EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

FACULTY OF
SCIENCE
Communication Networks

Informatik der Systeme – Chapter 3: Coding

Prof. Dr. Michael Menth

http://kn.inf.uni-tuebingen.de

Kodierungsarten in Übertragungssystemen

- Quellkodierung (source coding)
  - Daten enthalten oft redundante Informationen
  - Komprimierung
    - Ziel: Geringerer Speicherbedarf bzw. schnellere Übertragung von Informationen
    - Verlustbehafet oder verlustfrei
  - Beispiele
    - Komprimierte Datenformate: mpg, jpg, …
    - Tools um Daten zu komprimieren: zip, rar, …
    - Algorithmen: z.B. Run-Length Encoding, Huffman-Coding
- Verschlüsselung (mehr z.B. in Vorlesung Kommunikationsnetze)
- Kanalkodierung (channel coding)
  - Sichert Daten für Speicherung oder Übertragung gegen Bitfehler
  - Hinzufügen von redundanter Information
  - Bitfehler können erkannt und evtl. sogar korrigiert werden
- Leitungskodierung (line coding)
  - Aufbereitung von Bits für eine Übertragungsleitung
- Verfahren auch in anderen Systemen wie z.B. Speichern verwendbar

Run-Length Encoding (Lauflängenkodierung)

- Einfaches, verlustfreies (Quell-)Kodierungsverfahren
  - Entfernt redundante Information durch Ersetzen mittels Rekonstruktionsvorschrift
- Ersetzt Strom von aufeinanderfolgenden, wiederholten Sequenzen durch
  - Eine Marker-Sequenz, die angibt, dass die nächste Sequenz komprimiert ist
  - Die Anzahl von Wiederholungen minus einem Offset, der die minimale Anzahl von Wiederholungen für die Komprimierung angibt
  - Die wiederholte Sequenz
- Beispiel
  - Bytes als zweistellige Hexadezimalzahlen
  - 00000000000000000FF00000000000000000FFFFFF
  - AA ist Marker, Offset ist 4
  - Komprimiert: AA600FFAA600FFFFFF
  - Falls Marker X in Daten vorkommt, ersetze X in Daten mit XX
- Quelle: Wikipedia

Informationsgehalt und mittlere Codewortlänge

- Binäre Codes
  - Können unterschiedlich lange Codewörter enthalten
  - Kein Codewort darf Präfix eines anderen sein (präfixfrei)
- Quelle liefert N unterschiedliche Symbole x_i mit Auftrittswahrscheinlichkeit $p(x_i)$
  - Informationsgehalt eines Zeichens $I(x_i) = ld\left(\frac{1}{p(x_i)}\right)\ bit$
  - Informationsgehalt einer Quelle $H^* = \sum_{1 \le i \le N} p(x_i) \cdot ld\left(\frac{1}{p(x_i)}\right)\ bit$
- Sei Symbol x_i durch ein binäres Codewort der Länge S_i kodiert
  - Mittlere Codewortlänge $S_m = \sum_{1 \le i \le N} S_i \cdot p(x_i)\ bit$
- Redundanz der Quellkodierung $R_c = S_m - H^*$
- Satz von Shannon
  - Es existiert ein Code mit $ld\left(\frac{1}{p(x_i)}\right) \le S_i \le ld\left(\frac{1}{p(x_i)}\right) + 1$
  - $\Rightarrow$ Es existiert ein Code mit $H^* \le S_m \le H^* + 1$

Beispiel

Symbol	Auftrittswahr-scheinlichlichkeit	Code 1	Code 2
x_1	1/2	00	0
x_2	1/4	01	10
x_3	1/8	10	110
x_4	1/8	11	111
	$H^* = 1{,}75\ bit$	$S_m = 2\ bit$	$S_m = 1{,}75\ bit$

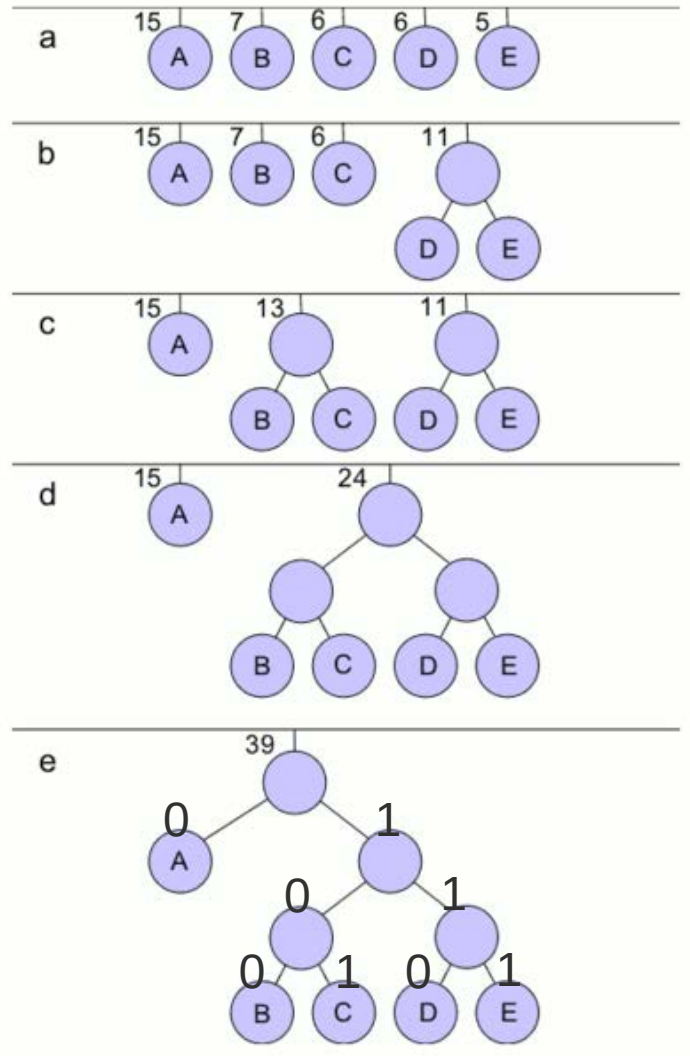
- In diesem Beispiel optimale Quellkodierung ohne Redundanz wegen der günstigen Auftrittswahrscheinlichkeitsverteilung möglich

Huffman-Kodierung

- David A. Huffmann, 1952
- Huffman-Kodierung
  - Liefert beweisbar optimalen Baum für Symbole mit gegebenen Häufigkeiten (oder alternativ Auftrittswahrscheinlichkeiten)
- Algorithmus
  - Erstelle einen Wald mit einem Baum für jedes Zeichen! Jeder dieser Bäume enthält nur einen einzigen Knoten: das Zeichen. Schreibe die Häufigkeit des Zeichens an den Baum!
  - Suche die beiden Bäume, die die geringste Häufigkeit haben! Fasse beide Bäume zusammen, indem sie die Teilbäume einer neuen Wurzel werden! Berechne die Summe der Häufigkeiten der beiden Teilbäume als die Häufigkeit des neuen Baumes!
  - Wiederhole Schritt 2 so oft, bis nur noch ein Baum übrig ist!
  - Nutze diesen Baum als Code-Baum zur Kodierung der Zeichen!
- Quelle: Wikipedia

Beispiel

- Gegeben
  - Symbole A, B, C, D, E mit Häufigkeiten
  - Auftrittswahrscheinlichkeiten: Häufigkeiten geteilt durch 39
- Code
  - A=0
  - B=100
  - C=101
  - D=110
  - E=111
- $H^* = 2{,}19\ bit$
- $S_m = 2{,}28\ bit$

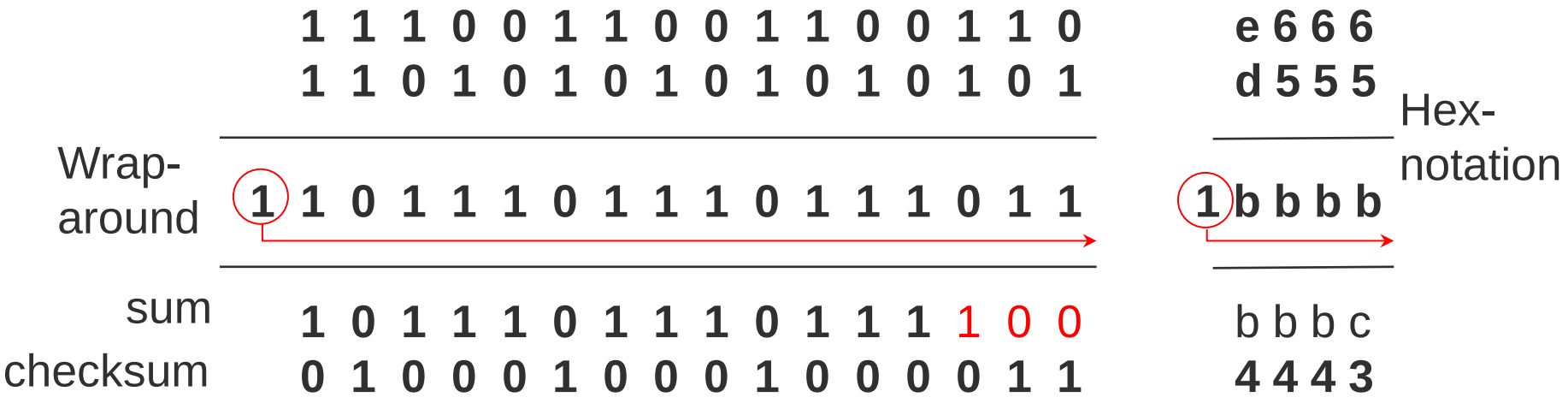


Paritäts-Code für Kanalkodierung

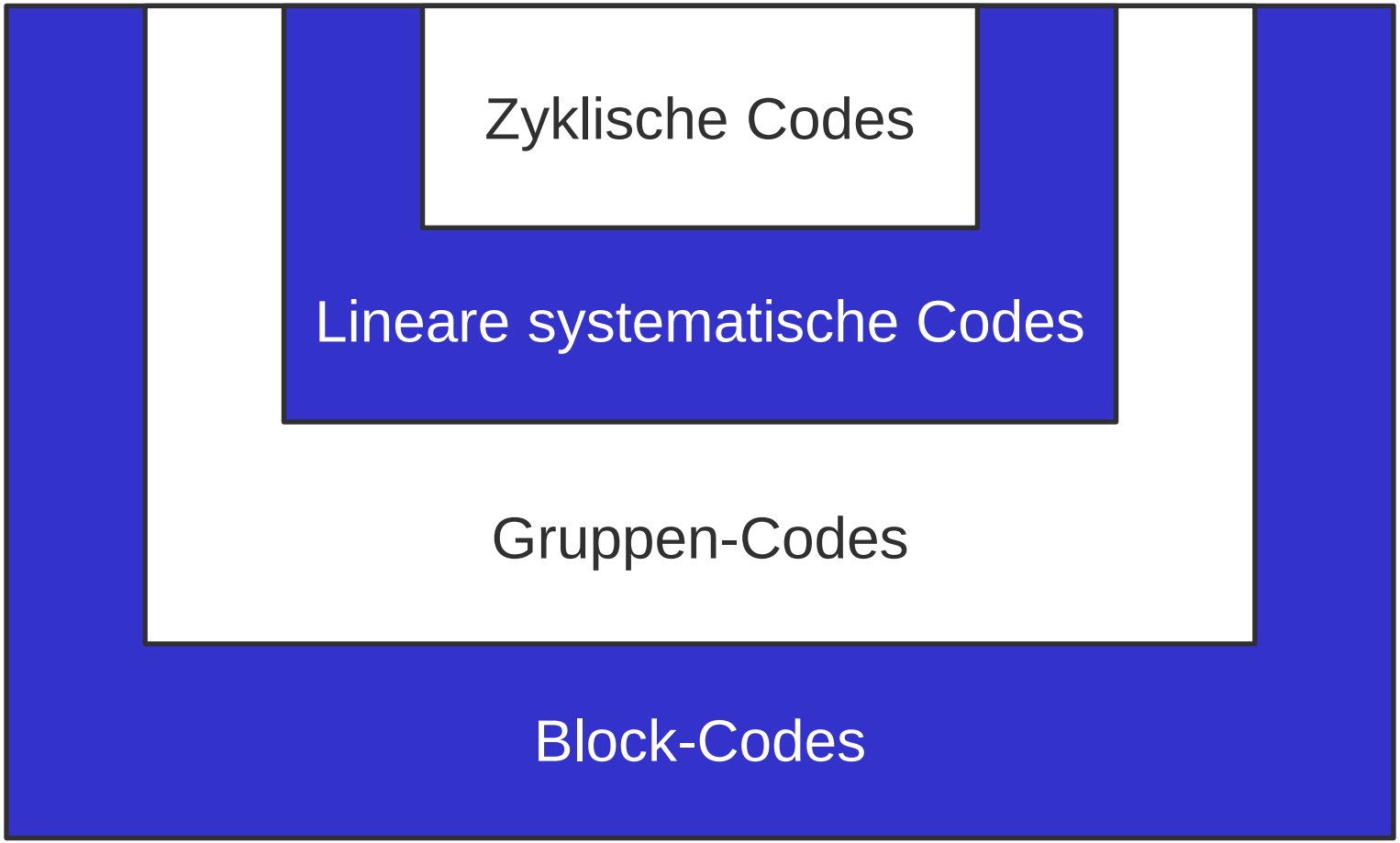
- Verwendung der Parität als Redundanz zur Fehlererkennung
- Parität: Addition der Bits eines Wortes mittels Exclusive-OR $\oplus$
  - $0 \oplus 0 = 0$, $1 \oplus 0 = 1$, $0 \oplus 1 = 1$, $1 \oplus 1 = 0$
- Beispiel
  - Gegeben: Nutzwort der Länge 2
  - Korrekte Code-Wörter: 00|0, 10|1, 01|1, 11|0
- Kann für Code-Wörter mit fester Länge auch als Gruppen-Code und sogar als linearer systematischer Code aufgefasst werden (siehe später)

Internet Checksum

- RFC1071, RFC1141, RFC1624
- Used by IP, UDP, TCP, …
- Uses 1's complementary addition
  - Like adding 16-bit binary integers
  - But: a carryout from the most significant bit is added to the result
- Example: checksum of 4 bytes
- The checksum is the inverted sum
- Checksum field contains zeros for checksum calculation
- Validation: correct if checksum calculated including checksum field yields only ones (= –0)



Weitere binäre Codes für Kanalkodierung



Binäre Block-Codes

- Abbildung von Nutzwörtern (NWs) auf Codewörter (CWs)
- Alle CWs eines Block-Codes müssen gleich lang sein
- Kodierungsvorschrift am Beispiel
  - $0 \rightarrow 001$ bzw. $00 \rightarrow 001$
  - $1 \rightarrow 010$ bzw. $01 \rightarrow 010$
  - $2 \rightarrow 100$ bzw. $10 \rightarrow 100$
  - $3 \rightarrow 101$ bzw. $11 \rightarrow 101$

Rechnen mit den Elementen 0 und 1

- Definition (rechte Seite)
  - Addition
    - Neutrales Element: 0
  - Multiplikation
    - Neutrales Element: 1

+=⊕ (XOR)	0	1
0	0	1
1	1	0

- Galois-Feld GF(2)
  - Aus der Algebra
  - Endlicher Körper
  - Charakteristik 2: Elemente 0, 1
  - Addition und Multiplikation
  - Es gelten Distributiv- und Kommutativgesetze

×=∧ (AND)	0	1
0	0	0
1	0	1

- http://de.wikipedia.org/wiki/Endlicher_Körper

Hamming-Distanz

- Summe $(C_i \oplus C_j)$ zweier CWs C_i und C_j
  - Addition der Stellen mittels Exclusive-OR
  - Beispiel: 101 $\oplus$ 100=001
- Gewicht $w(C)$ eines CWs C
  - Anzahl der Einsen in C
- Distanz $d(C_i, C_j)$ zweier CWs C_i und C_j
  - $d(C_i, C_j) = w(C_i \oplus C_j)$
- Hamming-Distanz eines Codes
  - $h = \min_{i \neq j} \left(d(C_i, C_j) \right)$
  - Mindestens h Stellen eines CWs müssten bei der Übertragung verfälscht werden, damit zufälligerweise ein anderes gültiges CW dabei entsteht
  - Mit einem Code mit Hamming-Distanz h kann man bis zu $\bar{\eta} = h - 1$ Bitfehler erkennen und $\eta = \left\lfloor \frac{\bar{\eta}}{2} \right\rfloor$ Bitfehler korrigieren

Triplet-Code

- Triplet-Code
  - Einfacher Block-Code für NWs 0 und 1
  - Verdreifachung eines jeden Bits
  - Ineffizient weil CWs dreimal so lang wie NWs
- Hamming-Distanz des Codes $h = 3$
- $\bar{\eta} = h - 1 = 2$ Bitfehler können korrekt erkannt werden
  - Fehlerfreie Übertragung: 000, 111
- $\eta = \left\lfloor \frac{\bar{\eta}}{2} \right\rfloor = 1$ Bitfehler kann korrekt korrigiert werden
  - Korrektur durch Majoritätsentscheidung
  - Generelles Prinzip: Korrektur zum ähnlichsten gültigen CW (kleinste Distanz)

Binäre Gruppen-Codes

- Unabhängigkeit einer Menge von CWs
  - Kein CW dieser Menge kann als Summe anderer CWs dieser Menge dargestellt werden

- Binärer Gruppen-Code
  - Gegeben
    - m unabhängige CWs $G_1, G_2, \dots, G_m$ mit $G_i = (g_{i1}, g_{i2}, \dots, g_{in})$, die eine Generatormatrix $G = \begin{pmatrix} G_1 \\ \vdots \\ G_m \end{pmatrix}$ bilden
    - NW: $X = (x_1, x_2, \dots, x_m)$
  - Kodierungsvorschrift
    - $C(X, G) = X \cdot G$
    - Addition bei Multiplikation im Sinne von $\oplus$

Beispiel Gruppen-Code

- Generatormatrix
  - $G = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix}$
- NWs und dazugehörige CWs
  - $00 \rightarrow 000$
  - $01 \rightarrow 011$
  - $10 \rightarrow 101$
  - $11 \rightarrow 110$
- Dieses spezielle Beispiel entspricht einem Paritäts-Code

Eigenschaften von Gruppen-Codes

- Definitionen
  - Gewichtsverteilung eines Codes $\mathcal{C}$
    - $A(k)$ ist Anzahl von CWs mit Gewicht k
  - Distanzverteilung eines CWs C
    - $B(C,k)$ ist Anzahl anderer CWs mit Distanz k
- Die CWs eines Gruppen-Codes $\mathcal{C}$ bilden eine Gruppe bezüglich $\oplus$
  - $(C_i \oplus C_j) \in \mathcal{C}$ ist wieder ein gültiges CW
  - Assoziativität $\big((C_i \oplus C_j) \oplus C_k = C_i \oplus (C_j \oplus C_k)\big)$, neutrales Element, inverses Element
- Eigenschaften von Gruppen-Codes
  - Die Distanzverteilung $B(C,k)$ ist gleich für alle CWs $C \in \mathcal{C}$
  - Wegen $B\big((0,\ldots,0),k\big) = A(k)$ gleicht sie der Gewichtsverteilung
  - Das Gewicht des CWs (ungleich 0…0) mit den wenigsten Einsen entspricht der Hamming-Distanz des Codes

Lineare systematische Binär-Codes

- Gegeben
  - Prüfschema P, z.B. $P = (P_1 \quad \dots \quad P_n) = \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 \end{pmatrix}$
  - k (Prüf-)Zeilen, $n = m + k$ Prüfspalten P_i
  - Rechts steht eine $k \times k$ Einheitsmatrix I_k
- Transposition einer Matrix (oder eines Vektors)
  - $M = \begin{pmatrix} m_{11} & \cdots & m_{1n} \\ \vdots & \ddots & \vdots \\ m_{k1} & \cdots & m_{kn} \end{pmatrix}$, $M^T = \begin{pmatrix} m_{11} & \cdots & m_{k1} \\ \vdots & \ddots & \vdots \\ m_{1n} & \cdots & m_{kn} \end{pmatrix}$
- Kodierungsvorschrift
  - Berechne für NW $X = (x_1, \dots, x_m)$ Kontrollstellen $Y = (y_1, \dots, y_k)$
    - $Y = X \cdot (P_1 \quad \dots \quad P_m)^T$ bzw. $Y = \sum_{1 \le i \le m} x_i \cdot (P_i)^T$
  - Resultierendes CW $C = XY$
  - Vorteil: NW in den ersten m Stellen des CW enthalten

Beispiel: Lineare systematische Binär-Codes

- Gegeben
  - Prüfschema $P = \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 \end{pmatrix}$
  - NW $X = (1011)$
- Berechnung der Kontrollstellen
  - $Y = X \cdot (P_1 \quad \ldots \quad P_m)^T = (1\,0\,1\,1) \cdot \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix} = (0\,0\,1)$
  - Resultierendes CW $C = XY = (1\,0\,1\,1\,0\,0\,1)$
- Basis eines äquivalenten Gruppencodes
  - Erhält man z.B. aus den CWs für die Einheitsvektoren von NWs
  - Beispiel: $(1\,0\,0\,0\,1\,1\,1), (0\,1\,0\,0\,1\,1\,0), (0\,0\,1\,0\,1\,0\,1), (0\,0\,0\,1\,0\,1\,1)$

Prüfen eines CWs mittels Prüfmatrix

- Gegeben: empfangenes Wort C^*
- Berechnung des Fehlersyndroms $Z = P \cdot (C^*)^T$
- Falls C^* unverfälscht empfangen wurde, dann gilt
  - $C^* = \mathrm{C} = \mathrm{XY}$ mit
  - $Z = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}$
- Beispiel
  - $Z = P \cdot (C^*)^T == \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$

Prüfen eines CWs mittels Prüfmatrix

- Gegeben: empfangenes Wort C^*
- Berechnung des Fehlersyndroms $Z = P \cdot (C^*)^T$
- Falls C^* unverfälscht empfangen wurde, dann gilt
  - $C^* = \mathrm{C} = \mathrm{XY}$ mit
  - $Z = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}$
- Beweis
  - $Z = P \cdot (C^*)^T$

$$= \left((P_1 \quad \dots \quad P_m) | I_k\right) \cdot \begin{pmatrix} X^T \\ (P_1 \quad \dots \quad P_m) \cdot X^T \end{pmatrix}$$

$$= (P_1 \quad \dots \quad P_m) \cdot X^T \oplus (P_1 \quad \dots \quad P_m) \cdot X^T$$

$$= \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}$$

Lokalisierung von Einzelbitfehlern

- Annahme
  - Empfangenes Wort C^* hat genau 1 Bitfehler
  - Bitfehlervektor $F = (f_1 \quad ... \quad f_n)$ (alles Nullen bis auf eine Eins)

- Lokalisierung des Bitfehlers anhand des empfangenen Wortes C^*
  - Originalwort C ist korrektes CW $\Rightarrow P \cdot C^T = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}$
  - Mit $C^* = (C \oplus F)$ ergibt $Z = P \cdot (C \oplus F)^T = P \cdot C^T \oplus P \cdot F^T = P \cdot F^T$
  - Da F nur eine Eins enthält, ist das Fehlersyndrom mit der Prüfspalte der Stelle identisch, an der der Fehler aufgetreten ist.

- Folge
  - Einzelfehler korrigierbar
  - Falsche Korrektur bei Mehrfachfehlern

Beispiel: Lokalisierung von Einzelfehlern

- Gegeben
  - Prüfschema $P = \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 \end{pmatrix}$
  - NW $X = (1011)$
- Berechnung der Kontrollstellen
  - $Y = X \cdot (P_1 \quad \dots \quad P_m)^T = (1\,0\,1\,1) \cdot \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix} = (0\,0\,1)$
  - Resultierendes CW $C = XY = (1\,0\,1\,1\,0\,0\,1)$, $\mathrm{Z} = P \cdot C^T = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}$
  - Fehlerhaftes CW $C^* = (1\,1\,1\,1\,0\,0\,1)$, $\mathrm{Z} = P \cdot C^{*T} = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}$

Eigenschaften linearer systematischer Codes

- Ein linearer systematischer Code, bei dem jeweils $\bar{\eta}$ Prüfspalten der Prüfmatrix linear unabhängig sind, hat eine Hamming-Distanz von $h = \bar{\eta} + 1$
  - Mit diesem Code können bis zu $\bar{\eta}$ simultane Bitfehler erkannt (nicht korrigiert) werden.
  - Es ergeben sich dann Fehlersyndrome ungleich $\begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}$
- Spezialfall $h = 3$ (Aufstellung des Prüfschemas einfach)
  - Alle Prüfspalten ungleich Null
  - Alle Prüfspalten paarweise unterschiedlich (lineare Unabhängigkeit)
  - Mit $n = m + k$ Prüfspalten à k Elementen gilt $2^k - 1 \geq n = m + k$
    - Die Länge des Codes kann auch kürzer gewählt werden, dann hat aber auch die Prüfmatrix weniger Spalten
- $h \geq 3$
  - Aufstellung des Prüfschemas schwierig
  - Zyklische Binär-Codes helfen dabei
  - Auch Mehrfachfehler unter bestimmten Voraussetzungen korrigierbar

Division von Polynomen über {0,1}

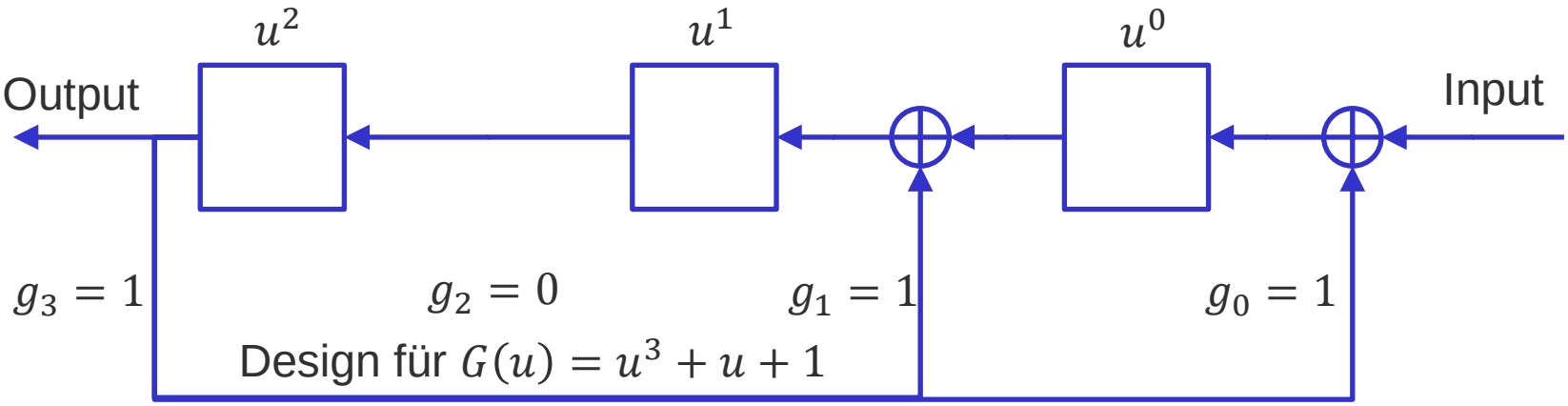
- Das Polynom $\sum_{0 \le i \le k} a_i \cdot u^i$ mit Koeffizienten aus $a_i \epsilon \{0,1\}$ hat den Grad k
  - Beispiel: $u^3 + u + 1$ hat Grad 3

- Polynomdivsion am Beispiel (u^6 und $u^3 + u + 1$ willkürlich gewählt)
  - $u^6 : (u^3 + u + 1\ ) = u^3 + u + 1$ Rest $u^2 + 1$
  - `1000000:1011=1011 Rest 101`

```
1011
 01100
  1011
   1110
   1011
    101
```

- Die Modulo-Operation berechnet nur den Rest
  - Beispiel: $u^6 \bmod (u^3 + u + 1) = u^2 + 1$

Rückgekoppeltes Schieberegister



- Getaktete Schaltung
  - Input und Registerausgänge dienen als Input für Logikschaltung
  - Registereingänge werden bei Taktwechsel übernommen
- Durchgeführte Operation
  - $A(u)$: $G(u) = B(u)$ Rest $C(u)$
  - Durch Schaltung implementiert: $G(u)$
  - Input: Koeffizienten von $A(u)$ in Reihenfolge $a_{k_A}, a_{k_A-1}, \dots, a_0$
  - Output: Koeffizienten von $B(u)$ in Reihenfolge $b_{k_B}, b_{k_B-1}, \dots, b_0$
  - Registerinhalte nach $k_A + 1$ Takten: Koeffizienten von $C(u)$

Restklassen und Primitivität

- Die Menge der Reste $u^k \bmod G(u)$, $k = 0, 1, 2, \ldots$ wird als Menge der Restklassen von $G(u)$ bezeichnet
- Bei einem Polynom von grad k kann es maximal $2^k - 1$ unterschiedliche nicht-triviale ($\neq 0$) Reste geben
- Die Reste wiederholen sich irgendwann, dadurch entsteht eine Periode p
- Falls die Menge der Restklassen tatsächlich $2^k - 1$ Elemente enthält, ist die Periode maximal und man nennt das Polynom $G(u)$ primitiv
- Primitive Polynome sind irreduzibel, d.h. sie lassen sich durch kein anderes Polynom von Grad k>0 teilen

- Beispiel $\mathrm{G}(u) = u^3 + u + 1$
  - $u^0 \bmod G(u) = 001$
  - $u^1 \bmod G(u) = 010$
  - $u^2 \bmod G(u) = 100$
  - $u^3 \bmod G(u) = 011$
  - $u^4 \bmod G(u) = 110$
  - $u^5 \bmod G(u) = 111$
  - $u^6 \bmod G(u) = 101$
  - $u^7 \bmod G(u) = 001$
  - $u^8 \bmod G(u) = 010$
  - …
  - Die Periode ist $p = 7 = 2^k - 1$
    $\Rightarrow \mathrm{G}(u) = u^3 + u + 1$ ist primitiv

Schnelle Primitivitätsprüfung

- Polynom $G(u)$ ist primitiv $\Leftrightarrow$
  $(u^q \oplus 1) \bmod G(u) \neq 0$ für $0 < q < 2^p - 1$, wobei p Grad von $G(u)$

- Beispiel: $G(u) = u^3 + u + 1$

```
10000001:1011=10111 Rest 0
1011
  1100
  1011
   1110
   1011
    1010
    1011
       1
```

Es werden so viele Nullen wie nötig sowie eine abschließende Eins aufgefüllt, bis ein Polynom der Form $u^q \oplus 1$ entsteht, das sich restlos durch $G(u)$ teilen lässt.

- Kleinstes Polynom der Form $u^q \oplus 1$, das sich restlos durch $G(u)$ teilen lässt, hat Grad $q = 7$
- Periode des Polynoms $G(u)$ ist $q = 7$
- Da Grad von $G(u)$ $p = 3$ ist, ist $G(u)$ wegen $q = 2^p - 1$ primitiv

Zyklische Binär-Codes

- Interpretation von Vektoren (z.B. CWs, NWs, Prüfspalten) als Polynome
  - $X = (x_1, \ldots, x_m) \leftrightarrow X(u) = \sum_{1 \le i \le m} x_i \cdot u^{m-i}$
  - Achtung: Indizes von CWs werden hier von links nach rechts, Koeffizienten von Polynomen aber von rechts nach links indiziert!
- Nutzen Potenzrestklassen eines Generatorpolynoms $G(u)$ zur Erstellung der Prüfspalten einer Prüfmatrix $P = (P_1, \ldots, P_n)$
  - $P_i(u) = u^{n-i} \bmod G(u)$
- Berechnung der Kontrollstellen
  - $Y = \sum_{1 \le i \le m} x_i \cdot (P_i)^T \;\leftrightarrow\; Y(u) = \sum_{1 \le i \le m} x_i \cdot \left(u^{n-i} \bmod G(u)\right) = \left(\sum_{1 \le i \le m} x_i \cdot u^{n-i}\right) \bmod G(u) = \left(X(u) \cdot u^k\right) \bmod G(u)$
- Darstellung von CWs
  - $C = XY \leftrightarrow C(u) = X(u) \cdot u^k \oplus Y(u)$
- Gültige CWs sind ein Vielfaches von $G(u)$
  - $C(u) \bmod G(u) = \left(X(u) \cdot u^k \oplus Y(u)\right) \bmod G(u) = \left(X(u) \cdot u^k\right) \bmod G(u) \oplus \left(\left(X(u) \cdot u^k\right) \bmod G(u)\right) \bmod G(u) = 0$

Beispiel

- Primitives Generatorpolynom $G(u) = u^3 + u + 1$
- Kodierung
  - $X = 1001$
  - $Y(u) = \left(X(u) \cdot u^k\right) \bmod G(u) = u^2 + u$
    - 1001000:1011=1010 Rest 110
  - $CW = 1001110$
- Prüfung
  - Angenommener Fehler $F = 0100000 \Rightarrow C^* = 1101110$
  - Fehlersyndrom $Z(u) = C^*(u)\ mod\ G(u) = u^2 + u + 1$
    - 1101110:1011=1111 Rest 111
  - Fehlerlokalisierung mit Prüfschema und Fehlersyndrom
  - $P = \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{pmatrix} \Rightarrow$ Fehler an 2. Stelle
  - Beachte: P enthält von re nach li genau die Folge $u^k \bmod G(u)$

Kodier- und Dekodierschaltung

- Kodierschaltung
  - Füge taktweise die Stellen $x_1 \ldots x_m$ eines NW X in ein Schieberegister ein, gefolgt von k Nullen
  - Nach Abschluss der Eingabe enthält das Schieberegister die Kontrollstellen $Y(u) = \left(X(u) \cdot u^k\right) \bmod G(u)$
- Dekodierschaltung
  - Füge taktweise die Stellen eines CW C in die Schaltung ein, und zwar in der Reihenfolge $x_1 \ldots x_m y_1 \ldots y_k$
  - Nach Abschluss der Eingabe enthält das Schieberegister das Fehlersyndrom $Z(u) = C(u) \bmod G(u)$

- Zyklische Binär-Codes können mit Hilfe rückgekoppelter Schieberegister effizient kodiert und dekodiert werden.
- Häufige Verwendung für Cyclic Redundany Check (CRC)
- Verwendete Generatorpolynome siehe http://de.wikipedia.org/wiki/Zyklische_Redundanzprüfung

Zyklische Hamming-Codes

- Zyklischer Binär-Code mit primitivem Generatorpolynom $G(u)$ zur Erzeugung der Prüfmatrix

- Sei k der Grad des Generatorpolynoms
  - Die Prüfmatrix hat $n = 2^k - 1$ Prüfspalten
  - Die CW-Länge ist ebenfalls $n = 2^k - 1 = m + k$

- Hamming-Distanz ist $h = 3$
  - $\bar{\eta} = h - 1 = 2$ Fehler erkennbar
  - 1 Fehler korrigierbar

- Wichtige Beispielanwendung
  - CRC-32 in Ethernet (IEEE 802.3)
  - $G(u) = u^{32} + u^{26} + u^{23} + u^{22} + u^{16} + u^{12} + u^{11} + u^{10} + u^8 + u^7 + u^5 + u^4 + u^2 + u^1 + u^0$

Zyklische Abramson-Codes

- Zyklischer Binär-Code
  - Abramson 1959
- Generatorpolynom $G(u) = G_1(u) \cdot (u + 1)$
  - $G_1(u)$ ist primitiv und von Grad k_1
  - $G(u)$ ist dann nicht primitiv und von Grad $k = k_1 + 1$
  - CW-Länge $n = 2^{k_1} - 1 = m + k$
- Erstellung der Prüfmatrix über die Potenzrestklassen von $G(u)$
- Hamming-Distanz $h = 4$
  - $\bar{\eta} = h - 1 = 3$ Fehler erkennbar
  - 1 Fehler korrigierbar

- Wichtige Beispielanwendung: CRC-CCITT (CRC-16)
  - CRC in HDLC
  - $G(u) = u^{16} + u^{12} + u^5 + u^0$

Weitere Zyklische Binär-Codes

- Fire-Code
  - Fire 1959
  - Erlaubt Korrektur von Büschelfehlern (Bitfehler an aufeinanderfolgenden CW-Stellen)

- BCH-Code
  - Bose-Chaudhuri-Hocquenghem 1960
  - Codes mit vorgebbarer Hamming-Distanz
  - Erlaubt Korrektur von Mehrfachfehlern

Leitungskodierung

- Anpassung eines Bitstroms an die Anforderungen eines Übertragungssystems

- Beispielanforderungen
  - Gleichstromfreiheit
    - Genauso viele Nullen wie Einsen benötigt
  - Taktrückgewinnung
    - „Self-Clocking“
    - Taktflanke bei der Übertragung eines jeden Bits vorhanden
  - Transformation von Bitstrom an n-wertige Logik
    - Beispiel: 3 Werte für Übertragung möglich: 1, 0, -1

- Häufiger Nachteil
  - Vermehrung der zu übertragenden Zeichen

Beispiele für Leitungskodierung

Block-Codes

- Mappt 2er Bitgruppen auf 3er Bitgruppen, die mindestens eine 0 und 1 enthalten
  - 00 → 001, 01 → 010
  - 10 → 101, 11 → 110
- Bitstrom vor Kodierung 00011011111110
- Bitstrom nach Kodierung 001010101110110110101
- Beispiele aus der Praxis
  - 4b5b-Code (100 Mbit/s Ethernet)
  - 8b10b-Code (Gigabit Ethernet, PCI-Express)
  - 64b66b-Code (10 Gigabit Ethernet)

Manchester-Codes

- Kodiert Einsen und Nullen als 0→1 und 1→0 Flanken innerhalb des übertragenen Symbols
- Dadurch definieren sich Nullen und Einsen
- Beispiel: 10 Mbit/s Ethernet
- Varianten existieren
- Bildquelle: Wikipedia