

Ziel: 1 Fehler korrigieren!

→ Hammingcode

Datenwort: 0110

Benennung: x_1 x_2 x_3 x_4 x_5 x_6 x_7

Struktur: p_1 p_2 D p_4 D D D

Datenbits: p_1 p_2 0 p_4 1 1 0
anfüllen

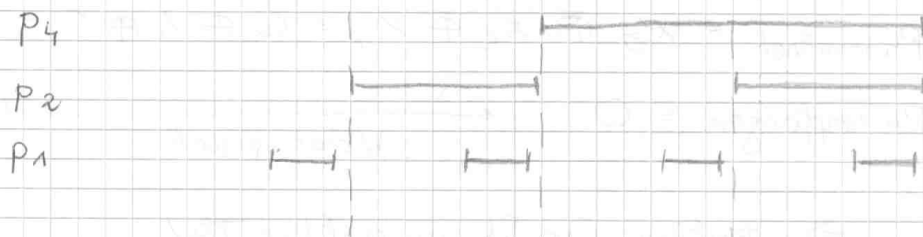
Prüfbits (p_1, p_2, p_4) = ?

Rechenvorschrift: wenn Widerspruch zw.
empfangenen und berechneten Prüfbit
vorliegt, der betroffene Bereich eindeutig
markieren.

Benennung: x_1 x_2 x_3 x_4 x_5 x_6 x_7

Struktur: p_1 p_2 D p_4 D D D

Datenb. anf.: p_1 p_2 0 p_4 1 1 0



$\oplus$ -Verknüpfung (XOR-Verknüpfung)

x	y	$x \oplus y$
0	0	0
0	1	1
1	0	1

Letzte Zeile lautet: 1 1 0

bei ungestörter Übertragung:

p_4, p_2, p_1 so wählen, dass die folgenden Ausdrücke 0 werden:

$$p_4 \oplus x_5 \oplus x_6 \oplus x_7 = 0$$

$$p_2 \oplus x_3 \oplus x_6 \oplus x_7 = 0$$

$$p_1 \oplus x_3 \oplus x_5 \oplus x_7 = 0$$

$$p_4 = x_5 \oplus x_6 \oplus x_7 = 1 \oplus 1 \oplus 0 = 0$$

$$p_2 = x_3 \oplus x_5 \oplus x_7 = 0 \oplus 1 \oplus 0 = 1$$

$$p_1 = x_3 \oplus x_5 \oplus x_7 = 0 \oplus 1 \oplus 0 = 1$$

$\oplus$ -Verknüpfungen sind assoziativ

es wird übertragen: ~~1~~ 1 1 0 0 1 1 0

empfangen: 1 1 0 0 0 1 0

(gestört bei x_5)

Prüfstellen p_4, p_2, p_1 neu berechnen:

$$p_{4 \text{ berechnet}} = x_5 \oplus x_6 \oplus x_7 = 0 \oplus 1 \oplus 0 = 1$$

$$p_{4 \text{ empfangen}} = 0$$

← Widerspruch

⇒ Fehler steckt im rechten Teil

p_1, p_2, p_4 ~~DD~~

$$p_{2 \text{ berechnet}}: x_3 \oplus x_6 \oplus x_7 = 0 \oplus 1 \oplus 0 = 1$$

$$p_2 \text{ empfangen: } 1$$

✓

⇒ Fehler liegt bei p_4

$$p_1 \text{ berechnet: } x_3 \oplus x_5 \oplus x_7 = 0 \oplus 0 \oplus 0 = 0$$

p_1 empfangen: 1 ← Widerspruch ↑

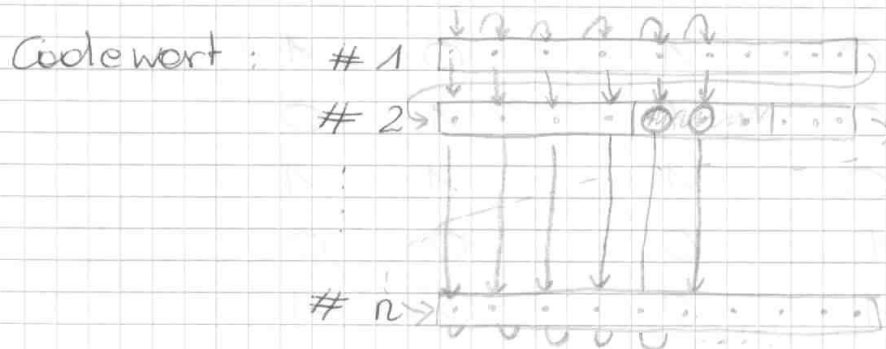
⇒ Ergebnis: $p_1 p_2 D p_4 D D D$

x_5 war verfälscht!

Korrektur durch Invertieren!

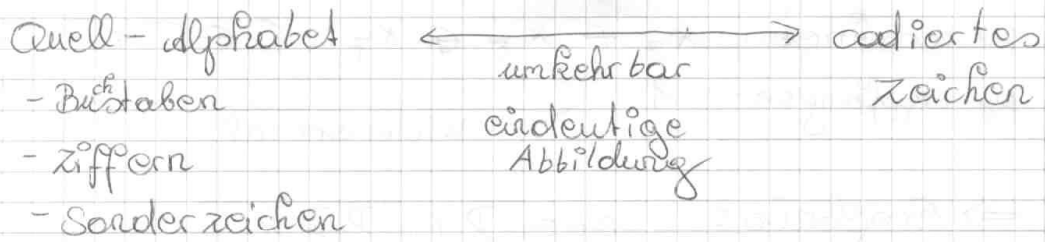
Der Hamming-Code kann nur einen Fehler korrigieren!

→ Oft treten aber sogenannte Burststörungen auf!



Codewörter zwischenspeichern und
spaltenweise übertragen, beim Empfänger
wieder unordnen, dann wird in jedem
Codewort nur ein Bit gestört!

Codierung



BCD - Code (Binary Coded Decimal)

	2^3	2^2	2^1	2^0
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1

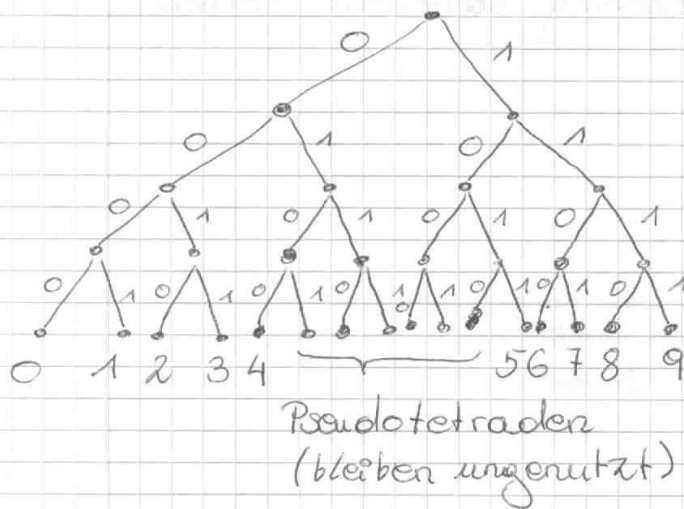
Dezimal - Code:

	2^3	2^2	2^1	2^0
	8	4	2	1
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0

Symmetrielinie

Letzte Zeile lautet: 5 1 0 1 1

Graphik geht auf nächster Seite weiter!

$$\begin{array}{ccccc} 6 & 1 & 1 & 0 & 0 \\ 7 & 1 & 1 & 0 & 1 \\ 8 & 1 & 1 & 1 & 0 \\ 9 & 1 & 1 & 1 & 1 \end{array}$$


Extended Binary Coded Decimal Interchange Code

EBCDI - Code : $2^8 = 256$

Darstellung von alphanumerischer Information nach CCITT No. 2

Übertragung mittels 5 Bits

American Standard Code for Information Interchange

ASCII $\rightarrow$ Code:

7 Bits

1 Byte = 8 Bits

7 Bits + 1 Bit $\Rightarrow$ Parity Check