Codes (6)

Fehlererkennende (EDC) bzw. fehlerkorrigierende Codes (ECC)

Definitionen:

Codewort:= mit zusätzlichen (redundanten) Kontrollbits versehenes Quellwort

m:= Länge des Quellwortes (Anzahl der Nutzdatenbits)

r:= Anzahl der Kontrollbits

n:= m + r := Länge des Codewortes ---> C: $\{0,1\}^m \longrightarrow \{0,1\}^n$

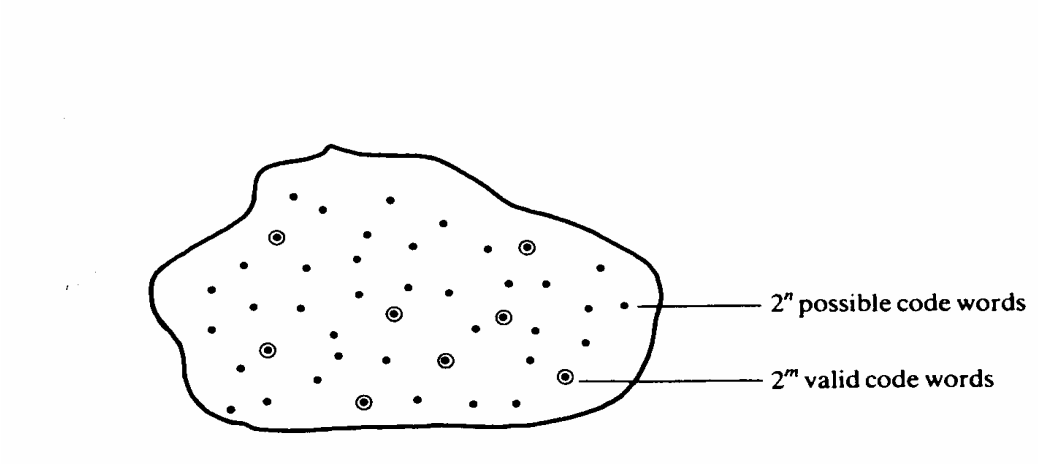
Seien x und y Codewörter im R^2_n . Dann heißt die Funktion
$d(x,y) := \sum x_i + y_i$ mit i= 1,.....,n Hammingdistanz von x und y. (+ := Addition modulo 2 := EOR-Funktion)

Die Hammingdistanz d(C) des gesamten Codes (Menge aller Codewörter) ist gleich der minimalen Hammingdistanz zwischen 2 Codewörtern dieses Codes.

==> Ist C ein Code fester Länge, dann ist d(C) >= 1, da C nach Definition eine injektive Abbildung ist.

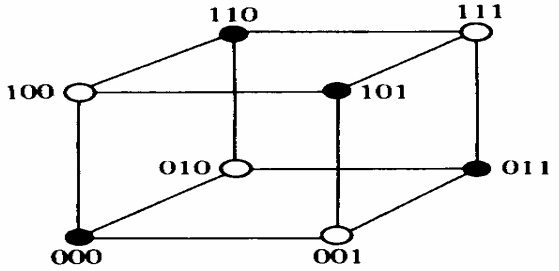
Codes (7)

Veranschaulichung des Codeprinzips:



räumliche Darstellung von Codes:

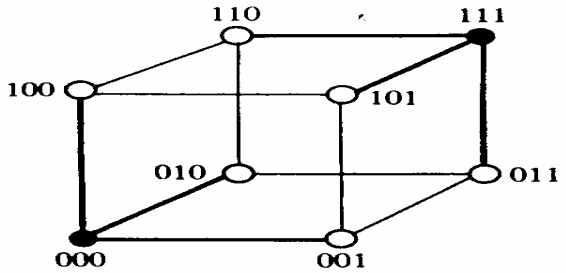
(a)



Valid code word

Invalid code word

(b) .



Valid code word

Invalid code word

(c)

Codes (8)

Begriffe:

Kippen während einer Codewortübertragung genau k ($k >= 1$) Bit, so liegt ein k-facher Übertragungsfehler vor, d,h, die Hamming-Distanz zwischen gesendetem Codewort und empfangenen Wort ist k.

Wir sprechen von einem k - Fehlermodell, wenn davon ausgegangen wird, dass bei einer Übertragung die Hammingdistanz h zwischen Codewort und gesendetem Wort im Bereich $0 <= h <= k$ liegt.

Definitionen:

Sei $C \subseteq B^n$ ein Code, d.h. die Bildmenge einer Abb. $c: A \dashrightarrow \{0,1\}^n$.

C ist ein *k-bit fehlererkennender* Code, wenn es dem Empfänger unter der Annahme des k - Fehlermodells möglich ist, zu erkennen, ob das gesendete Codewort durch einen Übertragungsfehler verfälscht wurde. Genauer:

C heißt auf k-Bit- Fehler prüfbar (erkennbar), wenn für jedes x aus C gilt:

$K(x,k) \cap C = \{x\}$, *wobei* $K(x,k) := \{y \mid d(y,x) \leq k\}$

C ist ein *k-bit fehlerkorrigierender* Code, wenn es dem Empfänger unter der Annahme des k - Fehlermodells möglich ist, bei einem Übertragungsfehler eigenständig die umgekippten Bits zu lokalisieren und somit das gesendete Codewort ohne Hilfe des Senders zu rekonstruieren. Genauer:

C heißt auf k-Bit- Fehler korrigierbar, wenn für alle x,y aus C gilt:

$K(x,k) \cap K(y,k) = \{\ \}$

---> C ist auf k-Bit- Fehler prüfbar (bzw. k-Bit- Fehler korrigierbar) gdw für alle x,y aus C gilt:

$d(x,y) \geq k+1$ *bzw.* $(d(x,y) \geq 2k+1$

Codes (10)

Anzahl der benötigten Kontrollbits für einen Einfach-ECC:

Word size	Check bits	Total size	Percent overhead
8	4	12	50
16	5	21	31
32	6	38	19
64	7	71	11
128	8	136	6
256	9	265	4
512	10	522	2

Einsatz von ECCs insbesondere bei:

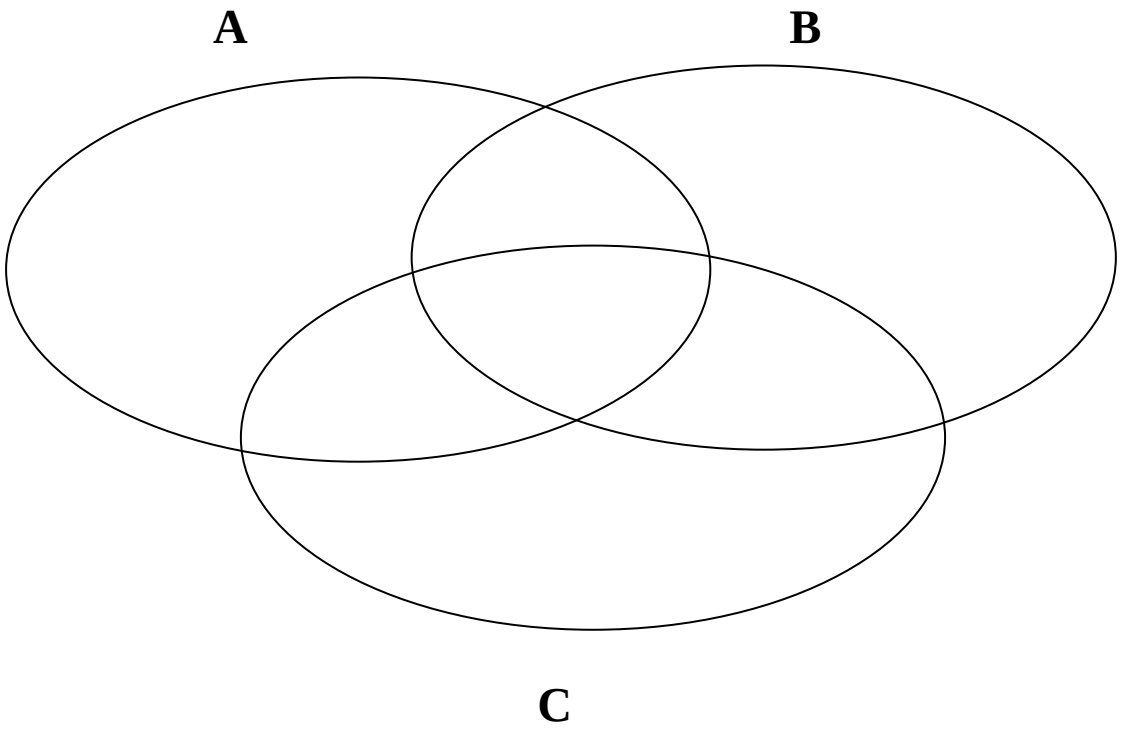
•stark gestörten Übertragungswegen

•militärische Datenübertragungen auf Funkwegen

•bei Computerspeichern, um Schreibfehler ohne anschließendes Korrekturlesen zu beheben

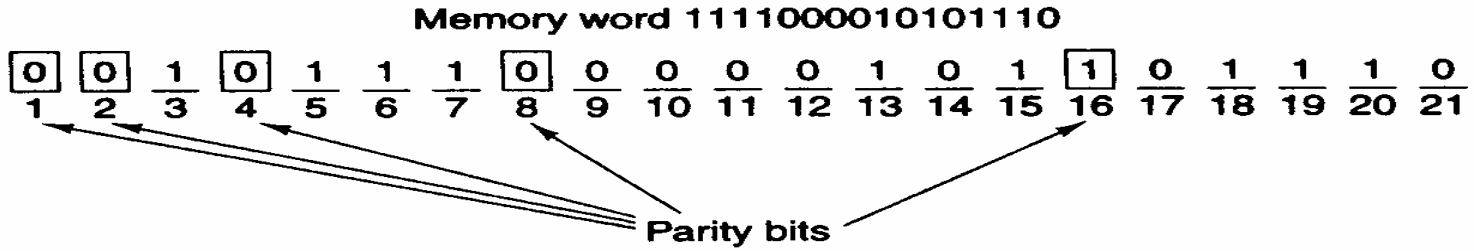
Codes (11)

Prinzipielle Idee hinter dem Hamming-Algorithmus:



Codes (13)

Beispiel:



Paritätsbit 1 kontrolliert (1,3,5,7,9,11,13,15,17,19,21 enthalten zusammen **6** Einsen)
Paritätsbit 2 kontrolliert (2,3,6,7,10,11,14,15,18,19, enthalten zusammen **6** Einsen)
Paritätsbit 4 kontrolliert (4,5,6,7,12,13,14,15,20,21 enthalten zusammen **6** Einsen)
Paritätsbit 8 kontrolliert (8,9,10,11,12,13,14,15 enthalten zusammen **2** Einsen)
Paritätsbit 16 kontrolliert (16,17,18,19,20,21 enthalten zusammen **4** Einsen)

Paritätsbit ***1*** **falsch** (1,3,5,7,9,11,13,15,17,19,21 enthalten zusammen **5** Einsen)
Paritätsbit 2 richtig (2,3,6,7,10,11,14,15,18,19, enthalten zusammen **6** Einsen)
Paritätsbit ***4*** **falsch** (4,5,6,7,12,13,14,15,20,21 enthalten zusammen **5** Einsen)
Paritätsbit 8 richtig (8,9,10,11,12,13,14,15 enthalten zusammen **2** Einsen)
Paritätsbit 16 richtig (16,17,18,19,20,21 enthalten zusammen **4** Einsen)

1 + 4 = 5 **====> Das Bit mit Positionsnummer 5 ist umgekippt!**

Codes (12)

Kurzdarstellung des Hamming-Algorithmus:

1. Kodierung auf der Sender (Schreib-)seite:

- Die Bits des Codewortes werden von 1 bis n (n=m+r) durchnumeriert, wobei die Positionsnummer als Binärzahl dargestellt wird.
- Als r Kontrollbits p_i werden jene Bits verwendet, deren Positionsnummer eine Zweierpotenz darstellt, also die Bits 1,2,4,....,2^{r-1} .
- Die restlichen Bits mit den Positionsnummern 3,5,6,7,9..... werden mit den m Datenbits x_i belegt.
- Jede Positionsnummer ist eindeutig als Summe von 2er - Potenzen darstellbar ---> Die Kontrollbits (2er-Potenzen) kontrollieren alle Positionsnummern, in denen Sie als Summand vorkommen. Sie sind so gesetzt, dass die Summe aller Einsen in den jeweiligen Positionsnummern gerade ist

2. Fehlerkontrolle auf der Empfänger (Lese-)seite:

- Auf der Basis des vorliegenden Codewortes werden erneut nach der gleichen Regel wie oben Kontrollbits p_i^* erzeugt
- Es wird das Syndrom S gebildet, wobei $S_i = p_i$ eor p_i^* für i = 1,2,4,....,2^{r-1} .
- Ist das zu kontrollierende Codewort fehlerfrei ---> S = 0

  Existiert genau ein Fehler ---> S als Binärzahl gelesen identifiziert die Positionsnummer des zu korrigierenden Bits.

Codes (13a)

Use of the Hamming - Code to correct burst errors: