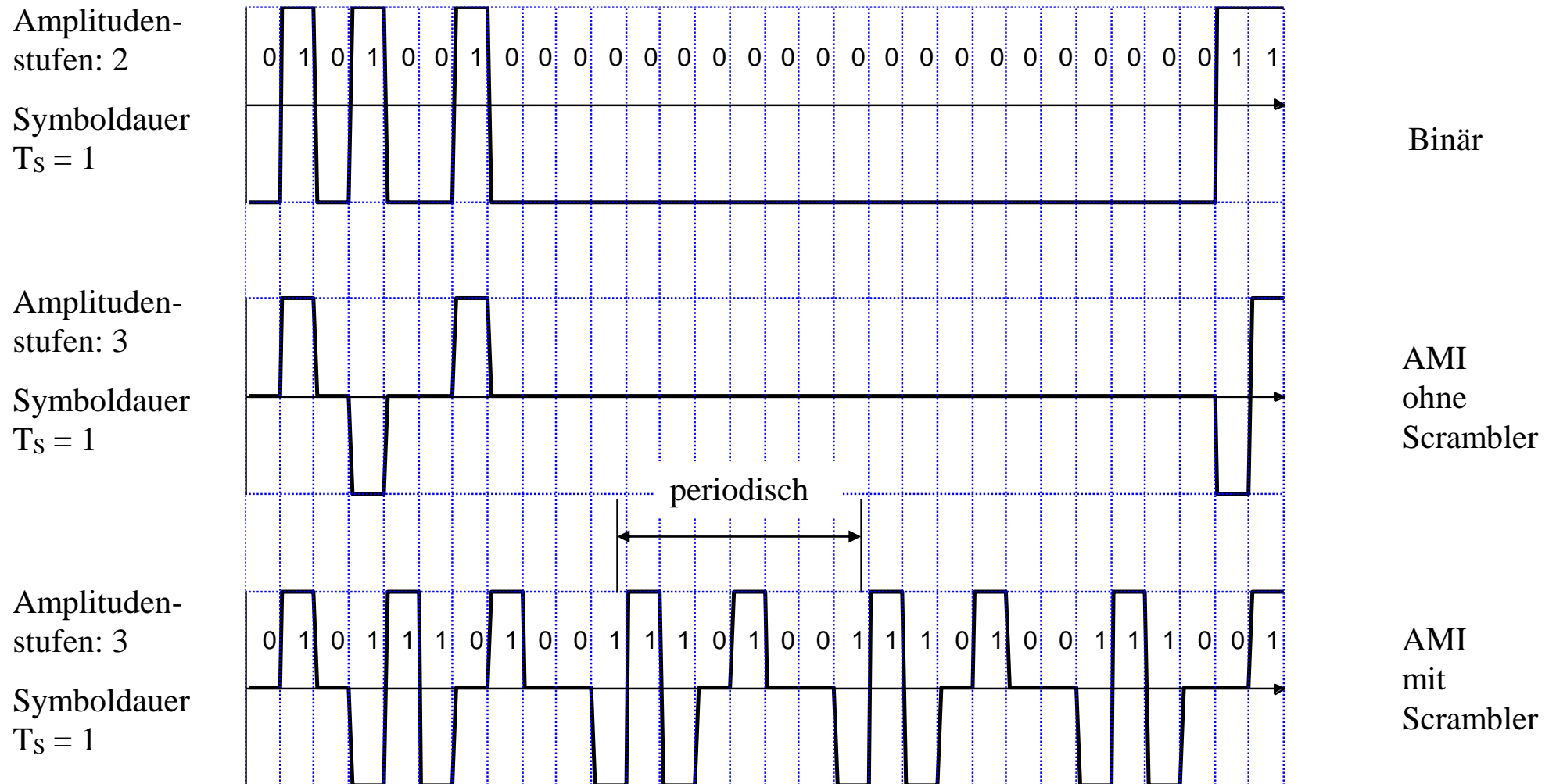
b(n-1)	1	0	0	0	0	1	0	0	0	1	1	0	0	1	0	1	0	1	1	1	1	0	0	0	0	1	0	0	0
b(n-2)	1	1	0	0	0	0	1	0	0	0	1	1	0	0	1	0	1	0	1	1	1	1	0	0	0	0	1	0	0
b(n-3)	0	1	1	0	0	0	0	1	0	0	0	1	1	0	0	1	0	1	0	1	1	1	1	0	0	0	0	1	0
b(n-4)	1	0	1	1	0	0	0	0	1	0	0	0	1	1	0	0	1	0	1	0	1	1	1	1	0	0	0	0	1
b(n-5)	1	1	0	1	1	0	0	0	0	1	0	0	0	1	1	0	0	1	0	1	0	1	1	1	1	0	0	0	0

Empfangsbitfolge

→ Synchronisation erfolgt nach maximal 1 Scramblerdurchlauf

c(n)      0 1 1 0

## Vergleich: AMI mit und ohne Scrambler



## Codes mit Alphabetwechsel

**Prinzip** = Mehrere Alphabete für die Codierung definiert  
 Alphabet wird in Abhängigkeit von laufender digitaler Summe gewählt  
 Codewörter, die nur „0“ enthalten, werden nicht verwendet

**Ziel:** Gleichanteil reduzieren, Redundanz verringern

**Beispiel:** MMS43 (Modified Monitored Sum 4B3T-Code)

**Codierungsgesetz:** Es gibt 4 Alphabete:

	$A_{-1}$	$A_0$	$A_{+1}$	$A_{+2}$
Digitale Wortsumme innerhalb der Alphabete	0 ... +3	-1 ... +1	-1 ... +1	-3 ... 0
Alphabet wird verwendet bei Schlußzustand	-1	0	+1	+2

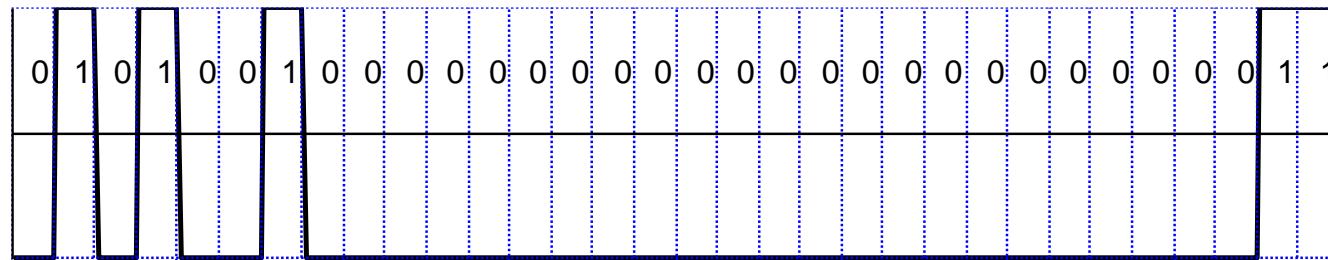
## Codetabelle des MMS43-Codes

Binärwort	Ternärwort			
	A <sub>-1</sub>	A <sub>0</sub>	A <sub>+1</sub>	A <sub>+2</sub>
0 0 1 1			0 - +	
0 1 0 1			- 0 +	
0 1 1 0			- + 0	
1 1 1 0			+ - 0	
1 1 0 1			+ 0 -	
1 0 1 1			0 + -	
1 0 0 0	+ - +	+ - +	+ - +	- - -
1 0 0 1	0 0 +	0 0 +	0 0 +	- - 0
1 0 1 0	0 + 0	0 + 0	0 + 0	- 0 -
1 1 0 0	+ 0 0	+ 0 0	+ 0 0	0 - -
0 1 1 1	- + +	- + +	- - +	- - +
1 1 1 1	+ + -	+ + -	+ - -	+ - -
0 0 0 1	+ + 0	0 0 -	0 0 -	0 0 -
0 0 1 0	+ 0 +	0 - 0	0 - 0	0 - 0
0 1 0 0	0 + +	- 0 0	- 0 0	- 0 0
0 0 0 0	+ + +	- + -	- + -	- + -

## Beispiel: MMS43-Code

Amplituden-  
stufen: 2

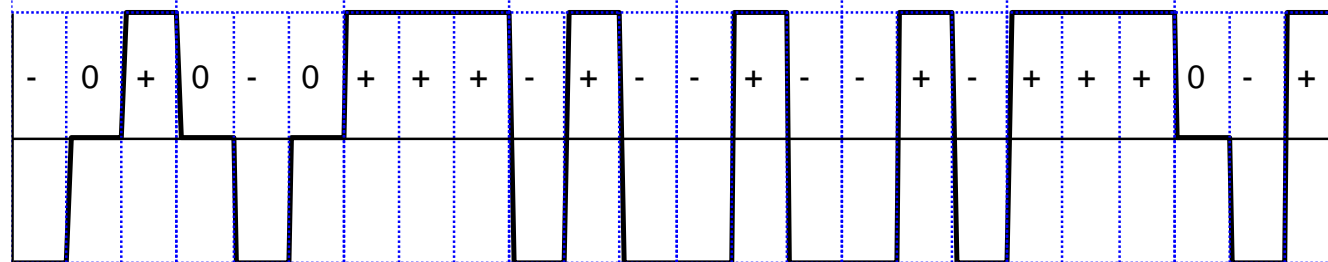
Symboldauer  
 $T_S = 1$



Binär

Amplituden-  
stufen: 3

Symboldauer  
 $T_S = 4/3$



MMS43

Laufende digitale Summe																							
-1	-1	0	0	-1	-1	0	1	2	1	2	1	0	1	0	-1	0	-1	0	1	2	2	1	2
Digitale Wortsumme																							
0				-1			3			-1			-1		-1			3			0		
Schlusszustand																							
0				-1			2			1			0		-1			2			2		
Alphabet																							
A0			A0			A-1			A+2			A+1			A0			A-1			A+2		

## Redundanz

Formel:

$$r = 1 - \frac{m}{n \cdot \log_2(N)}$$

mit  $\log_2$  = Logarithmus zur Basis 2 =  $\log_2$

Beispiele: 1B1T-Code ([HDB3](#))

$m = 1$  (1 Binäres Symbol)

$n = 1$  (1 Ternäres Symbol)

$N = 3$  (3 Amplitudenstufen)

$$r = 1 - \frac{1}{\log_2(3)} = 1 - \frac{\log_{10}(2)}{\log_{10}(3)} = 0,369$$

4B3T-Code ([MMS43](#))

$m = 4$  (4 Binäre Symbole)

$n = 3$  (3 Ternäre Symbole)

$N = 3$  (3 Amplitudenstufen)

$$r = 1 - \frac{4}{3 \log_2(3)} = 0,159$$

$\Rightarrow$  HDB3 „verschwendet“ 37 %, MMS43 „verschwendet“ ca. 16 %

# Partial-Response-Codes

**Prinzip** = Symbol wird aus mehreren Bits der Eingangsbitfolge gewonnen  
→ kontrollierte Intersymbol-Interferenz (Nachbarzeichenbeeinflussung)  
→ Symbol ist mehrstufig, Symbole sind korreliert

**Ziel:** Spektrum formen

**Beispiel:** Duobinärcode

**Codierungsgesetz:**

$$c(n) = a(n) + a(n-1)$$

$c(n)$  = Folge der codierten Symbole  
 $a(n)$  = binäre Quellenfolge

**Beispiel:**

$a(n)$	=	0	1	0	1	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0
$c(n)$	=	0	1	1	1	1	0	1	2	2	2	1	0	0	0	1	2	2	2	1

→  $c(n)$  kann drei Werte annehmen: 0, 1, 2 → ternärer Code

## Weitere Partial-Response-Codes

Beispiel: modifizierter Duobinärcode

Codierungsgesetz:

$$c(n) = a(n) - a(n-2)$$

$c(n)$  = Folge der codierten Symbole  
 $a(n)$  = binäre Quellenfolge

$$\begin{array}{l} a(n) = 0 \ 1 \ 0 \ 1 \ 0 \ 0 \ 1 \ 1 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1 \ 1 \ 1 \ 0 \\ c(n) = 0 \ 1 \ 0 \ 0 \ 0 \ -1 \ 1 \ 1 \ 0 \ 0 \ -1 \ -1 \ 0 \ 0 \ 1 \ 1 \ 0 \ 0 \ -1 \end{array}$$

→  $c(n)$  kann drei Werte annehmen: -1, 0, +1 → ternärer Code

Beispiel: 5-stufiger Partial-Response-Code

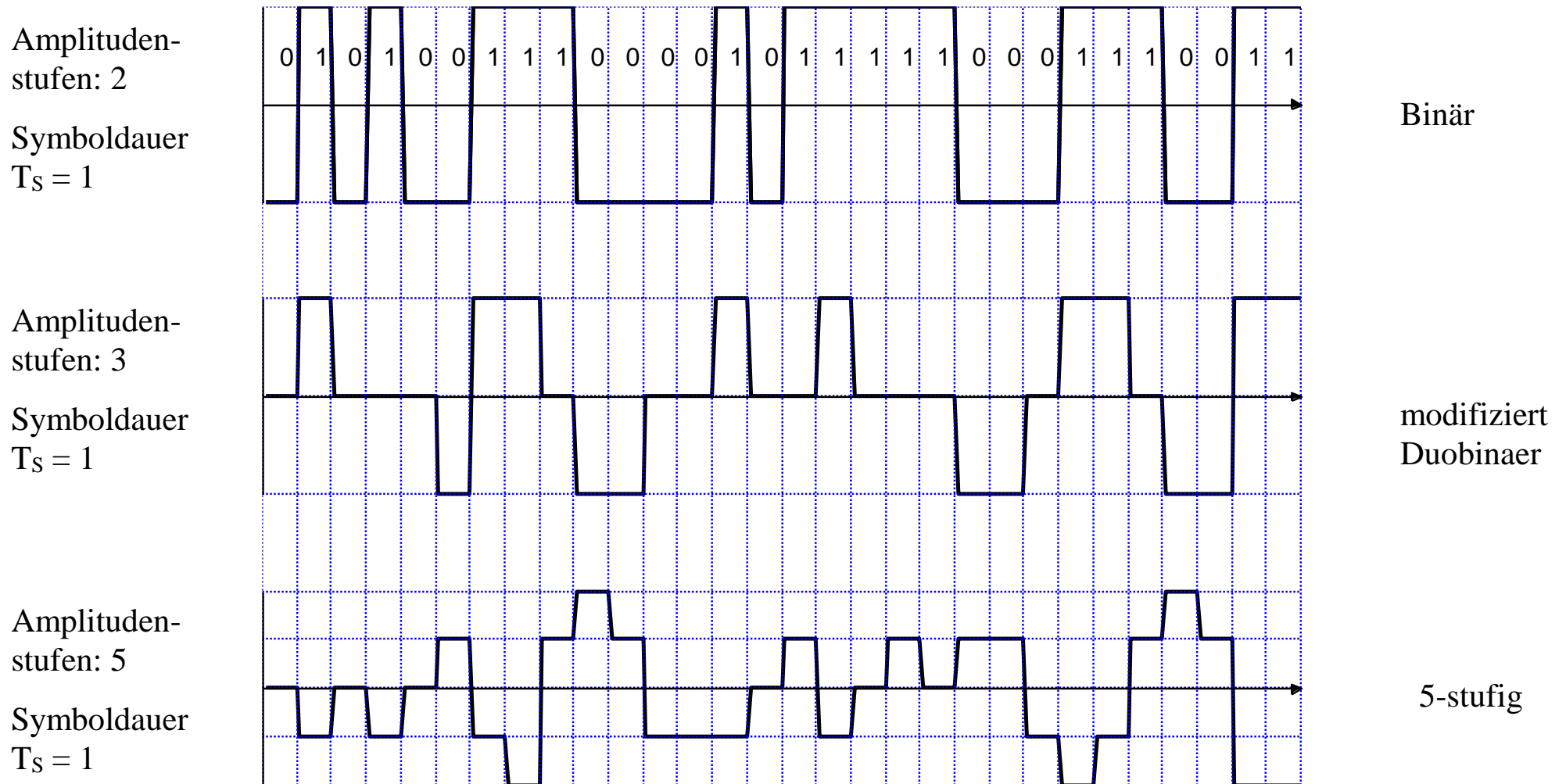
Codierungsgesetz:

$$c(n) = -a(n) + 2 a(n-2) - a(n-4)$$

$$\begin{array}{l} a(n) = 0 \ 1 \ 0 \ 1 \ 0 \ 0 \ 1 \ 1 \ 1 \ 1 \ 0 \ 0 \ 1 \ 0 \ 0 \ 1 \ 1 \ 1 \ 0 \\ c(n) = 0 \ -1 \ 0 \ 1 \ 0 \ 1 \ -1 \ -2 \ 1 \ 1 \ 1 \ 1 \ -2 \ -1 \ 2 \ -1 \ -2 \ 1 \ 2 \end{array}$$

→  $c(n)$  kann fünf Werte annehmen: -2, -1, 0, +1, +2 → 5-stufiger Code

# Vergleich verschiedener Partial-Response-Codes



## Vergleich der Leitungscodes

Code		Stufenzahl N	Redundanz	Symbolrate	Gleichstrom- freiheit
Binär	1B1B	2	0	1	nein
Quaternär	2B1Q	4	0	1/2	nein
Oktonär	3B1O	8	0	1/3	nein
AMI	1B1T	3	36,9 %	1	ja
HDB3	1B1T	3	36,9 %	1	ja
MMS43	4B3T	3	15,9 %	3/4	ja
Modifiziert Duobinär		3	36,9 %	1	ja
5-Stufig Partial Response		5	56,9 %	1	ja

