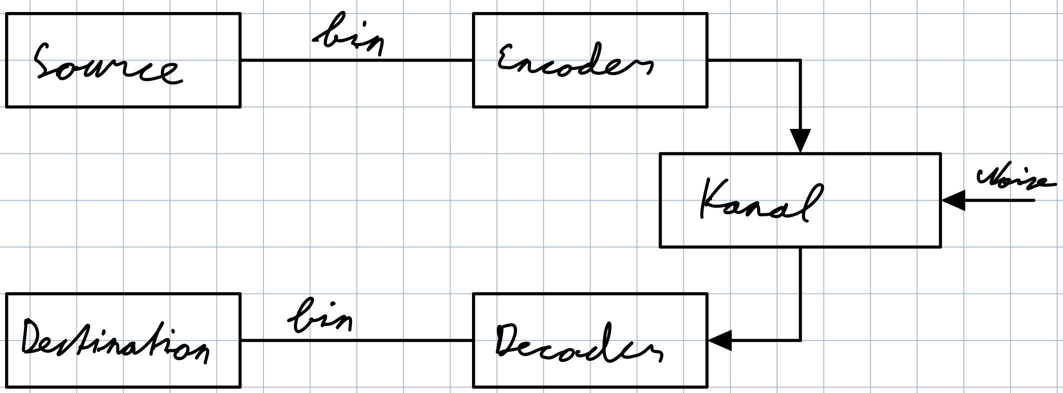



Encoder: fügt zusätzliche Informationen zum Signal hinzu

Decoder: Fehler erkennen und (wenn möglich) korrigieren

## Grundbegriffe:

1) Datenübertragungsrate

Die digital Datenmenge, die innerhalb einer Zeitspanne über einen Übertragungskanal übertragen wird

$c = \frac{D}{t}$  Einheit: Bit/s (bps)

maximal mögliche Datenübertragungsrate die fehlerfrei über einen Kanal übertragen werden kann, wird Kanalkapazität bezeichnet

Bsp: Gigabit Ethernet  125 MBaud → 4 Adernpaare je 2 Bit pro Symbol

$C = 1G$ Bit/s $\neq$ 1G Baud

↳ Symbol 8 Bit

Baudrate, Schrittgeschwindigkeit

Symbolrate

Kanalkapazität:

$$C_{max} = B \cdot \log_2(1 + SNR)$$

B… Bandbreite in Hz

SNR…. Signal to Noise Ratio

$$SNR = \frac{\text{Nutzsignal-Leistung}}{\text{Rauschsignal-Leistung}} \qquad SNR_{dB} = 10 \log(SNR)$$

Diese Gesetzmäßigkeit gilt nur bei weißem Rauschen.

$R \leq C_{max}$ ⇓ Übertragungsrate

$R = f \cdot \log(M)$

f ↑ Symbolrate (Hz, Baud)

M ↳ Anz. Bit / Symbol

Ist R kleiner als Kapazität $C_{max}$, so ist es möglich die Information mit beliebiger Zuverlässigkeit zu übertragen. Ist R größer als $C_{max}$ ist dies nicht mehr möglich und es treten Störungen auf.

Bsp Telefonleitung:

300 – 3400 kHz

56 kBit/s $= C_{max}$

$SNR_{dB} =$

B = 3100 Baud

$$C_{max} = B \cdot \log_2(1 + SNR)$$

$$\frac{C_{max}}{B} = \log_2(1 + SNR)$$

$$2^{\frac{C_{max}}{B}} = 1 + SNR$$

$$2^{\frac{C_{max}}{B}} - 1 = SNR = 274132$$

„E" = Erkennen
„B" = Beheben

# Hamming-Distanz (Hammingabstand)

Maß für die Unterschiedlichkeit zweier Codewörter

Ermittlung: Bit für Bit vergleich, untersch. Stellen zählen.

Bsp:

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| X = | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| Y = | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 1 |
| | x | – | – | x | x | x | x | x |

$\boxed{6}$

Hammingdistanz von X&Y

Bsp: Wiederholungscode $n=5$

Codew. 1 00000

2 11111

$h = 5$ $(n=h)$

| Anzahl | E | B |
|---|---|---|
| 1 | x | x |
| 2 | x | x |
| ≥3 | x | – |

E: $h-1$

B: $r = \frac{h-1}{2}$

Bsp: $n = 6$  Um $r$ Fehler zu korrigieren, muss mind $h=2r+1$

| Anzahl | E | B |
|---|---|---|
| 1 | x | x |
| 2 | x | x |
| 3 | x | – |
| 4 | x | – |
| 5 | x | – |
| 6 | – | – |

Bsp:

W: 10 10 10 10
X: 00 11 10 10
Y: 10 10 00 11
Z: 00 01 00 10

| | W | X | Y | Z |
|---|---|---|---|---|
| W | | 2 | 2 | 4 |
| X | | | 4 | 2 |
| Y | | | | 4 |
| Z | | | | |

$h=2$ (kleinste)

$E = h-1 = 1$

$B = \frac{h-1}{2} = 0{,}5 \approx 0$

Erkennen: 1

Beheben: 0

0
0 0 00

1
1 1 1 1

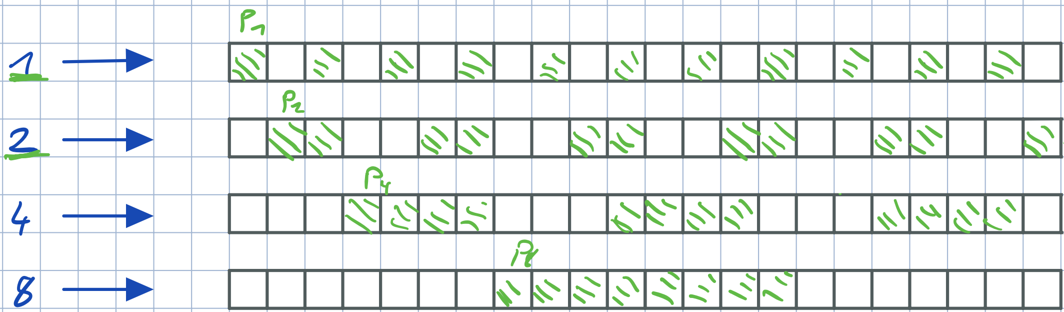
| | A | B | C | D |
|---|---|---|---|---|
| A | – | 0 | 0 | 0 |
| B | – | – | 0 | 0 |
| C | – | – | – | 0 |
| D | – | – | – | – |

| | | $h = 3$ | $h = 4$ |
|---|---|---|---|
| Erkennen: | $h-1$ | 2 | 3 |
| Beheben: | $\frac{h-1}{2}$ | 1 | 1 |

0 1 0 1 1 1 1 gerade even
0 1 0 1 1 0 ung odd

## Hamming-Code

1) Parity-Bits, $2^*$
   (1, 2, 4, 8, 16, … )
2) Rest - Daten
   (3, 5, 6, 7, 9, 10 … )
3) Parity Bits




Bsp:

Hamming Code (7, 4)

1011

| | p1 | p2 | d1 | p4 | d2 | d3 | d4 |
|---|---|---|---|---|---|---|---|
| Nachricht | | | 1 | | 0 | 1 | 1 |
| p1 | 0 | | | | | | |
| p2 | | 1 | | | | | |
| p4 | | | | 0 | | | |
| Codewort | 0 | 1 | 1 | 0 | 0 | 1 | 1 |

Bsp:

0 1 0 1 1 0 1 1

$2^0\, 2^1\, 2^2\, 2^3\, 2^4\, 2^5\, 2^6\, 2^7$

0

0 0 0 1 1 0 1 1 1 0 1 1

1 2 3 4 5 6 7 8 9 10 11 12

$P_1$ $P_2$ $P_4$ $P_8$

0 0 0 1 1 0 0 1 1 0 1 1

$P_1$: ? 0 1 1 1 1 : 0

$P_2$ ? 0 0 1 0 1 : 0

$P_4$: ? 1 0 1 1 : 1

$P_8$: ? 1 0 1 1 : 1

$P_1$: 0 0 1 0 1 1 ✗

$P_2$: 0 0 0 0 0 1 ✗

$P_4$: 1 1 0 0 1 ✗

$P_8$: 1 1 0 1 1 ✓

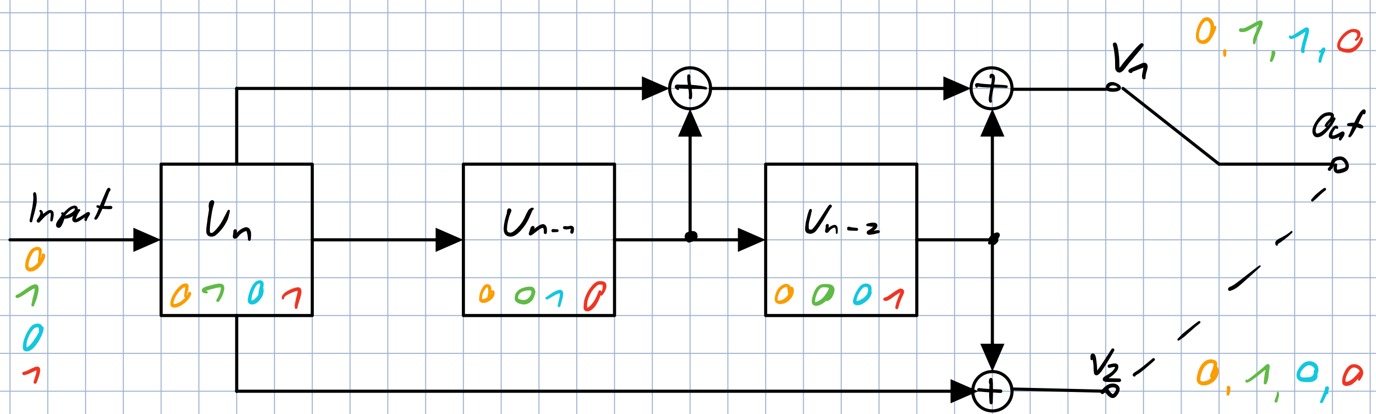
1 + 2 + 4 = 7. Stelle Fehler

## Faltungs Code

| $x_1$ | $x_0$ | y |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

} ExOr = $\oplus$

$U_n \oplus U_{n-1} \oplus U_{n-2} = V_1$



Bsp:

1 0 1 1

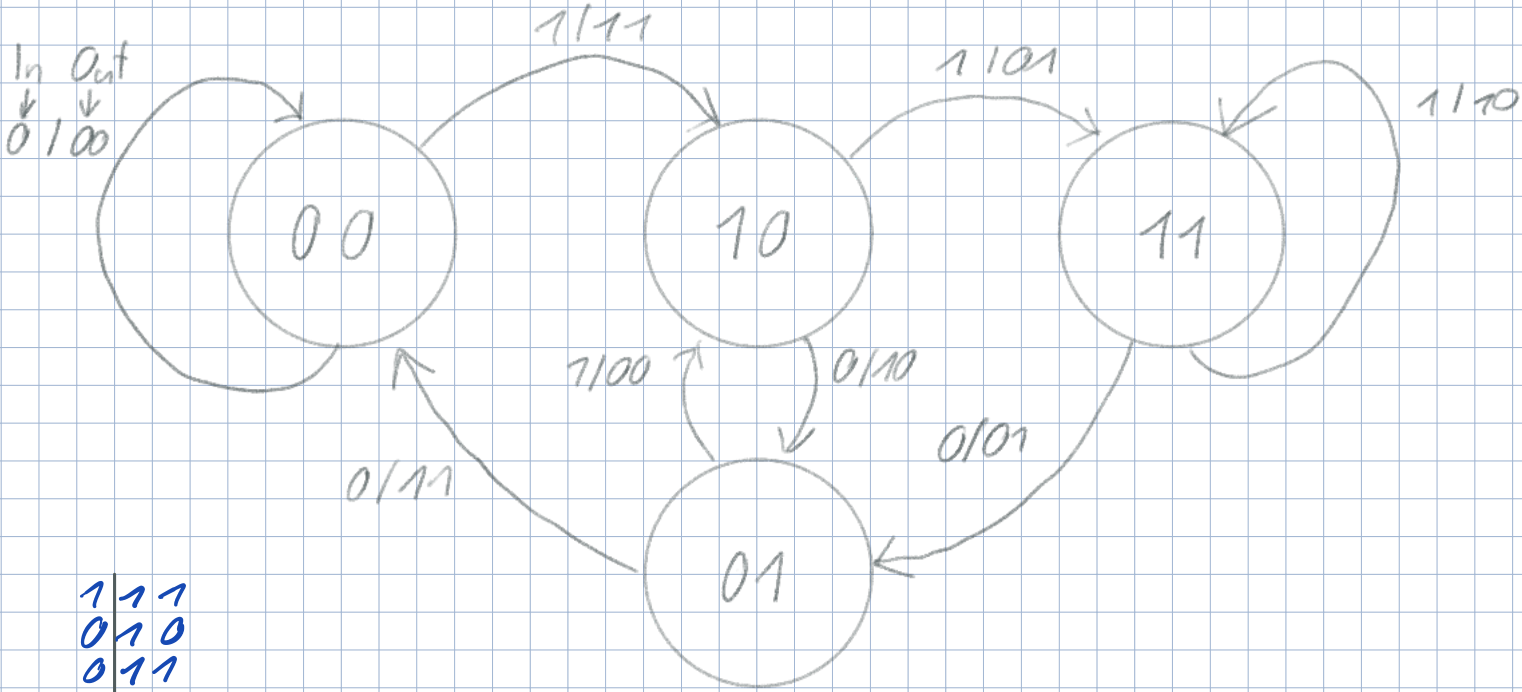
| In | $U_{n-1}$ | $U_{n-2}$ | $V_1$ | $V_2$ | Out |
|---|---|---|---|---|---|
| 1 | 0 | 0 | 1 | 1 | 11 |
| 0 | 1 | 0 | 1 | 0 | 10 |
| 1 | 0 | 1 | 0 | 0 | 00 |
| 1 | 1 | 0 | 0 | 1 | 01 |

$\rightarrow$ 1 1 1 0 0 0 0 1

– 5AHELS –

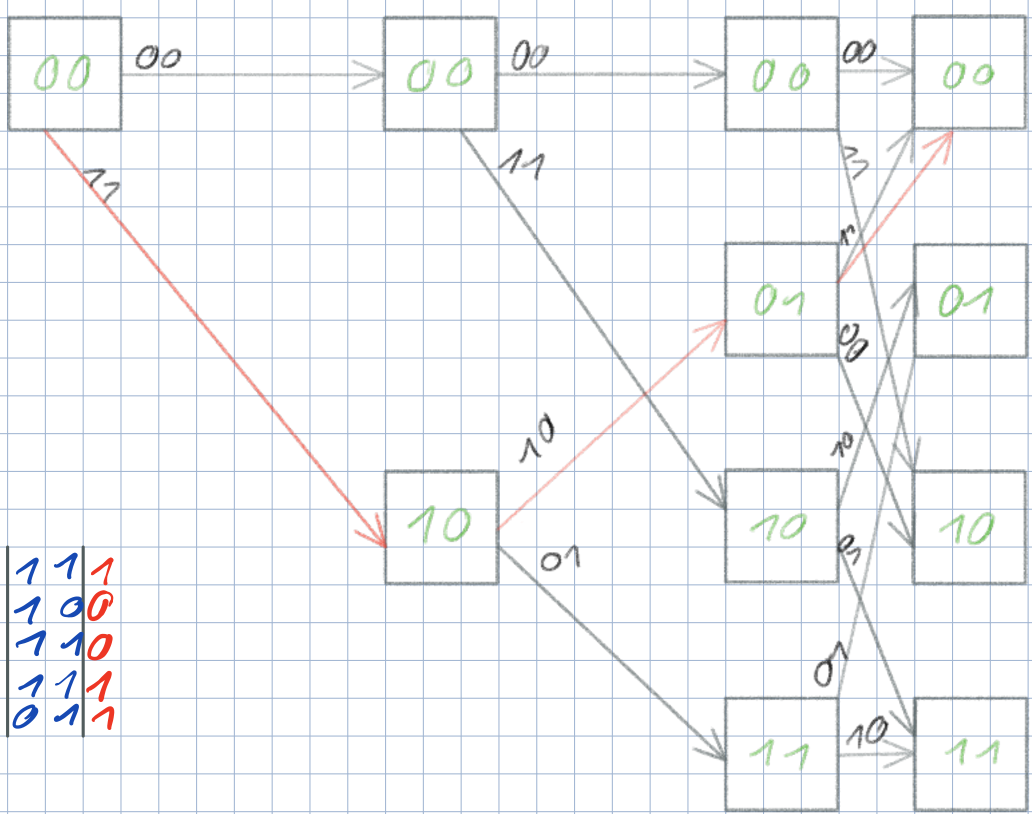
# Faltungscode

## Zustandsdiagramm (State Machine)




| | | |
|---|---|---|
| 1 | 1 | 1 |
| 0 | 1 | 0 |
| 0 | 1 | 1 |
| 1 | 1 | 1 |
| 1 | 0 | 1 |

## Trellis Diagramm




| | | | |
|---|---|---|---|
| 1 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 1 | 1 | 1 |
| 1 | 0 | 1 | 1 |

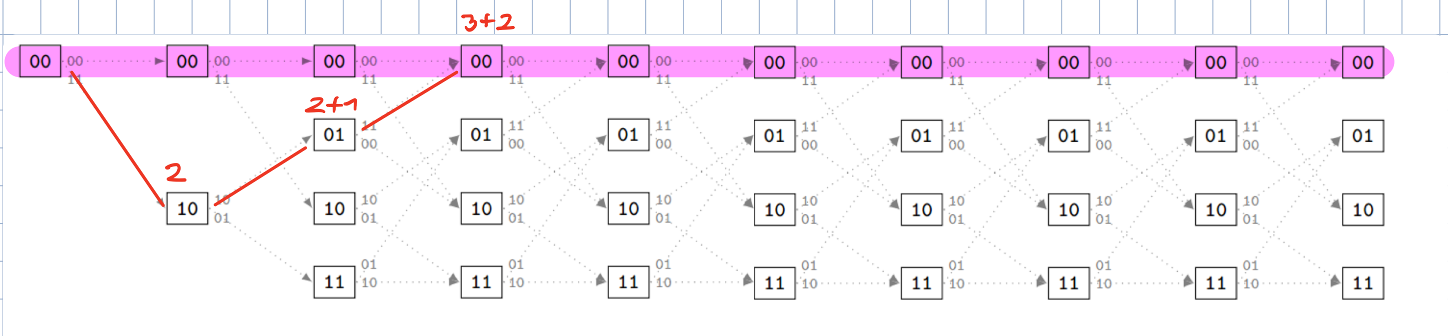
# Vieterbi Bit Decoder

Vergleicht für jeden Zustand die Teilpfade, die zu ihm führen. streicht den Pfad, der verglichen mit der empfangenen Sequenz, die meisten Fehler aufweist. Fehler wird mit Hemming-Distanz gemessen. Wenn 2 Teilpfade gleich viele Fehler haben, spielt es keine Rolle, welchen der beiden er wählt.

**WICHTIG!**

Um das Ende einer Sequenz zu erkennen, werden 2 "Dummy-Nullen" angehängt. Das zwingt den Kodierer am Ende der Sequenz in den Zustand 00 zurück.

5 Hemming-Distanz




0 0 0 0 0 0 0 0 | 0 0 0 0

1 1 1 0 1 1 0 0 | 0 0 0 0

5

$$f = \frac{d-1}{2} = 2$$

Behebbare Fehler

$d_{free}$ ≙ freie Distanz

(min. Distanz zurück zum