Fehlerkorrektur

Gliederung

- Kanalstörungen
- Einfache Verfahren
- Hamming-Abstand
- Technische Schaltungen
- Binäre Arithmetik
- Matrizenrechnung
- Typische Codes

Fehlertypen

Merksätze:

- Alle Fehler sind statistisch
- Alle Aussagen über Fehlergrößen sind auch statistisch
- Fehlerwerte multiplizieren sich etwa

Wenn 1-Bit-Fehler etwa 10^{-3}
dann 2-Bit-Fehler etwa 10^{-6}
und 3-Bit-Fehler etwa 10^{-9}
usw.

Folien/92/1.5.92;ECC2.txt

Fehlerarten

Alle Aussagen werden auf einen Block bezogen

- **Synchron-Fehler**, der Wortanfang wir nicht erkannt oder Empfänger verliert den Takt. Dieser Fehler ist nicht korrigierbar und wird daher bei der Fehlererkennung nicht betrachtet
- ***1-Bit*-Fehler** bedeutet: in einem Block tritt genau ein Fehler auf, es ist nur die Position zu bestimmen, und dieses Bit ist zu invertieren
- ***n-Bit*-Fehler** bedeutet: in einem Block treten maximal *n* Fehler auf; sie liegen irgendwo verteil im Block; es ist zu bestimmen: die Anzahl m ≤ n; ihre Positionen und alle diese Bit sind zu invertieren
- ***Burst*** der Länge *b* bedeutet: in einem Block gibt es eine zusammenhängende Bitfolge, der maximalen Länge $b < n$, hierin können die einzelne Bit falsch oder richtig sein, es ist zu bestimmen: der Beginn des Burst; die relativen Positionen der fehlerhaften Bits

Einzelfehler besitze die Wahrscheinlichkeit *p*. Es gelte Unabhängigkeit der Fehlereinflüsse
Für ein Wort der Länge *n* gelte noch:

$$p_n = (1-p)^n.$$

Für die Summe aus Einzel-, Doppel-, Dreifach-Fehlern usw., für *l*-fach-Fehler gilt:

$$p_{kl} = \sum_{1}^{n} p^l \cdot (1-p)^{n-1}$$

Folglich gilt für die Summe aller Fehler

$$p_k = \sum_{l=1}^{n} \sum_{l}^{n} p^l \cdot (1-p)^{n-l}$$

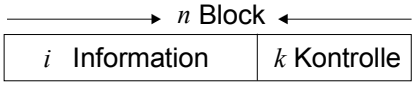
Folien/92/1.5.92;ECC2.txt

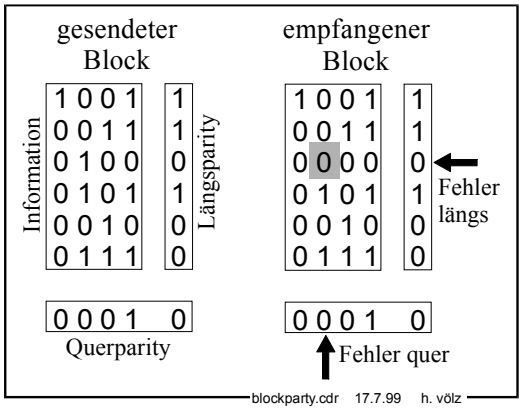
4 wichtige Methoden

A Parity gezählt werden die "1" mod 2; zusätzliche "1" oder "0" unterscheiden gerade / ungerade Parity
mehrere Parity-Bit möglich; erkennen dann mehrere Fehler, korrigiert wesentlich weniger
Blockparity *x* : *y* korrigiert im Block genau einen Fehler
B gleiches Gewicht genau festgelegte Anzahl von "1" in der Verteilung liegt die Information
gültig: 110011, 101011, 110110 usw., evtl. auch: 10101010, 1111 usw.
C Symmetrie gültig: 110011, 101101, 011110 usw.
D komplexe Verfahren

Folien/92/1.5.92;ECC1.txt

Standardblock besteht aus (Bezeichnung unterschiedlich in Literatur):
n = Blocklänge
i = Informationsbit
k = Kontrollbit





Fehlertolerante Code

Gesetz der großen Zahl: um **Fehlerrate** von 10^{-6} zu verifizieren sind mindestens 10^{10} Wörter testen. Geht oft aus Zeitgründen nicht

8 Ecken eines **Drahtwürfels** ermöglichen
- 8 gültige Wörter mit 1-Bit-Abstand.
- 4 gültige Wörter mit 2-Bit-Abstand, ermöglichen eine Erkennung von 1-Bit-Fehlern.
- 2 gültige Wörter mit 3-Bit-Abstand, ermöglichen eine Erkennung von 2-Bit-Fehlern, oder eine Korrektur von 1-Bit-Fehlern.

Problem ist bei *n*-Bit-Wörtern die Auswahl der gültigen Wörter so, daß der **Hamming-Abstand** maximal ist. Das nächste Problem ist dann die Auswahl des Verhältnisses von **Fehlerkorrektur und Fehlererkennung.**
Hamming-Abstand *d*
erkennbare Fehler $e_{max.} = d\text{-}1$.
maximal korrigierbare Fehler c_{max} = INT((*d*-1)/2)
Sollen nur $c < c_{max}$ korrigiert werden, so sind noch $e(d, c) = d\text{-}c\text{-}1$ erkennbar.

Linear-Codes: es existieren m linear unabhängige Generatorpolynome G_1 bis G_m. Alle Code-Wörter x_i der Länge *m* werden durch Linearkombination gebildet:

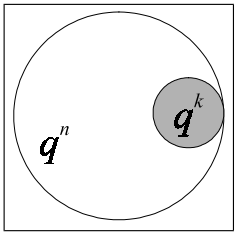
$$X = x_1 \cdot G_1 + x_2 \cdot G_2 + \ldots + x_m \cdot G_m$$

Linear systematischen Code zusätzlich definierte Anordnung der Informations- und Kontroll-Bit
Zyklischen Codes Erzeugerpolynom *G*(x), dem eine der Redundanz entsprechende Anzahl von Nullen vorn angefügt wird, dann zyklische Vertauschen erzeugt die Generator-Wörter.
(*n*,*m*)-Codes Wörter der Länge *m* und Polynom mod X^n+1, Erzeugerpolynom $G(x)$ vom Grad $k = n - m$, welches Teiler von x^n+1 ist.
Beipiel Erzeugermatrix:

```
000000000110101
000000001011111
000000010001011
000000100010110
000001000011001
000010000000111
000100000001110
001000000011100
010000000001101
100000000011010
```

Hamming-Codes lineare, also nichtzyklische Codes, besitzen *m* Kontrollstellen und Code-Wörter der Länge $2^m - 1$. Hamming-Abstand 3, 1-bit-Fehler zu korrigieren.
Blockfreie Codes (rekurrente, stetige, Convulotional- oder Baum-Codes) Quelle emittiert stetig, nach *m* Zeichen werden jeweils *n* - *m* Kontroll-Bit in den Datenstrom einfügt.
Code-Spreizung verlagert die einzelnen Bit über einen großen Abstand, macht Bündelfehler zu 1-Bit-Fehler.

Linear, Linear-systematischer Code



Ein $\Gamma(\boldsymbol{n}, \boldsymbol{k}, \boldsymbol{d_{min}})_q$-Code besteht aus:
- Blöcken / Wörtern der Länge $\boldsymbol{n}$, die
- Informations- / Kanal-Bit der Länge $\boldsymbol{k}$ enthalten.
- Daraus leiten sich die Control-Bit $\boldsymbol{c} = n\text{-}k$ ab.
- Die minimale Distanz (Hamming-Abstand) $\boldsymbol{d_{min}} = \boldsymbol{h}$
- Weiter ist die Bittiefe (Stufenzahl) gleich $\boldsymbol{q}$.

Es gibt also q^n mögliche Code-Wörter und
q^k Kanal- / Quellen-Wörter = gültige Code-Wörter.
Das Wortverhältnis

$$w=\frac{q^n}{q^k}=q^{n-k}=q^c>>1$$

ist ein Maß für die Redundanz der Codewörter. Wichtiger ist jedoch hierfür oft die Code-Rate, die angibt, welcher Anteil des Datenstromes „nützlich“ ist:

$$r=\frac{k}{n}<1\cdot$$

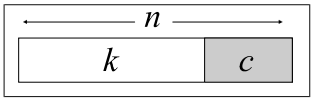
Für den Hamming-Abstand gilt die Abschätzung

$$d \le q^c$$

Das Problem ist das **Finden** günstiger (bestmöglicher) Codewörter

Linear-Systematischer Code

verlangt gegenüber dem Linear-Code nur eine bestimmten Anordnung der Informations- und Controll-Bit im Wort



Dies ist immer erreichbar.

Übergang zum Binärcode q=2

1. erkennbare Fehler = f

Hierfür gilt

$$f \le 2^c\text{-}1 = h\text{-}1$$

oder bzgl. der benötigten Control-Bit

$$c \ge |\mathrm{ld}(f+1)| = |\mathrm{ld}(h)|.$$

Darin bedeutet „| |“ den nächtgrößten ganzzahligen Wert.

2. korrigierbare Fehler = e

Wenn der Hamming-Abstand bekannt ist, dann gilt

$$e_{max} = \mathrm{INT}[(h\text{-}1)/2]$$

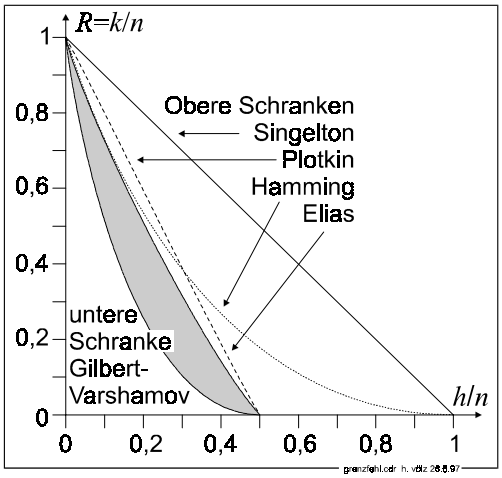
Eine andere Bestimmung führt über die auszuschließenden Wörter mit kleinem Abstand zu den notwendigen Control-Bit

$$c \ge ld\left(\sum_{x=1}^{e}\binom{n}{x}\right)$$

Am besten erfolgt die Berechnung iterativ gemäß:

```
i = 1
for x = e-1 to 0 step -1
   i = i*(n-1)/x -x +1
next x
print ld(i)
print „wähle für c nächstgrößten
                ganzzahligen Wert“
```

Hierzu gibt es Schranken



Galois-Feld $\mathcal{F}_q$

Es enthält q **verschiedene Elemente**; q= 2 ist also binär
Für die Elemente und Folgen daraus sind zwei Operationen definiert

Addition und **Multiplikation.**

Sie entsprechen vereinfacht der üblichen Addition und Multiplikation, jedoch mit dem Zusatz Modula q.

Beispiel $\boldsymbol{F_2}$

*	0	1
0	0	0
1	0	1
+	0	1
0	0	1
1	1	0

Beispiel $\boldsymbol{F_5}$

+	0	1	2	3	4
0	0	1	2	3	4
1	1	2	3	4	0
2	2	3	4	0	1
3	3	4	0	1	2
4	4	0	1	2	3
*	0	1	2	3	4
0	0	0	0	0	0
1	0	1	2	3	4
2	0	2	4	1	3
3	0	3	1	4	2
4	0	4	3	2	1

Matrizen-Multiplikation

Drei Signale können z. B. wie folgt dargestellt werden

$$\mathscr{A} = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 \end{bmatrix}$$

Sie kann mit einem binären Vektor

$$\ell = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$$

multipliziert werden. Dann entsteht ein „Codewort“ gemäß

$$k = \ell^{\mathrm{T}} \cdot \left[\mathscr{A} \right]$$

gemäß

```
  11011
 +00000
 +11000
-------
=  00011
```

Generator-Matrix

Für einen linearen $(n, k)_q$-Code gilt:

„jede Linearkombination gültiger Codewörter ergibt wieder ein gültiges Codewort“.

Daher muß eine Generator-Matrix $n \times k$ existieren, aus der sich alle gültigen Codewörter erzeugen lassen. Sie besteht aus k linear unabhängigen Zeilen mit n-Bit-Wörtern.

Für einen $(7, 4)_2$-Hamming-Code kann sie z.B. lauten:

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

Darin fällt die Einheitsform auf.

$$\boldsymbol{G} = (\boldsymbol{E_k} \mid \boldsymbol{P})$$

Der Anteil $\boldsymbol{P}$ beschreibt darin dann in eindeutiger Weise die Codeeigenschaften.

Sie muß nicht existieren, läßt sich aber stets herstellen. Hier ein Beispiel:

x_i	1 1 0 1 0 0 1 1 0 1 0 0 1 1 1 1 1 0 1 0 0	Generator-Matrix
Vektoren		
0 0 0	0 0 0 0 0 0 0	
0 0 1	1 1 1 0 1 0 0	
0 1 0	1 0 1 0 0 1 1	
0 1 1	0 1 0 0 1 1 1	Codewörter
1 0 0	1 1 0 1 0 0 1	
1 0 1	0 0 1 1 1 0 1	
1 1 0	0 1 1 1 0 1 0	
1 1 1	1 0 0 1 1 1 0	

Prüf-Matrix

Zu jedem $(n, k)_q$-Linear-Code gibt es eine Prüf-Matrix $\boldsymbol{H}$ gemäß

$$\boldsymbol{H} \neq 0 \text{ und } \boldsymbol{G}\boldsymbol{H}^{\mathrm{T}} = 0$$

Auch für Sie existiert eine Einheitsform

$$\boldsymbol{H} = (-\boldsymbol{P}^{\mathrm{T}} \mid \boldsymbol{E}_{n-k})$$

Beispiel:

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix} \qquad H = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

Wird ein beliebiges Wort $\boldsymbol{y}$ der Länge $\boldsymbol{n}$ mit der Prüfmatrix multipiziert, so entsteht das **Syndrom**

$$\boldsymbol{S} = .\boldsymbol{y} \cdot \boldsymbol{H}^{\mathrm{T}}$$

Hierfür gilt

$\boldsymbol{S} = 0$ bei gültigen Codewörtern

$\boldsymbol{S} \neq 0$ sonst.

Damit ist die **Fehlererkennung** im Rahmen der Grenzen des Codes möglich.
Für die Fehlerkorrektur sind jedoch folgende Bedingungen zu beachten:
Die Anzahl der (fehlerhaften) Empfangsworte ist immer größer als die Anzahl möglicher Syndrome

$$\boldsymbol{q}^{\boldsymbol{n}} - \boldsymbol{q}^{\boldsymbol{k}} > \boldsymbol{q}^{\boldsymbol{n-k}}$$

Deshalb sind komplizierte „Berechnungen“ erforderlich. Begriffe die hierzu gehören sind **Nebenklassen** und **Anführer**.

Hamming-Codes

Dies ist eine Klasse von (nichtzyklischen) linearen Kodes der Form

$$\Gamma(\boldsymbol{n}, \boldsymbol{n}\text{-}3, 3)_{\boldsymbol{q}}$$

Für sie gilt folglich:

2 Fehler erkennen; 1 Fehler korrigieren

Die Ordnung der Codes ist $\boldsymbol{r}$ und es gilt dann

$$n = \frac{q^r - 1}{q - 1} = 1 + q + q^2 + q^3 + \cdots + q^{r-1}$$

Die ersten (wichtigsten) binären Hamming-Codes zeigt die Tabelle.
Hier wird die generelle Tendenz deutlich:
mit längeren Blocks kann bei sonst gleichbleibenden Fehler-eigenschaften immer eine wachsende Effektivität erreicht werden.
Hamming-Codes sind perfekt.

r	(n, k)	R %
2	3, 1	33
3	7, 4	57
4	15, 11	73
5	31, 26	84
6	63, 57	90
7	127, 120	94
8	255, 247	97
9	511, 502	98
10	1023, 1013	99
11	2047, 2036	99
12	4095, 4083	99

Zyklische Codes

Beschreibung:

1. Sie sind eine Teilmenge der Linear-Codes (mit Bandmatrizen):
2. Ihre Beschreibung erfolgt (daher) mit Polynomen.
3. Jede **zyklische Verschiebung** eines gültigen Code-Wortes ist wieder ein gültiges Code-Wort.
4. Zu einem Code können aber mehrere Zyklen gehören.

Die Vorteile sind:

1. Die Decodierung und Fehlerbehandlung wird wesentlich einfacher.
2. Sie sind besonders für Bündelfehler (bursts) geeignet.
3. Den Polynomen entsprechen Schaltungen (rückgekoppelte Schieberegister)

Polynom-Definition

Ein Wort (Vektor) der Länge ***n***

$$a = (a_0, a_1, a_2, ..., a_{n-1}) \quad \in \mathcal{F}_q^n$$

wird ein-eindeutig auf ein Polynom abgebildet

$$a(x) = \sum_{i=0}^{n-1} a_i \cdot x^i$$

x ist nur Platzhalter, hängt weder mit Signalen noch mit dem Code zusammen.

Beispiel: $10101 \leftrightarrow 1 \cdot x^4 + 0 \cdot x^3 + 1 \cdot x^2 + 0 \cdot x^1 + 1 \cdot x^0 = x^4 + x^2 + 1$

Ein Polynom steht u. a. für / als

Generatorpolynom, Signalwörter, Schieberegister-Schaltungen

Für die Polynome werden die „üblichen“ Operationen verwendet:

Addition, Multiplikation, Subtraktion und Division

z. B. folgt für: $10101 \cdot 101$:

```
1 |  10101
0 | 00000
1 |10101
--+-------
= |1101001
```

Teilbarkeit

Beispiel: $x^{13} \ / \ x^5+x^4+x^2+1$

```
10000000000000  :  110101 = 111011001
110101          (1) Divisor      Ergebnis
 101010
 110101         (1)
  111110
  110101        (1)
   010110
   000000       (0)
    101100
    110101      (1)
     110010
     110101     (1)
      001110
      000000    (0)
       011100
       000000   (0)
        111000
        110101  (1)
         01101=  Rest der Division
```

Es gibt Polynome ohne Rest, Analogie Primzahlen). Sie heißen **irreduzibel.** Die ersten lauten:

11, 111, 1011, 10011, 11111, 100101, 101111, 110111

links und rechts ist vertauschbar, Symmetrische sind auch irredzibel. Sie erzeugen in entsprechend rückgekoppelten Schieberegistern nur einen Zyklus und werden daher bevorzugt für CRC-Zeichen-Bildung verwendet.

Andere Polynome erzeugen mehr **Zyklen**, z. B. 110011

2 Zyklen der Länge 8

00001, 00010, 00100, 01000, 10000, 10011, 10101, 11001
00111, 01110, 11100, 01011, 10110, 11111, 01101, 11010

3 Zyklen der Länge 4

00011, 00110, 01100, 11000
00101, 01010, 10100, 11011

01001, 10010, 10111, 11101
1 Zyklus der Länge 2
01111, 11110
1 Zyklus der Länge 1
10001

Zyklische Codes 2

(7, 4)-Hamming-Code	
Signal	Code
0000	0000000
0001	0001111
0010	0010110
0011	0011001
0100	0100101
0101	0101010
0110	0110011
0111	0111100
1000	1000011
1001	1001100
1010	1010101
1011	1011010
1100	1100110
1101	1101001
1110	1110000
1111	1111111

(7, 4)-Zyklischer Code		
Signal	Code	
0000	0000000	1. Zyklus Gw. 0
1011	1111111	2. Zyklus Gw. 7
1000	1101000	
0100	0110100	
0010	0011010	
0001	0001101	3. Zyklus Gw. 3
1110	1000110	
0111	0100011	
1101	1010001	
0011	1110010	
0101	0111001	
0110	1011100	4 Zyklus Gw. 4
1001	0101110	
1010	0010111	
1100	1001011	
1111	1100101	

Ein **Generator-Polynom** $g(x)$ für zyklische Codes besitzt immer die **eindeutige Form:**

Γ ist zyklisch $\leftrightarrow$ $g(x)$ ein Teiler von x^n-1

Mit dem Generator-Polynom $g(x)$ wird der **Code** $a(x)$ einfach durch Multiplikation **erzeugt**

$$u(x) \rightarrow a(x) = u(x) \cdot g(x)$$

Das Prüfpolynom $h(x)$ ergibt sich zu

$$g(x) \cdot h(x) = x^n - 1$$

Das Restpolynom

$$R_{x^n-1}[a(x) \cdot h(x)] \rightarrow a(x) \cdot h(x) = u(x) \cdot g(x) \cdot h(x) = u(x) \cdot (x^n - 1)$$

ist für gültige Code-Wörter =0, andernfalls $\neq 0$.

CRC-Codes

cyclic redundancy code, dienen der Fehlererkennung,
ihre „Güte“ wird durch einen Parameter r bestimmt, dann gilt:

$$\Gamma(2^r-1,\ 2^r-r-2,\ 4)_2$$

Wortlänge - Bitlänge(Nutz) - Hamming - $_{\text{Galois}}$

und für das Generator-Polynom

$$g(x) = (1+x) \cdot p(x)$$

mit $p(x)$ primitiv und vom Grad r.

1. Schieberegisterkette wird „Null“ gesetzt
2. Die Information (der Block) „durchläuft“ das Generator-Polynom, zugehöriges Schieberegister.
3. Die Bit, welche nach dem Block im Schieberegister stehen, werden abschließend gesendet.
4. Beim Empfang erfolgt das Gleiche, nur werden die beim Empfang erzeugten Bit mit den gesendeten verglichen.

Fehler werden dann für folgende Muster erkannt

- Fehler bis zum Gewicht 3
- Alle ungeraden Fehler
- Bündelfehler bis zur Länge r+1
- Bündelfehler bis zur Länge r+2 mit der Wahrsacheinlichkeit $1-2^{-r}$
- Bündelfehler der Längen $\geq r$+3 mit der Wahrsacheinlichkeit $1-2^{-(r+1)}$

Wichtige genormte CRC-Polynome sind:

Länge	Fehler	Polynom
8 Bit	$4 \cdot 10^{-3}$	$x^8 + 1$
12 Bit	$2 \cdot 10^{-4}$	$x^{12} + x^{11} + x^3 + x^2 + x + 1$
16 Bit	$2 \cdot 10^{-5}$	$x^{16} + x^{15} + x^2 + 1$; $x^{16} + x^{12} + x^2 + 1$; $x^{16} + x^{12} + x^5 + 1$
32 Bit	$2 \cdot 10^{-10}$	$x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$

Fire-Code

ältester Code für Bündelfehler bei Festplatten
sehr einfach zu beschreiben und zu implementieren
das Generator-Polynom lautet:
$g(x) = (x^{2 \cdot t-1}-1) \cdot p(x)$
Es gelten die folgenden Eigenschaften:

- Das Polynom $p(x)$ ist ein **primitives** vom Grad m
- Es ist t eine Zahl mit $t \leq m$
- Es darf $p(x)$ kein Teiler von $(x^{2 \cdot t-1}-1)$ sein
- Die **Blocklänge** beträgt $n = (2 \cdot t - 1) \cdot q^m - 1$
- Es werden **Bündelfehler** bis zur Länge t korrigiert
- Die Zahl der **Prüfstellen** beträgt mindestens $3 \cdot t - 1$

Decodierung

Es gelten folgende **Prinzipien:**

- Nur **Fehlererkennung**, es wird nur das Syndrom $\neq 0$ angezeigt.
- **Maximum Likelihood-Decodierung:** es wird auf das wahrscheinlichste Wort codiert (halber Hamming-Abstand). Falls mehrere gleichwahrscheinliche Wörter vorliegen, wird die Auswahl vorgegeben oder zufällig gewählt.
  Hierbei kann eine **Fehler-Decodierung** erfolgen.
- **Begrenzte Mindestdistanz-Decodierung.** Es wird eine Decodierkugel gewählt, deren Radius kleiner als der maximal mögliche ist. Es lassen sich Fehlererkennung und -Korrektur verbinden. Dennoch sind fehlerhafte Decodierung möglich.
- In selten Fällen wird auch versucht über den halben Hamming-Abstand hinauszugehen. Es können dann aber viele Fehldecodierungen auftreten.

Bezüglich des **Ergebnisses** der Decodierung werden unterschieden:

- Korrekte Decodierung.
- Falsche Decodierung, z. B. wenn das falsche Kanalwort ein gültiges Wort ist, oder n-Bit-Fehler für einen m-Fehler angesehen wird $m < n$.
- Decodier-Versagen, wenn der Decodierer keine Lösung findet.

Wichtige, nützliche Literatur

bewußt wenig Zitate und nur gut verfügbare Bücher ausgewählt.

Fano, R. M.: Informationsübertragung. Oldenburg 1966, brauchbare Erweiterung zur Shannon-Original-Arbeit.

Friedrichs, B.: Kanalcodierung; Springer 1996, wohl bestes und gut verständliches Buch für digitale Signale, Codierung, Fehlerbehandlung, auch neueste Codes.

Fritzsche, G.: Theoretische Grundlagen der Nachrichtentechnik. 2. Aufl. 1972. brauchbare und verständliche Einfürung in (vor allem analoge) Systeme, Signale und Informationstheorie.

Gerdsen, P. u. Kröger, P.: Digitale Signalverarbeitung in der Nachrichtenübertragung. Springer 1993. Etwas stärker hardwarebezogene (mit formalen Schaltbildern und Programmbeispielen) Betrachtung

Milde, T.: Videokompressionsverfahren im Vergleich, dpunkt 1995. Für JPEG, MPEG Wavelets und Fraktale wichtig.

Ohm, J.-R.: Digitale Bildcodierung. Springer, 1995. Vor allem für Codierung und Komprimierung

Peterson, W. W.: Prüfbare und korrigierbare Codes. Oldenbourg 1967, grundlegendes Werk, beste Einführung ind das Gebiet, neue Codes fehlen allerdings

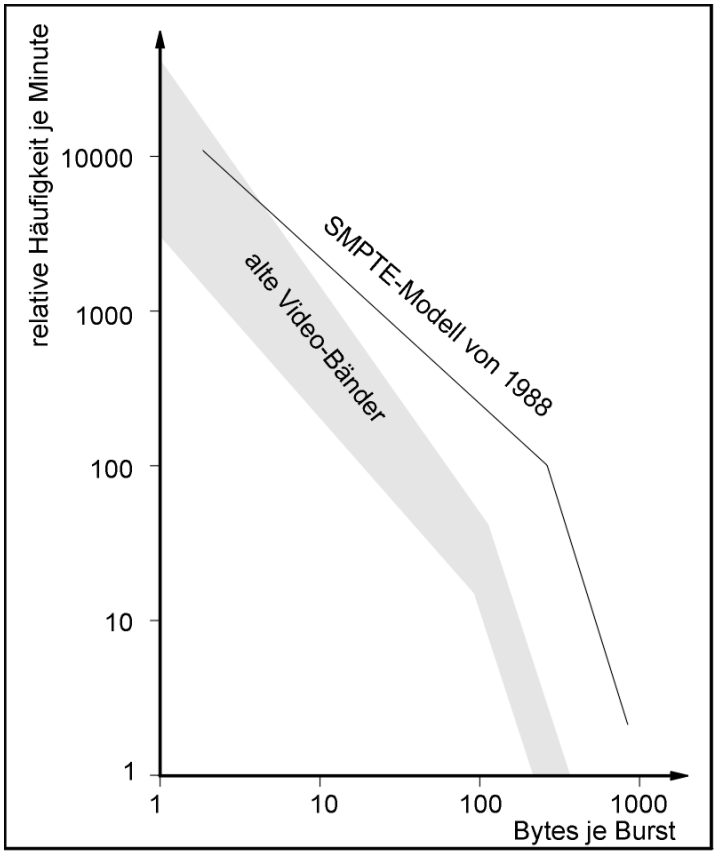
Shannon, Cl. U. Weaver, W.: Mathematische Grundlagen der Informationstheorie. Oldenbourg 1976. Meines Erachtens unverzichtbares Buch. Weaver sollte vergessen werden.

Stadler, E.: Modulationsverfahren, Vogel-Verlag 7. Auflg. 1993, Besonders für die analogen Verfahren gut geeignet.

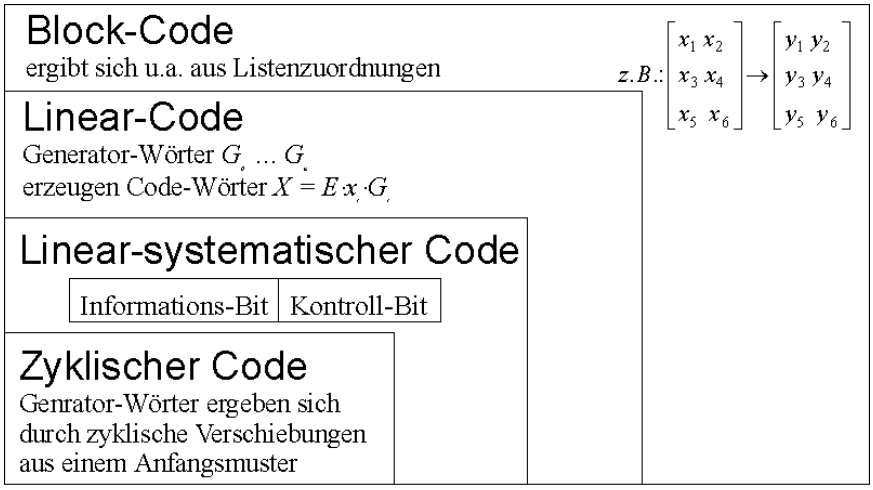
Unbehauen, R.: System-Theorie. Oldenburg 1980. Wichtige Einführung für die Signal-Betrachtungen.

Völz, H.: Informationsspeicher. Expert-Verlag 1996

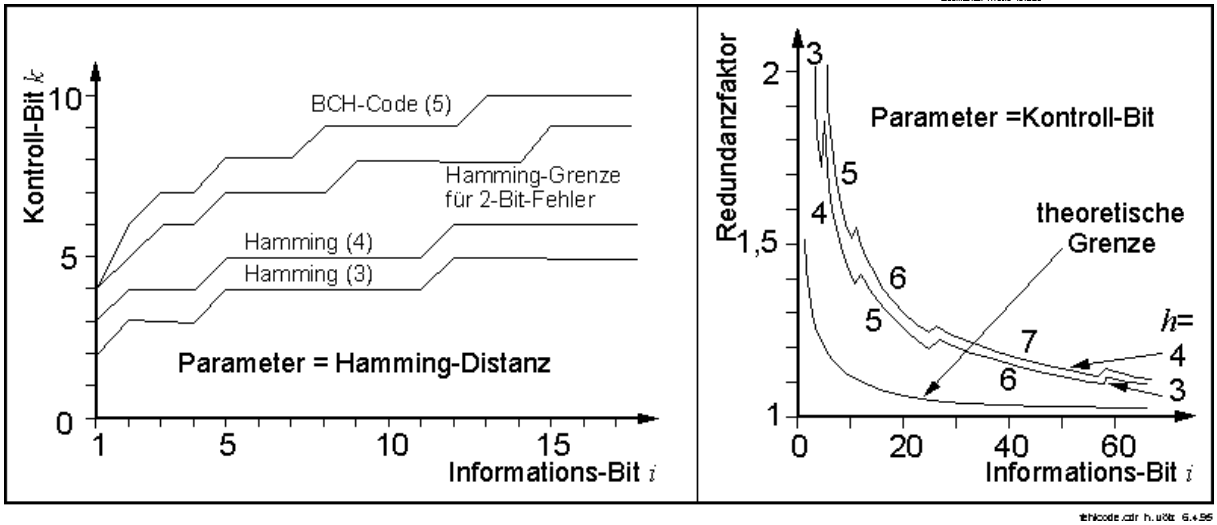
Folien/97/Linearcode.doc; h. völz; 26.6.97



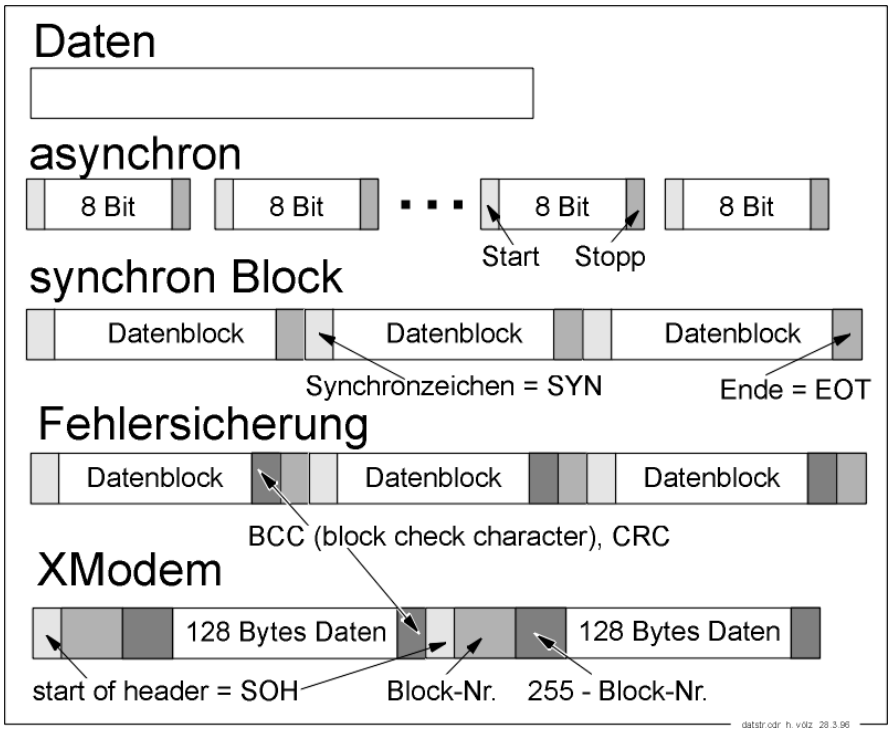
bursts.cdr h. völz 6.4.95

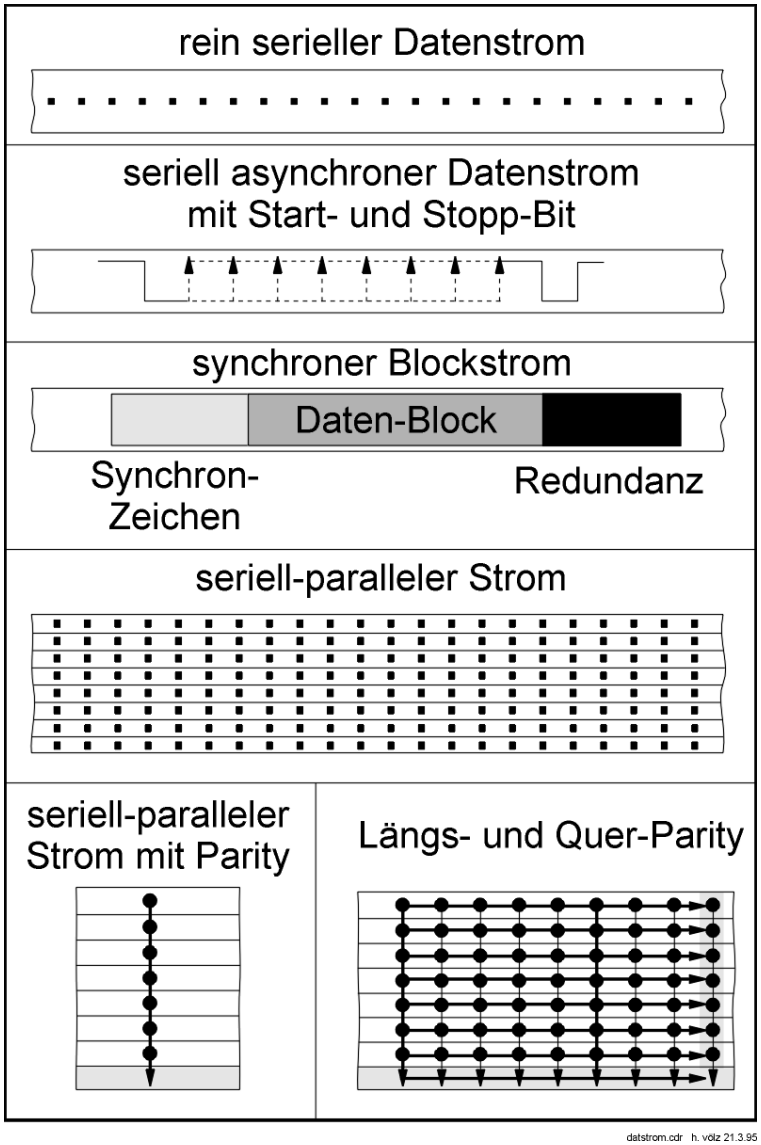


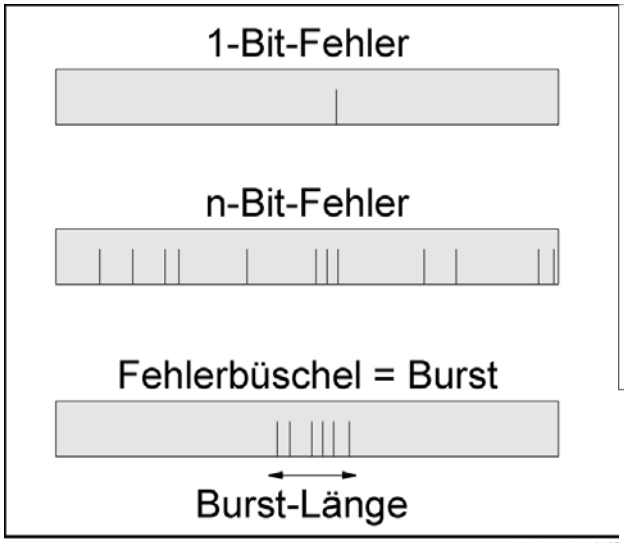
codeklas.cdr h.völz 19.3.96

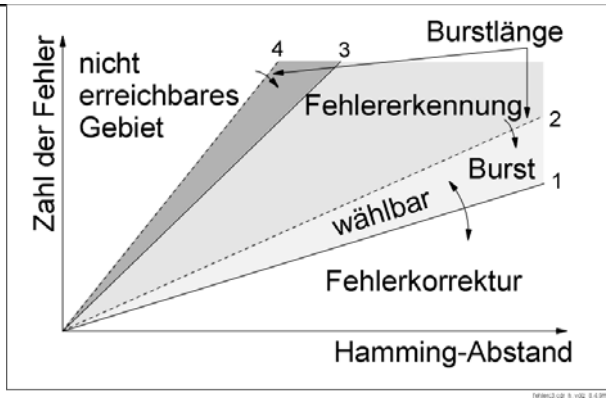


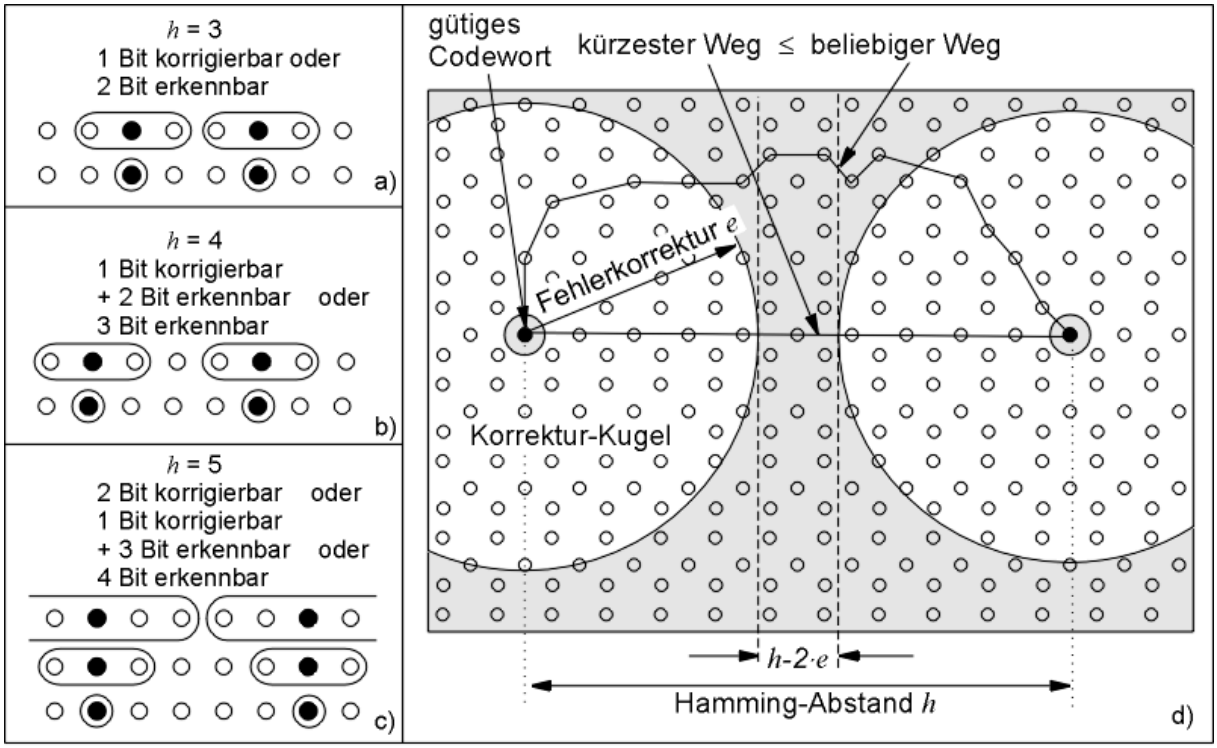
fehlcode.cdr h.völz 6.4.95

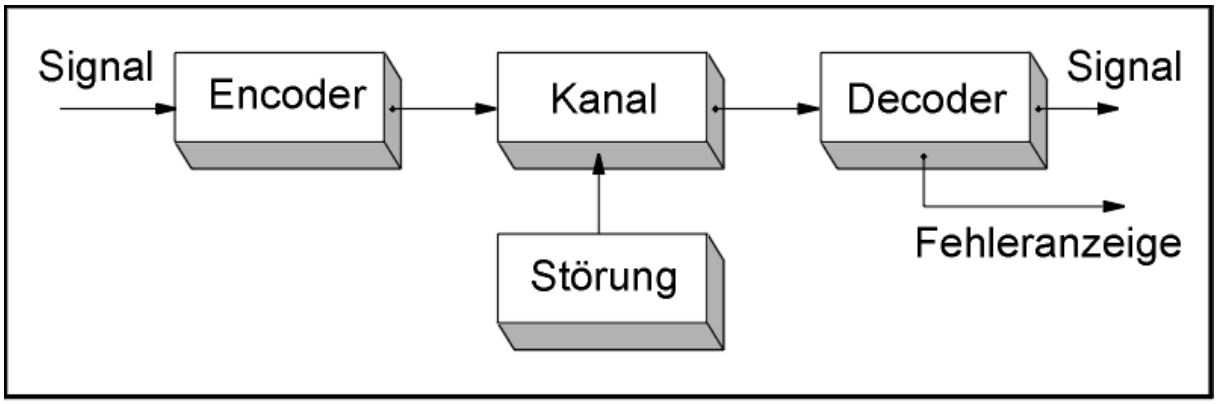

Daten
asynchron
8 Bit
8 Bit
8 Bit
8 Bit
Start
Stopp
synchron Block
Datenblock
Datenblock
Datenblock
Synchronzeichen = SYN
Ende = EOT
Fehlersicherung
Datenblock
Datenblock
Datenblock
BCC (block check character), CRC
XModem
128 Bytes Daten
128 Bytes Daten
start of header = SOH
Block-Nr.
255 - Block-Nr.
datstr.cdr h. völz 28.3.96

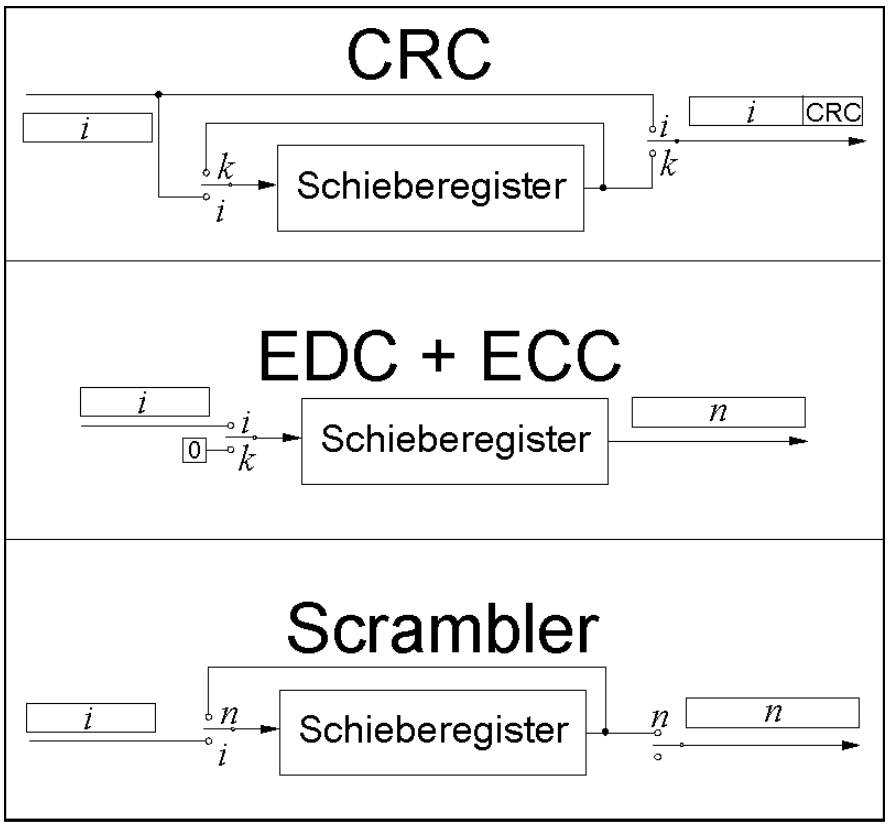

rein serieller Datenstrom
seriell asynchroner Datenstrom
mit Start- und Stopp-Bit
synchroner Blockstrom
Daten-Block
Synchron-
Zeichen
Redundanz
seriell-paralleler Strom
seriell-paralleler
Strom mit Parity
Längs- und Quer-Parity
datstrom.cdr h. völz 21.3.95


1-Bit-Fehler
n-Bit-Fehler
Fehlerbüschel = Burst
Burst-Länge

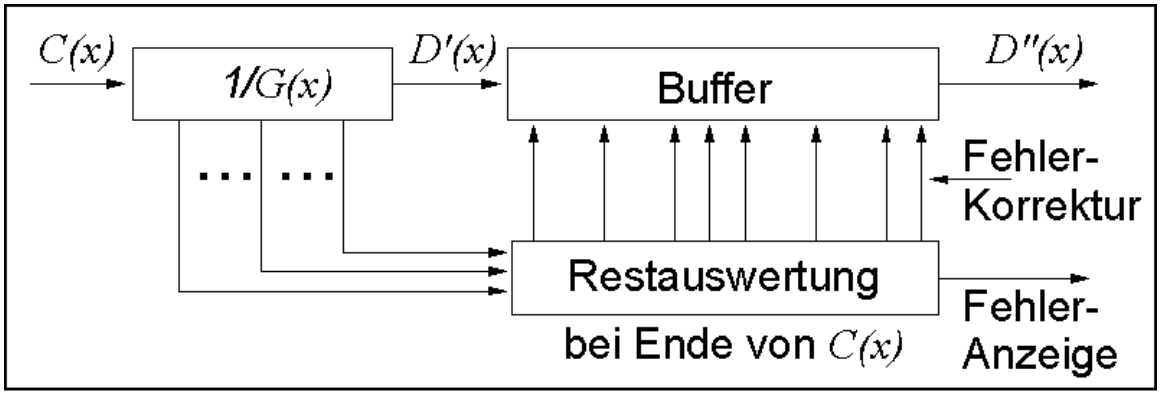

Burstlänge
Zahl der Fehler
nicht erreichbares Gebiet
4
3
2
1
Fehlererkennung
Burst
wählbar
Fehlerkorrektur
Hamming-Abstand


h = 3
1 Bit korrigierbar oder
2 Bit erkennbar
a)
h = 4
1 Bit korrigierbar
+ 2 Bit erkennbar oder
3 Bit erkennbar
b)
h = 5
2 Bit korrigierbar oder
1 Bit korrigierbar
+ 3 Bit erkennbar oder
4 Bit erkennbar
c)
gütiges Codewort
kürzester Weg ≤ beliebiger Weg
Fehlerkorrektur e
Korrektur-Kugel
h-2·e
Hamming-Abstand h
d)

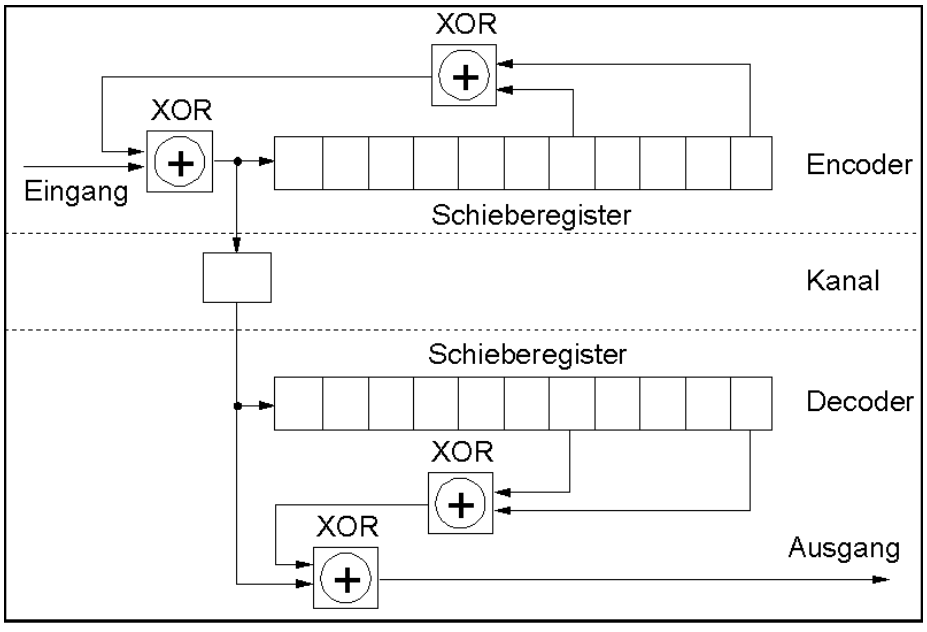

Signal
Encoder
Kanal
Decoder
Signal
Störung
Fehleranzeige



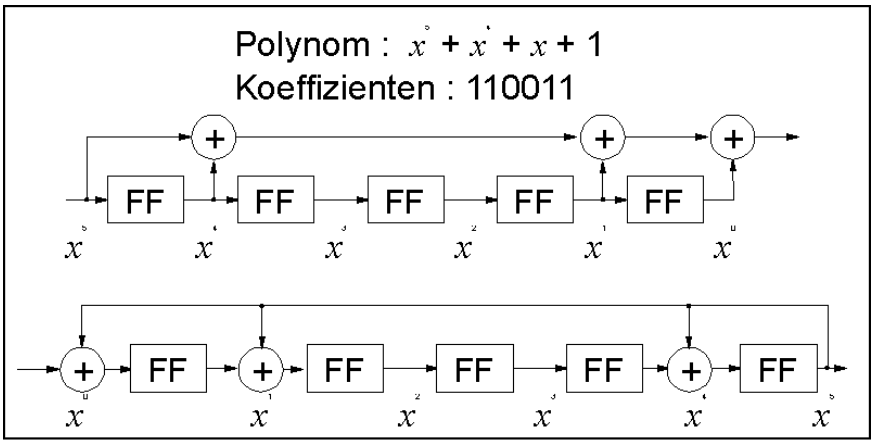
fehlrvar.cdr h. völz 14.4.95

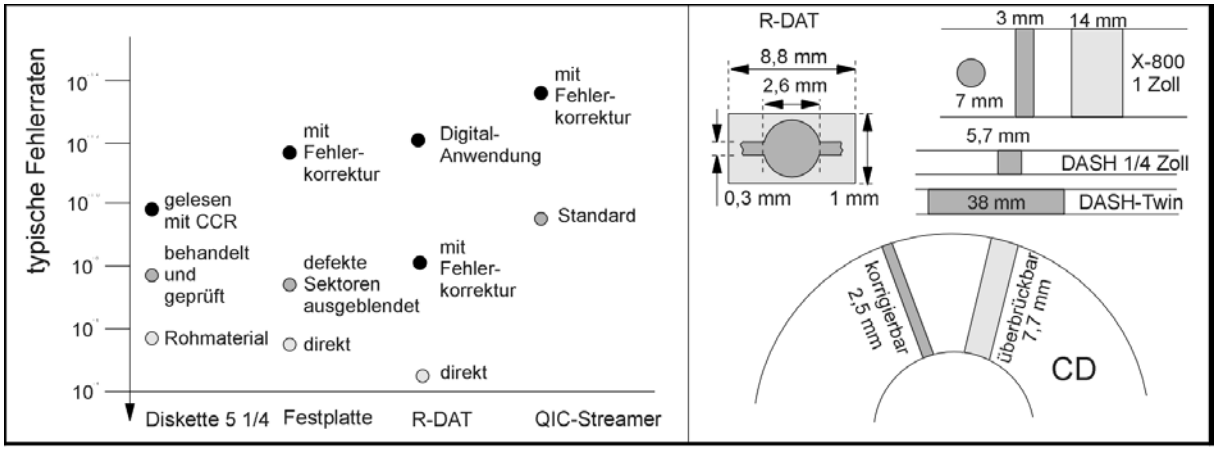


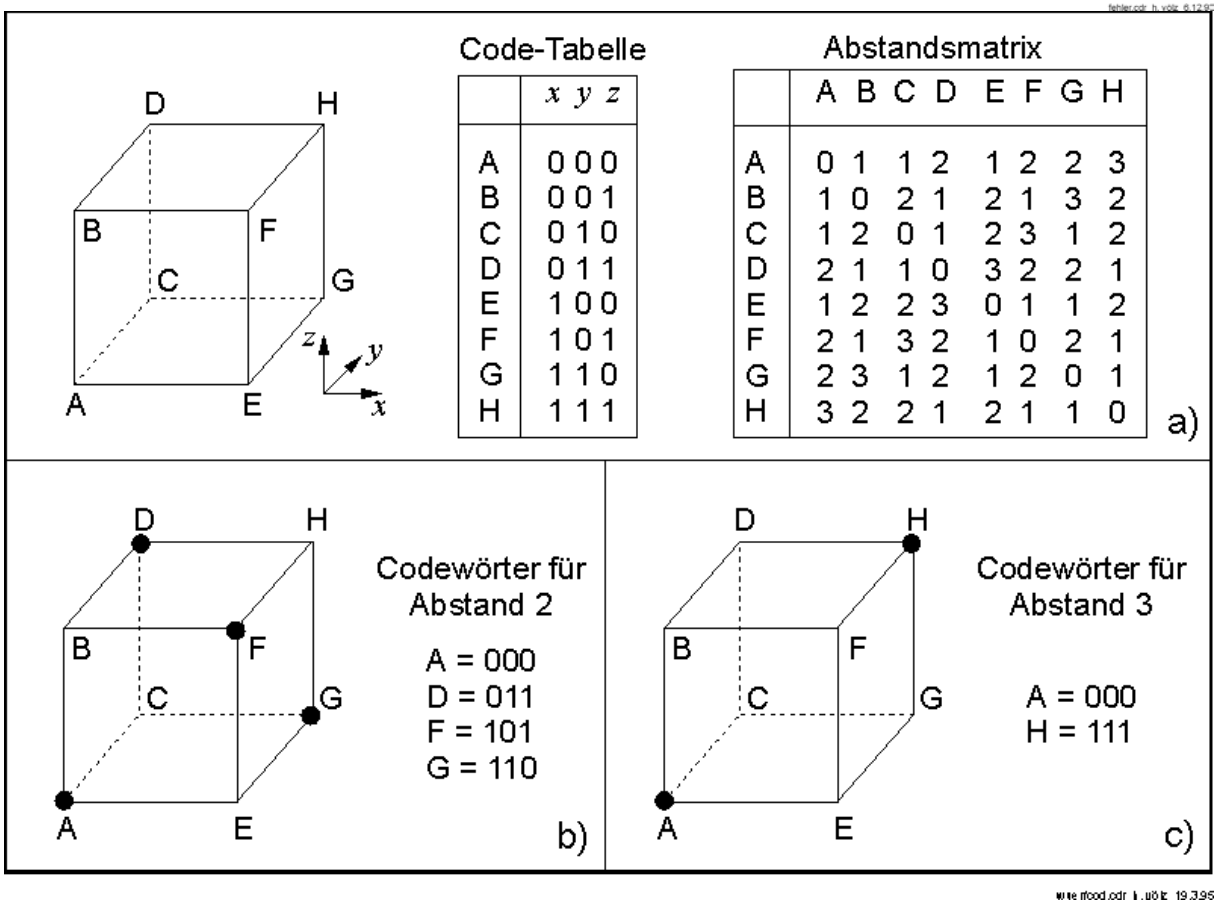
koschal.cdr h.völz 8.4.951



mlz.cdr h. völz 18.3.95

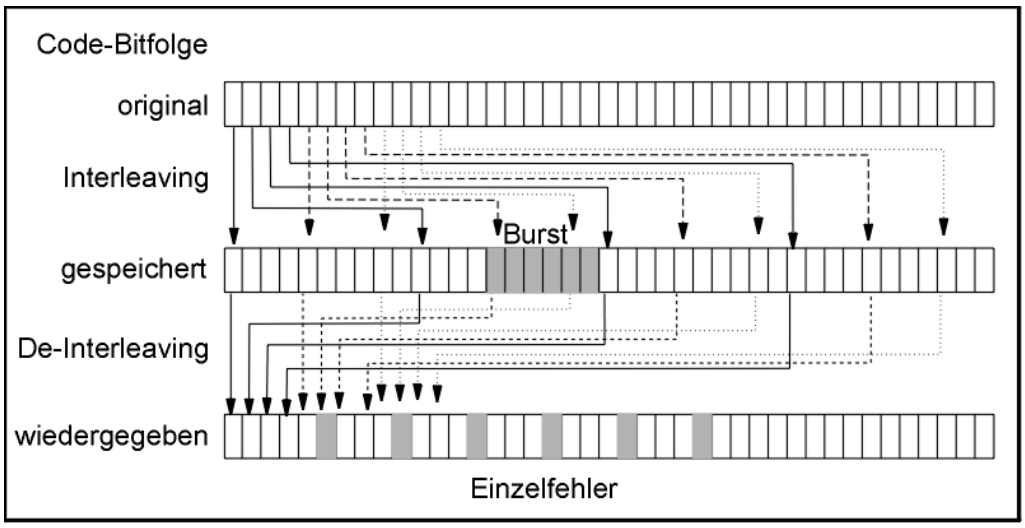


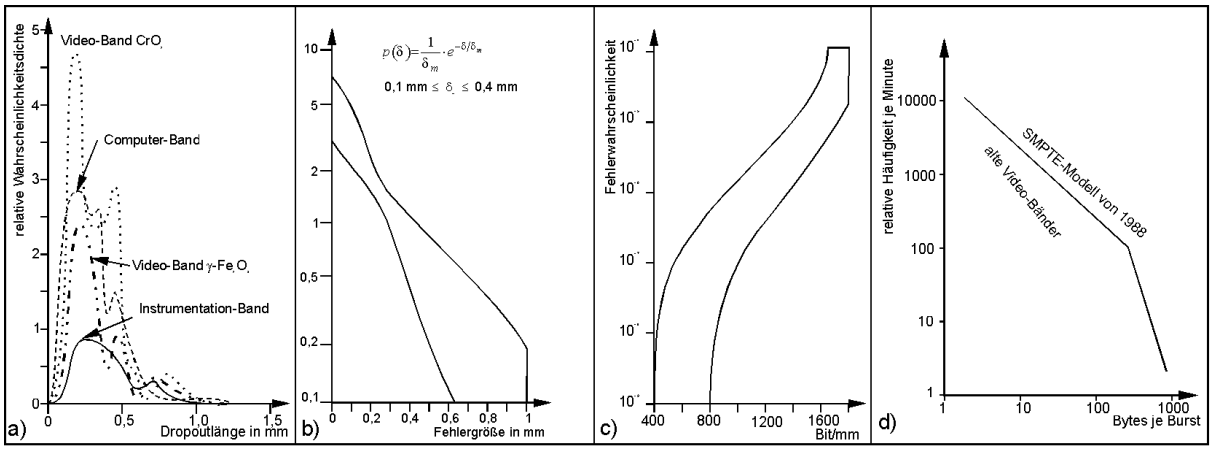


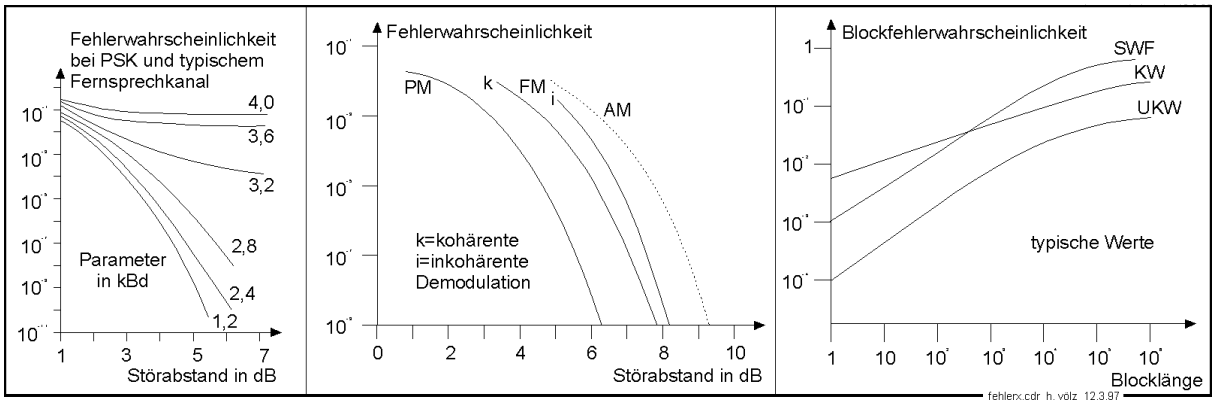


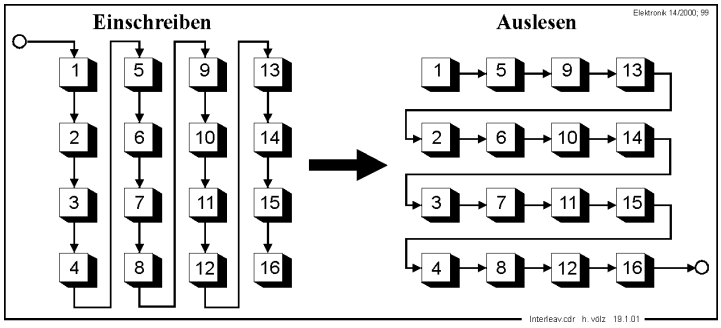
	x y z
A	0 0 0
B	0 0 1
C	0 1 0
D	0 1 1
E	1 0 0
F	1 0 1
G	1 1 0
H	1 1 1

	A	B	C	D	E	F	G	H
A	0	1	1	2	1	2	2	3
B	1	0	2	1	2	1	3	2
C	1	2	0	1	2	3	1	2
D	2	1	1	0	3	2	2	1
E	1	2	2	3	0	1	1	2
F	2	1	3	2	1	0	2	1
G	2	3	1	2	1	2	0	1
H	3	2	2	1	2	1	1	0


Code-Bitfolge
original
Interleaving
Burst
gespeichert
De-Interleaving
wiedergegeben
Einzelfehler
spreizen.cdr h. völz 8.4.95


relative Wahrscheinlichkeitsdichte
Video-Band CrO2
Computer-Band
Video-Band γ-Fe2O3
Instrumentation-Band
Dropoutlänge in mm
a)
0,1 mm ≤ δm ≤ 0,4 mm
Fehlergröße in mm
b)
Fehlerwahrscheinlichkeit
Bit/mm
c)
relative Häufigkeit je Minute
SMPTE-Modell von 1988
alte Video-Bänder
Bytes je Burst
d)


Fehlerwahrscheinlichkeit
bei PSK und typischem
Fernsprechkanal
4,0
3,6
3,2
2,8
2,4
1,2
Parameter
in kBd
Störabstand in dB
Fehlerwahrscheinlichkeit
PM
k
FM
i
AM
k=kohärente
i=inkohärente
Demodulation
Störabstand in dB
Blockfehlerwahrscheinlichkeit
SWF
KW
UKW
typische Werte
Blocklänge
fehlerx.cdr h. völz 12.3.97


Einschreiben
Auslesen
Elektronik 14/2000; 99
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
Interleav.cdr h. völz 19.1.01