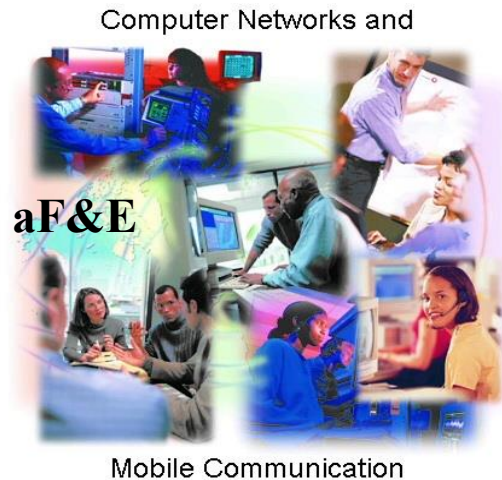


Informations- und Codierungstheorie

4. Blockcodes und zyklische Codes



Prof. Dr.-Ing. Andreas Rinkel
andreas.rinkel@hsr.ch

Sprechstunde: Jeden Montag 16:00 bis 17:00, Raum: 6.110

Tel.: +41 (0) 55 2224928

Mobil: +41 (0) 79 3320562

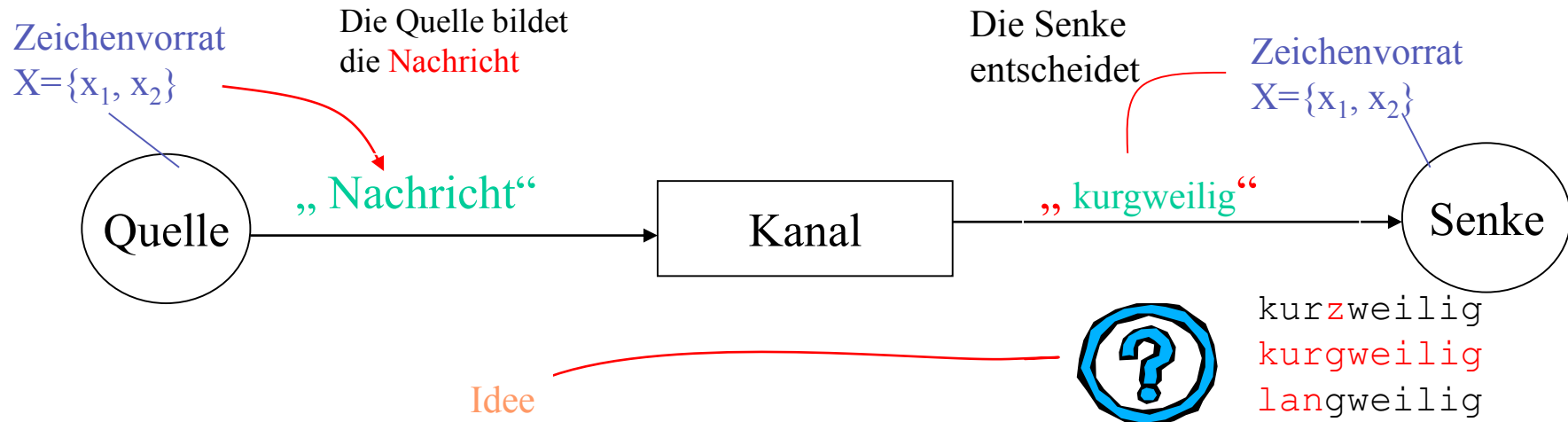
<http://rinkel.ita.hsr.ch>

Blockcodes

- Coderaum
- Hamming Blockcode
- Zyklische Hammingcode
- Abramson Code
- Weitere Codes

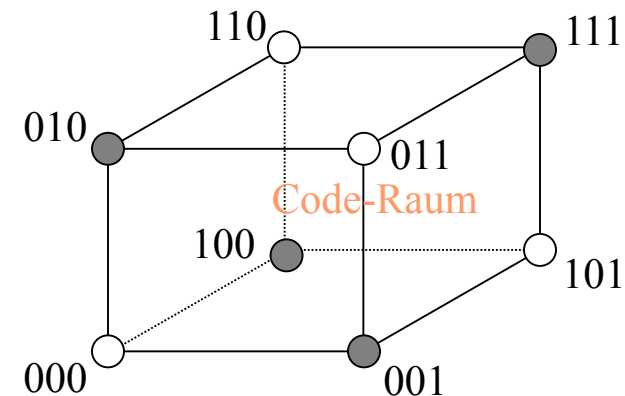
Selbststudium: **Die Modulo-Funktion bei Polynomen aus: Mathematische Grundlagen ohne Balast**

Kanalcodierung Wann können Fehler erkannt werden?



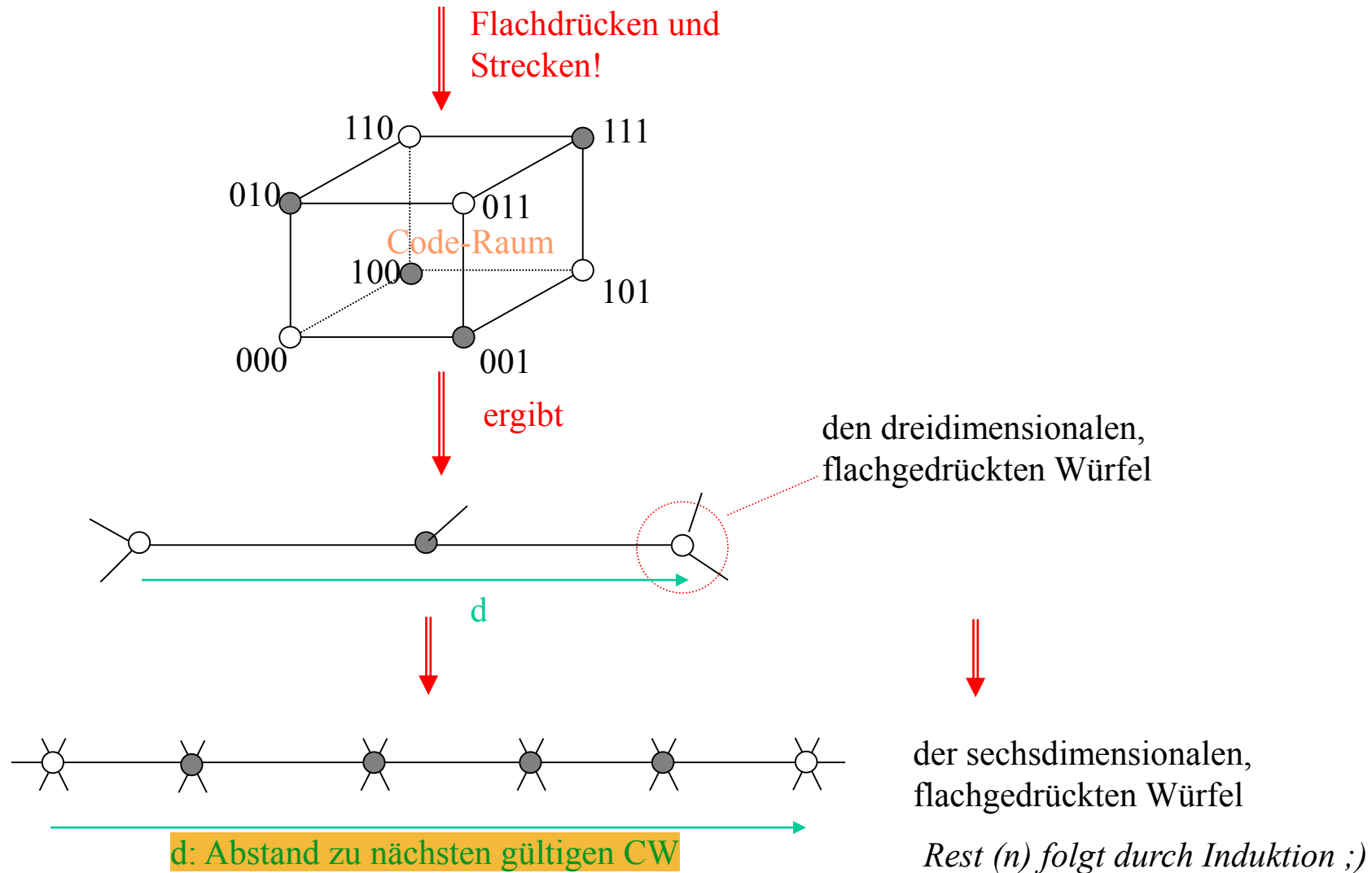
Hinzufügen von Redundanz,
so dass sich der zur Verfügung
stehende Coderaum in gültige und
ungültige
Codeworte (CW: Codewort) aufteilt.

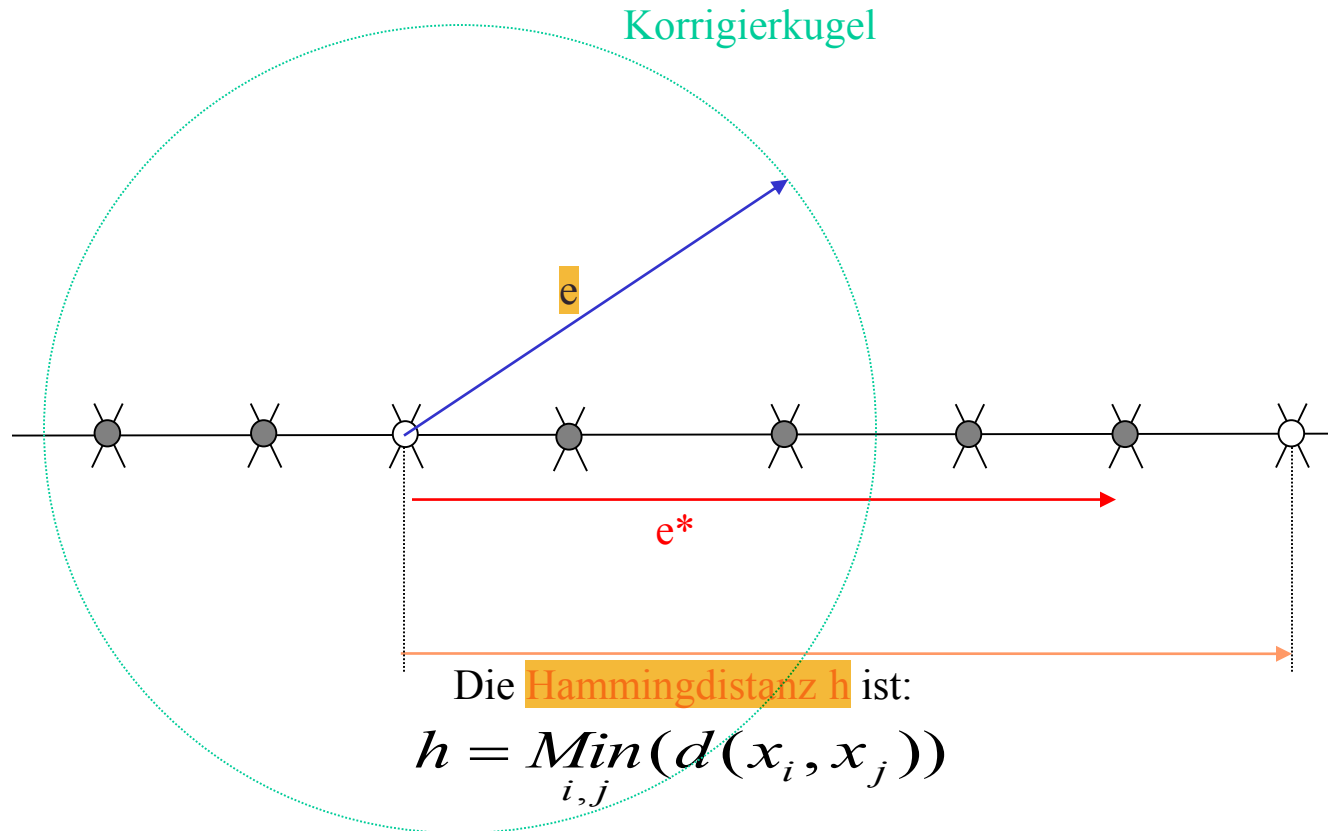
Technisches
Beispiel



1 Fehler kann erkannt werden.

Der n-Dimensionale Coderaum





h = minimaler Abstand Falls
ungerade muss gerechnet
werden

Anzahl der sicher
erkennbaren Fehler

$$e^* = h - 1$$

Anzahl der sicher
korrigierbaren Fehler

➤ h gerade:

$$h = 2e + 2 \Rightarrow$$

$$e = \frac{h - 2}{2}$$

➤ h ungerade:

$$h = 2e + 1 \Rightarrow$$

$$e = \frac{h - 1}{2}$$

Coderaum: *Dichtgepackt* oder nicht, das ist hier die Frage.

Der Coderaum ist *Dichtgepackt*, wenn sich alle Codewörter (gültige und ungültige) in einer Korrigierkugel befinden.

Sei :

- n die Dimension des Code (Anzahl aller CW = 2^n),
- m die Dimension der Nachrichten (Anzahl aller gültigen CW = 2^m)
- k die Dimension der Kontrollstellen mit $n = m + k$

⇒ So folgt die Codeabschätzung:

e = anz. korrigierbarer Fehler

wenn h gerade, nie *Dichtgepackt*

$$2^m \cdot \sum_{w=0}^e \binom{n}{w} \leq 2^n$$

Anzahl der CW bzw. Korrigierkugeln

Anzahl der CW pro Korrigierkugel

Anzahl aller CW

Gilt:

$$2^m \cdot \sum_{w=0}^e \binom{n}{w} = 2^n$$

So ist der Code dichtgepackt!

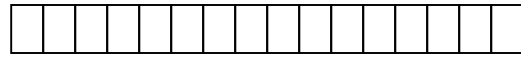
Blockcodes: Einführung

Beispiel: Quersummencode

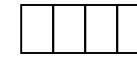
$m=2$ $k=1$

x_1	x_2	x_3
0	0	0
0	1	1
1	0	1
1	1	0
0	0	1
0	1	0
1	0	0
1	1	1

m Nachrichtenstellen



k Kontrollstellen



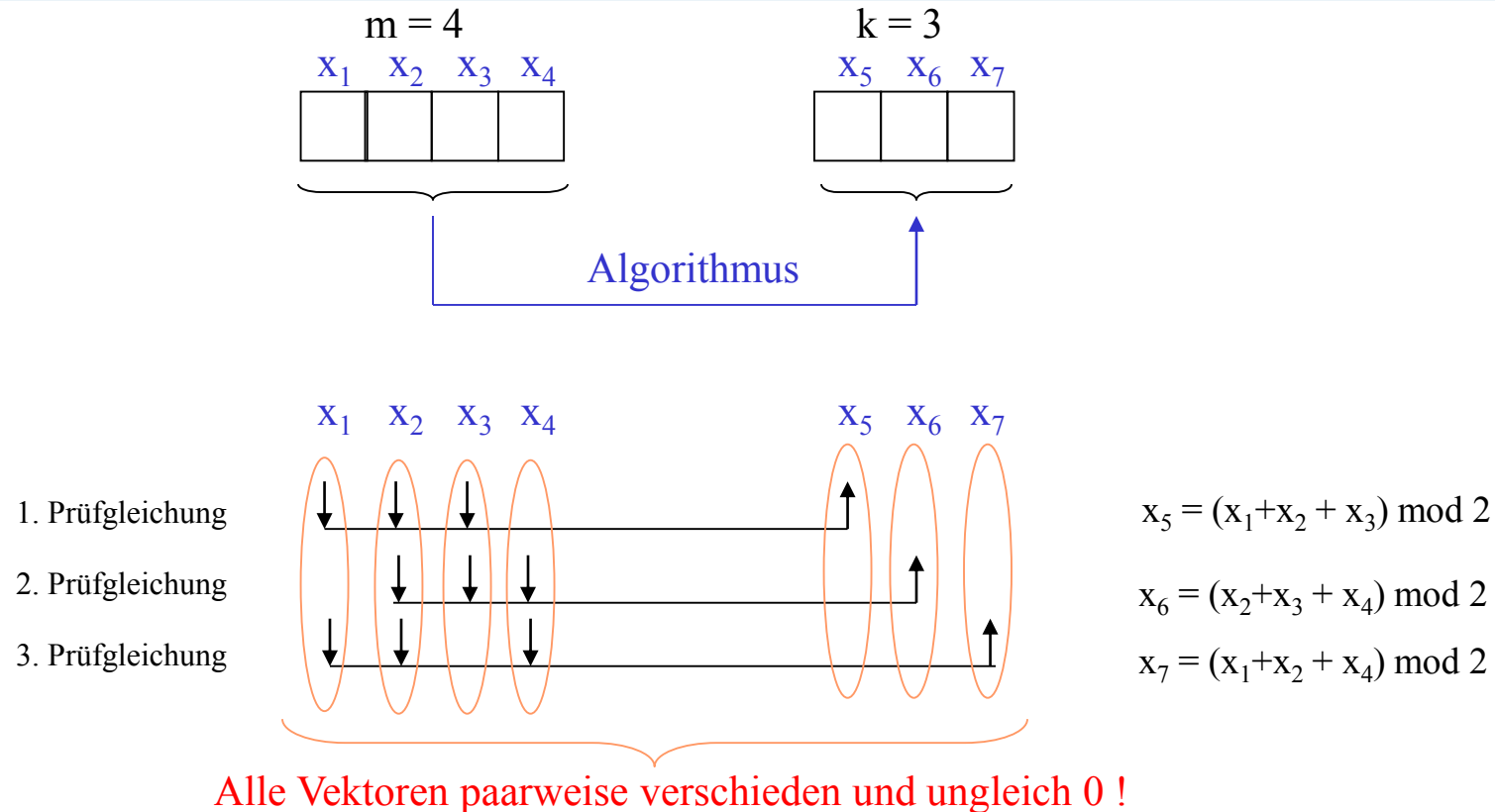
Algorithmus

Gültige Codeworte,
sie erfüllen den Algorithmus

Ungültige Codeworte,
sie erfüllen den Algorithmus nicht,
d.h. sie liefern ein *Fehlermuster*

Algorithmus zur Berechnung
der Kontrollstellen
$$x_3 = (x_1 + x_2) \bmod 2$$

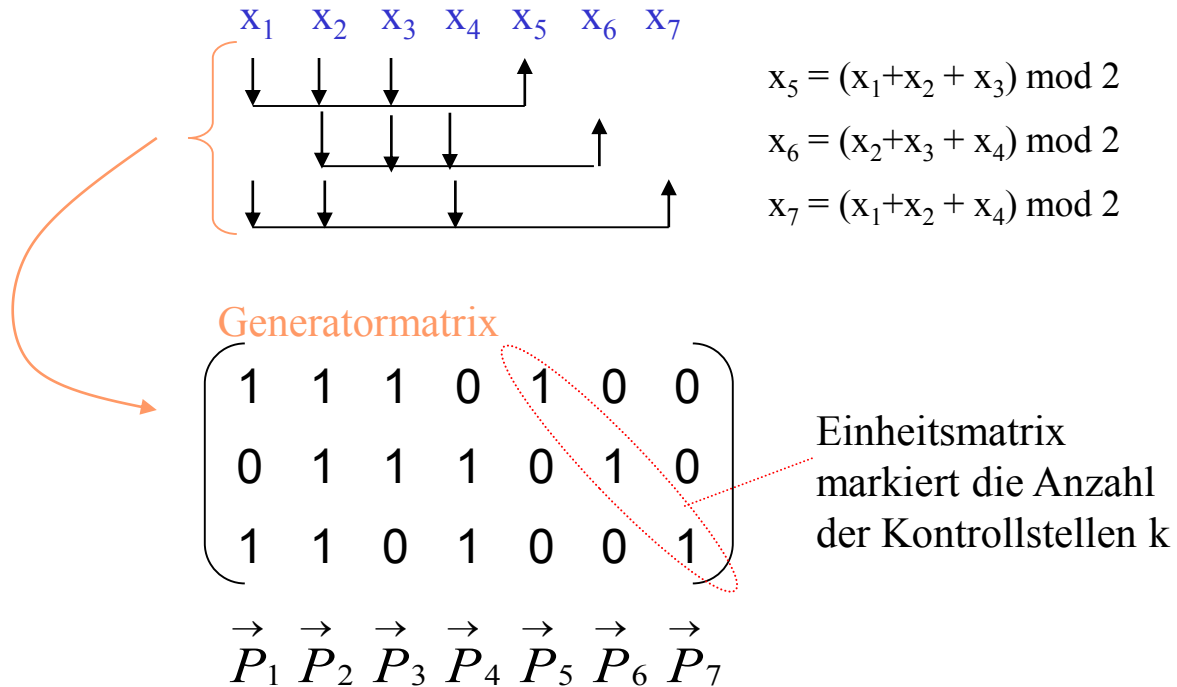
Blockcodes: Hamming-Code I



Interpretation: wird eine Stelle des CW verletzt, so werden jeweils andere Kombinationen von Prüfgleichungen verletzt, d.h. es müsste ein Fehlersyndrom geben, dass es erlaubt, den Fehlerort zu lokalisieren.

Frage: wie viele Fehler können nicht mehr erkannt werden? >2

Blockcodes: Hamming-Code II



Hieraus folgt
die Codebedingung:

$$\sum_i x_i \cdot \vec{P}_i \equiv \vec{0} \bmod 2$$

Blockcodes: Hamming-Code III

Tabelle der gültigen Codeworte (Prüfung!!)

x1	x2	x3	x4	x5	x6	x7
0	0	0	0	0	0	0
0	0	0	1	0	1	1
0	0	1	0	1	1	0
0	0	1	1	1	0	1
0	1	0	0	1	1	1
0	1	0	1	1	0	0
0	1	1	0	0	0	1
0	1	1	1	0	1	0
1	0	0	0	1	0	1
1	0	0	1	1	1	0
1	0	1	0	0	1	1
1	0	1	1	0	0	0
1	1	0	0	0	1	0
1	1	0	1	0	0	1
1	1	1	0	1	0	0
1	1	1	1	1	1	1

Nachrichtenstellen

Kontrollstellen

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

Hammingdistanz anhand der Matrix

Die Codebedingung:

$$\sum_i x_i \cdot \vec{P}_i \equiv \vec{0} \mod 2$$

Wird für alle gültigen Codeworte (Tabelle) erfüllt.

Was ergibt die Berechnung der Codebedingung bei einem Bitfehler?

$$x_5 = (x_1 + x_2 + x_3) \mod 2$$

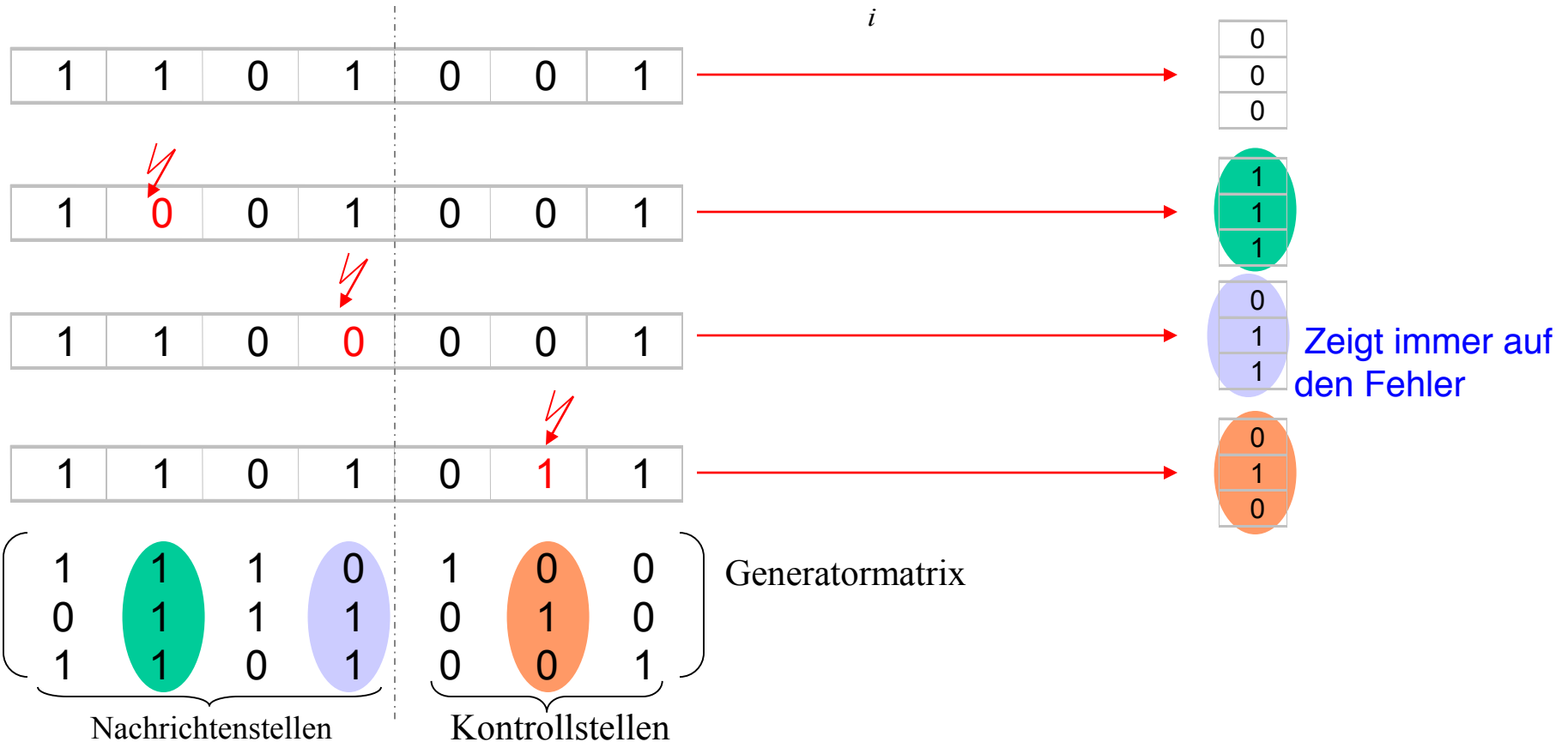
$$x_6 = (x_2 + x_3 + x_4) \mod 2$$

$$x_7 = (x_1 + x_2 + x_4) \mod 2$$

Blockcodes: Hamming-Code IV

Das Syndrom Z:

$$\vec{Z} = \sum_i x_i \cdot \vec{P}_i \text{ mod } 2$$



Hamming-Code: Das Fehlersyndrom

Gesendetes
Codewort

$$X = [x_1, x_2, x_3, \dots, x_n]$$

Überlagert durch
das Fehlermuster

$$F = [f_1, f_2, f_3, \dots, f_n]$$



Empfangenes Wort

$$X' = X + F$$

$$= [x_1 + f_1, x_2 + f_2, x_3 + f_3, \dots, x_n + f_n] \bmod 2$$

$$= [x'_1, x'_2, x'_3, \dots, x'_n]$$

Aus der Codebedingung
folgt das Syndrom

$$\vec{Z} = \sum_i x'_i \cdot \vec{P}_i = \sum_i (x_i + f_i) \cdot \vec{P}_i$$

$$= \sum_i x_i \cdot \vec{P}_i + \sum_i f_i \cdot \vec{P}_i$$

Codebedingung = 0

$$\Rightarrow \vec{Z} = \sum_i f_i \cdot \vec{P}_i$$

Das heisst, bei genau einem Fehler
markiert die Prüfspalte den Fehlerort.

Zyklische Codes: Mathematische Beschreibung

Idee: Generatormatrix kann durch Generatorpolynom beschrieben werden!

Ziel: Vereinfachte Berechnung der Kontrollstellen durch rückgekoppelte Schieberegister.

Generatorpolynom $G(u)$

$$G(u) = \sum_{i=0}^k g_i \cdot u^i$$

Codewortpolynom $X(u)$

$$X(u) = \sum_{i=0}^n g_i \cdot u^i$$

Codebedingung

Das Codewortpolynom ist ohne Rest durch das Generatorpolynom teilbar
(in mod-2-Rechnung)

$$X(u) \div G(u) \equiv Q(u) \text{ mod } 2$$

$$X(u) \equiv Q(u) \cdot G(u) \text{ mod } 2$$

$$g_i \in \{0,1\} \text{ mit } g_0 = g_k = 1$$

Grad **k** entspricht der Anzahl der Prüfstellen.
Grad **n** entspricht der Anzahl der Codewortstellen.
Die Zahl der Nachrichtenstellen ist **m**
 $\Rightarrow n = m + k$

Zyklische Codes: Ermittlung der Kontrollstellen durch Polynomdivision

Sei: $m = 4, k = 3, n = 7$
 Nachricht: $(x_1, x_2, x_3, x_4) = (1 \ 0 \ 0 \ 0)$
 Generator: $G(u) = u^3 + u + 1 \Rightarrow (g_3 \ g_2 \ g_1 \ g_0) = (1 \ 0 \ 1 \ 1)$

u^6	u^5	u^4	u^3	u^2	u^1	u^0		u^3	u^2	u^1	u^0		u^3	u^2	u^1	u^0	
1	0	0	0	1	0	1	:	1	0	1	1	\equiv	1	0	1	1	mod 2
1	0	1	1														
%	0	1	1														
	0	0	0	0													
%		1	1	0													
		1	0	1	1												
%			1	1	1												
			1	0	1	1											
%				1	0	1											

101 sind die gesuchten Kontrollstellen, die die Codebedingung erfüllen.

101 sind die gesuchten Kontrollstellen, die die Codebedingung erfüllen.

Zyklische Codes: Ermittlung der Kontrollstellen durch Mehrfachaddition

Sei: $m = 4, k = 3, n = 7$
 Nachricht: $(x_1, x_2, x_3, x_4) = (1 \ 0 \ 0 \ 0)$
 Generator: $G(u) = u^3 + u + 1 \Rightarrow (g_3 \ g_2 \ g_1 \ g_0) = (1 \ 0 \ 1 \ 1)$

Idee: $X(u)$ ist durch $G(u) \bmod 2$ teilbar, also muss $X(u)$ durch Addition von $G(u) \bmod 2$ erzeugbar sein!

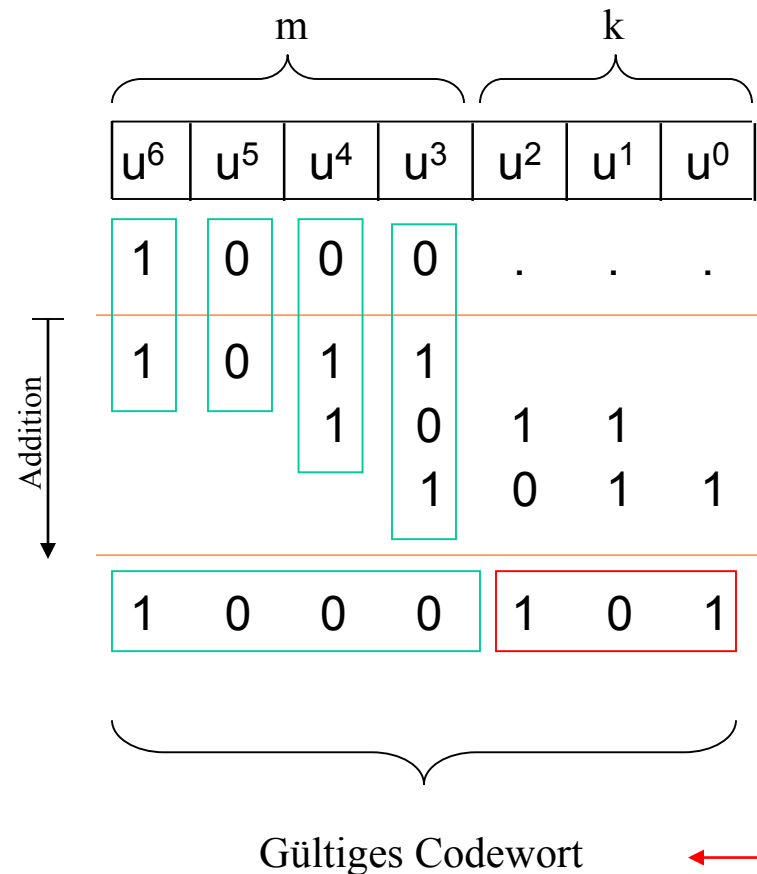
Addition des ersten Terms $G(u)$ erzeugt die Stellen u^6, u^5 von $X(u)$

Addition des zweiten Terms $G(u)$ erzeugt die Stelle u^4 von $X(u)$

Addition des dritten Terms $G(u)$ erzeugt die Stelle u^3 von $X(u)$

Der Rest muss nach Codebedingung die Kontrollstellen bilden!

Gültiges Codewort



Zyklische Codes: Prüfen der Codebedingung

Empfangenes

Codewort:

1 0 0 0 1 0 1

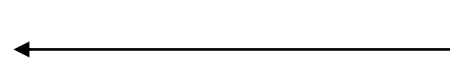
1 0 1 1

1 0 1 1

1 0 1 1

0 0 0 0 0 0 0

Generator: 1 0 1 1



Code- Bedingung erfüllt!

Idee:

Durch die Codebedingung muss die fortgesetzte Addition (mod 2) des Generators zum empfangenen CW Das Nullwort ergeben.

$$X(u) \div G(u) \equiv Q(u) \text{ mod } 2$$

$$X(u) \equiv Q(u) \cdot G(u) \text{ mod } 2$$

Empfangenes

Codewort:

1 0 0 **1** 1 0 1

1 0 1 1

1 0 1 1

0 0 0 0 0 1 1

Code- Bedingung **nicht** erfüllt!

Fehlersyndrom



Zyklischer Hamming- Code und Generatormatrix ?

Gültiges Codewort: 1 0 0 0 1 0 1

$$\begin{array}{r} \text{⚡} \\ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 1 \\ \hline 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 1 \end{array}$$

$$\begin{array}{r} \text{⚡} \\ 1 \ 1 \ 0 \ 0 \ 1 \ 0 \ 1 \\ 1 \ 0 \ 1 \ 1 \\ 1 \ 0 \ 1 \ 1 \\ \hline 1 \ 0 \ 1 \ 1 \\ 0 \ 0 \ 0 \ 0 \ 1 \ 1 \ 1 \end{array}$$

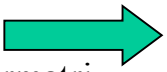
$$\begin{array}{r} \text{⚡} \\ 1 \ 0 \ 1 \ 0 \ 1 \ 0 \ 1 \\ 1 \ 0 \ 1 \ 1 \\ 1 \ 0 \ 1 \ 1 \\ \hline 0 \ 0 \ 0 \ 0 \ 1 \ 1 \ 0 \end{array}$$

$$\begin{array}{r} \text{⚡} \\ 1 \ 0 \ 0 \ 1 \ 1 \ 0 \ 1 \\ 1 \ 0 \ 1 \ 1 \\ 1 \ 0 \ 1 \ 1 \\ \hline 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1 \end{array}$$

$$\begin{array}{r} \text{⚡} \\ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \\ 1 \ 0 \ 1 \ 1 \\ 1 \ 0 \ 1 \ 1 \\ 1 \ 0 \ 1 \ 1 \\ \hline 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \end{array}$$

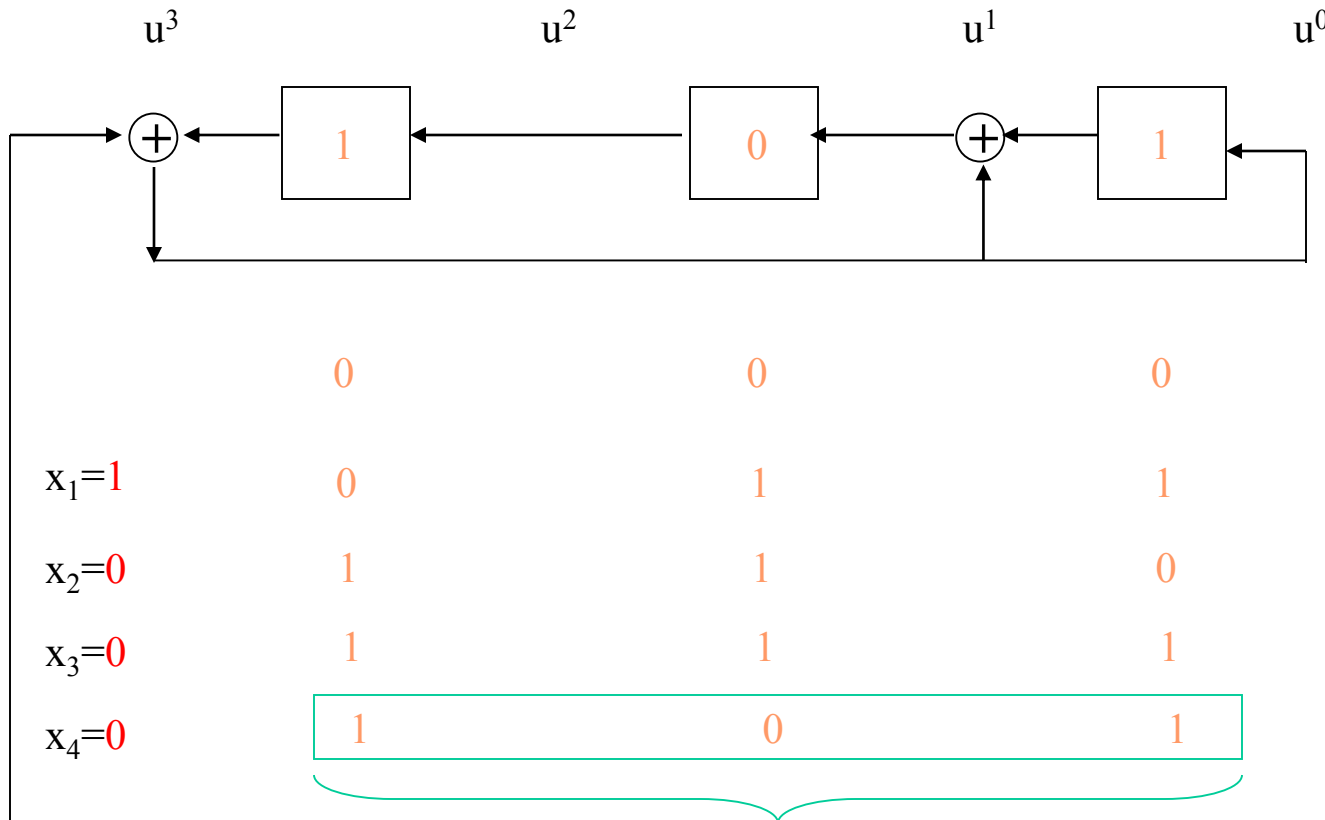
$$\begin{array}{r} \text{⚡} \\ 1 \ 0 \ 0 \ 0 \ 1 \ 1 \ 1 \\ 1 \ 0 \ 1 \ 1 \\ 1 \ 0 \ 1 \ 1 \\ 1 \ 0 \ 1 \ 1 \\ \hline 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \end{array}$$

$$\begin{array}{r} \text{⚡} \\ 1 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \\ 1 \ 0 \ 1 \ 1 \\ 1 \ 0 \ 1 \ 1 \\ 1 \ 0 \ 1 \ 1 \\ \hline 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \end{array}$$

Generatormatrix 

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

Zyklische Codes: Ermittlung der Kontrollstellen durch rückgekoppeltes Schieberegister



Ermittelten Kontrollstellen

$\Leftarrow G(u)$

Sei: $m = 4, k = 3, n = 7$

Nachricht: $(x_1, x_2, x_3, x_4) = (1 \ 0 \ 0 \ 0)$

Generator: $G(u) = u^3 + u + 1$

Vorbelegung

Nach Übernahme x_1

Nach Übernahme x_2

Nach Übernahme x_3

Nach Übernahme x_4

\oplus Modulo 2 Addierer (XOR)

Zyklische Hamming-Codes:

Hammingdistanz $h=3$

Diese werden gebildet durch sogenannte primitive Polynome $p(x) = g(x)$:

$$p(x) = 1+x+x^3$$

$$p(x) = 1+x+x^4$$

$$p(x) = 1+x^2+x^5$$

$$p(x) = 1+x+x^6$$

$$p(x) = 1+x^3+x^7$$

$$p(x) = 1+x^2+x^3 + x^4+x^5+x^6+x^7$$

$$p(x) = 1+x^2+x^3 + x^4+x^5+x^8$$

$$p(x) = 1+x^4+x^9$$

$$p(x) = 1+x^3+x^{10}$$

$$p(x) = 1+x^2+x^{11}$$

$$p(x) = 1+x + x^4+x^6+x^{12}$$

$$p(x) = 1+x+x^3 + x^4+x^{13}$$

$$p(x) = 1+x^2+x^6+x^{10}+x^{14}$$

$$p(x) = 1+x+x^{15}$$

$$p(x) = 1+x^5+x^{23}$$

$$p(x) = 1+x+x^2+x^4+x^5 + x^7+x^8+x^{10}+x^{11}+x^{12}+x^{16} + x^{22}+x^{23}+x^{26}+x^{32}$$

Zyklische Abramson-Codes bzw. CRC-Codes:

Hammingdistanz $h=4$

Diese werden gebildet durch die Multiplikation eines primitiven Polynoms mit dem Term $(1+x)$

Abramson-Code: $g(x) = p(x)(1+x)$

Bsp.:

$$g(x) = (1+x+x^3)(1+x)$$

$$g(x) = 1+x^2+x^3+x^4$$