Zusammenfassung DigCod

# Inhalt

[1 Inhalt 1](#_Toc139230906)

[2 Mathematische Grundlagen 3](#_Toc139230907)

[2.1 Begriffe 3](#_Toc139230908)

[2.2 Modulo 3](#_Toc139230909)

[2.3 Datenwörter 3](#_Toc139230910)

[2.4 Polynome 4](#_Toc139230911)

[2.4.1 Polynom-Multiplikation 4](#_Toc139230912)

[2.4.2 Polynom-Division 4](#_Toc139230913)

[2.4.3 Polynom-Division rückgängig machen 4](#_Toc139230914)

[2.5 Zyklische Gruppe / Galois 5](#_Toc139230915)

[3 Wahrscheinlichkeit 5](#_Toc139230916)

[3.1 Begriffe 5](#_Toc139230917)

[3.2 Wahrscheinlichkeit 6](#_Toc139230918)

[3.3 Kombinatorik 6](#_Toc139230919)

[3.4 Lotto 6](#_Toc139230920)

[3.5 Bitfehler 6](#_Toc139230921)

[4 Informationstheorie 7](#_Toc139230922)

[4.1 Begriffe und Formeln 7](#_Toc139230923)

[4.2 Shannon’sches Codierungstheorem 7](#_Toc139230924)

[4.3 Bedingte Wahrscheinlichkeiten (mit gedächtnis) 7](#_Toc139230925)

[5 Komprimierung 9](#_Toc139230926)

[5.1 Huffman-Codierung 9](#_Toc139230927)

[5.2 Lauflängenkomprimierung 10](#_Toc139230928)

[5.3 Lempel-Ziv 10](#_Toc139230929)

[5.3.1 Codierung 10](#_Toc139230930)

[5.3.2 Decodierung 10](#_Toc139230931)

[6 Verschlüsselung 11](#_Toc139230932)

[6.1 Symmetrische Verfahren 11](#_Toc139230933)

[6.1.1 Substitionsverfahren/Caesar Chiffre 11](#_Toc139230934)

[6.1.2 Transpositionsverfahren 11](#_Toc139230935)

[6.1.3 Vigenère-Chiffre 11](#_Toc139230936)

[6.1.4 Data Encryption Standard (DES) / Advanced Encryption Standard (AES) 12](#_Toc139230937)

[6.2 Asymmetrische Verfahren 12](#_Toc139230938)

[6.2.1 RSA 12](#_Toc139230939)

[6.2.2 Probleme Handhabung asymmetrisches Verfahren 13](#_Toc139230940)

[6.2.3 Kombination Komprimierung/Verschlüsselung 13](#_Toc139230941)

[7 Kanalmodell 13](#_Toc139230942)

[7.1 Kanalmatrix 14](#_Toc139230943)

[7.2 Maximum-Likelihood-Verfahren 14](#_Toc139230944)

[7.3 Transinformation 15](#_Toc139230945)

[7.4 Verbundentropie 16](#_Toc139230946)

[7.5 Äquivokation/Verlust/Rückschlussentropie 16](#_Toc139230947)

[7.6 Irrelevanz/Rauschen/Streuentropie 16](#_Toc139230948)

[8 Blockcodes 16](#_Toc139230949)

[8.1 Begriffe und Formeln 16](#_Toc139230950)

[8.2 Hammingcode 17](#_Toc139230951)

[8.2.1 Kontrollstellen eines Codeworts berechnen 17](#_Toc139230952)

[8.2.2 Fehlersyndrome 17](#_Toc139230953)

[8.2.3 Mehrfachaddition zur Bestimmung aller gültigen Codeworte 17](#_Toc139230954)

[8.2.4 Mehrfachaddition zur Bestimmung der Prüfmatrix 18](#_Toc139230955)

[8.2.5 Polynomdivision zur Ermittlung eines Fehlersyndroms an Stelle 18](#_Toc139230956)

[8.3 Korrigierkugel 18](#_Toc139230957)

[8.4 Dichtgepackter Code 19](#_Toc139230958)

[8.5 Zyklische Abramson Codes / CRC Codes 19](#_Toc139230959)

[9 Faltungscodes 19](#_Toc139230960)

[9.1 Zustandsdiagramm 20](#_Toc139230961)

[9.2 Decodierung mithilfe des Netzdiagramms 20](#_Toc139230962)

[10 Binärzahlen 21](#_Toc139230963)

[10.1 Hexadezimalsystem 21](#_Toc139230964)

[10.1.1 Grösste darstellbare Zahl 21](#_Toc139230965)

[10.2 Darstellung von Zahlen 21](#_Toc139230966)

[10.3 Berechnungen 22](#_Toc139230967)

[10.3.1 Umwandlung Dezimal – Binär 22](#_Toc139230968)

[10.3.2 Addition 22](#_Toc139230969)

[10.3.3 Subtraktion 22](#_Toc139230970)

[10.3.4 Multiplikation 22](#_Toc139230971)

[10.3.5 Division 22](#_Toc139230972)

[10.4 Signed Integers 23](#_Toc139230973)

[10.4.1 Umwandlung negative Dezimalzahl ins Zweierkomplement (Inversionsverfahren) 23](#_Toc139230974)

[10.4.2 Umwandlung Zweierkomplement in negative Dezimalzahl (Inversionsverfahren) 23](#_Toc139230975)

[11 Boolsche Logik 24](#_Toc139230976)

[11.1 Arität von Funktionen 24](#_Toc139230977)

[11.2 Unäre Funktionen 24](#_Toc139230978)

[11.3 Binäre Funktionen 24](#_Toc139230979)

[11.4 Terme 25](#_Toc139230980)

[11.5 Disjunktive Normalformen 25](#_Toc139230981)

[12 ASCII 25](#_Toc139230982)

[13 Fixkommazahlen 26](#_Toc139230983)

[13.1 Multiplikation 26](#_Toc139230984)

[14 Gleitkommazahlen 26](#_Toc139230985)

[14.1 Dezimalzahl als Float darstellen 27](#_Toc139230986)

[14.2 Addition von Floats 28](#_Toc139230987)

[14.3 Runden 28](#_Toc139230988)

# Mathematische Grundlagen

## Begriffe

* **Mengen**: ℤ (ganze Zahlen), ℤ = {…, -2, -1, 0, 1, 2, …}
* **Rechenoperationen**: +, -, \*, /
* **Wertigkeit**: entspricht der Basis hoch der Stelle der ursprünglichen Zahl, angefangen mit 0 beim Least Significant Bit (LSB), z.B. bei Hexadezimal   
  Chart

  Description automatically generated
* **Algebraische Strukturen**:
  + **Gruppe** (werden Elemente aus der Menge verknüpft, so muss das Ergebnis wieder ein Element der Menge ergeben):
  + **Ring**:
  + **Körper**:
  + ein Ring ist ein Körper, wenn:
    - jedes Element des Rings außer der Null ein (multiplikatives) Inverses hat
    - die Multiplikation kommutativ ist: ab = ba
    - das Distributivgesetz gilt: a(1-b) = a - ab
    - wenn er eine 1 hat (multiplikatives neutrales Element)

## Modulo

Wenn wir eine Multiplikation durchführen und die Abgeschlossenheit im haben wollen, müssen wir rechnen.

Beispiel: aber

Text

Description automatically generated with low confidence

## Datenwörter

Alle vier Darstellungsweise hier sind dasselbe:

* Zahl:
* Tupel:
* Vektor:
* Polynom:

## Polynome

Dezimalzahl 1205 als Polynom:

Beispiel eines Polynoms 5-ten Grades mit Koeffizienten aus :

### Polynom-Multiplikation

### Polynom-Division

Aufgabe:

A picture containing chart

Description automatically generated

1. Dividieren:
2. Multiplizieren:
3. Subtrahieren:
4. Nun geht es von vorne los
5. Dividieren:
6. Multiplizieren:
7. Subtrahieren:
8. Usw.

### Polynom-Division rückgängig machen

Text

Description automatically generated with medium confidence

Mod 2 nicht vergessen!

## Zyklische Gruppe / Galois

Primitives Polynom:

Text, letter

Description automatically generated

Die Zykluslänge ist somit 15.

Folgende Elemente wurden erzeugt:

Die binären Zahlen sind aus der Polynomschreibweise oben abgeleitet.

# Wahrscheinlichkeit

## Begriffe

* Ergebnismenge: Menge aller möglichen Ausgänge, wird mit bezeichnet
* Ein einzelnes Element aus der Ergebnismenge wird mit bezeichnet

## Wahrscheinlichkeit

Wahrscheinlichkeit heisst auf Englisch probability, daher sind die Formeln jeweils mit p

* Wahrscheinlichkeit eines Ereignisses:
* Wahrscheinlichkeit, dass ein Ereignis nicht auftritt:
* Wahrscheinlichkeit eines unmöglichen Ereignisses:
* Wertebereich der Wahrscheinlichkeit:
* Seien E1 und E2 zwei Ereignisse.
  + Formel, wenn Ereignis E2 nur eintreten kann, wenn E1 bereits eingetreten ist:
  + Formel, wenn die Ereignisse unabhängig sind:

## Kombinatorik

Buchstaben:

* Anzahl der Zahlenkugeln in der Kiste n
* Anzahl der Ziehungen k
* Anzahl Möglichkeiten N

Es gibt 4 Varianten:

* Geordnet, mit Wiederholung/Zurücklegen (die Reihenfolge ist relevant und die gleiche Zahl kann mehrmals vorkommen)
* Geordnet, ohne Wiederholung/Zurücklegen (die Reihenfolge ist relevant und die gleiche Zahl kann nicht mehrmals vorkommen)
* Ungeordnet, mit Wiederholung/Zurücklegen (die Reihenfolge ist egal und die gleiche Zahl kann mehrmals vorkommen)
* Ungeordnet, ohne Wiederholung/Zurücklegen (die Reihenfolge ist egal und die gleiche Zahl kann nicht mehrmals vorkommen)

## Lotto

Lotto ist ein Beispiel einer geordneten Stichprobe ohne Wiederholung.

Beispiel Formel für 5 richtige aus 49 Zahlen:

## Bitfehler

Wahrscheinlichkeit Bitfehler:

Wahrscheinlichkeit, dass Datenblock mit 153600 Bits fehlerhaft ist:

Wahrscheinlichkeit, dass höchstens 3 Fehler auftreten:

# Informationstheorie

## Begriffe und Formeln

* Zeichenvorrat: z.B.
* Anzahl der Zeichen im Zeichenvorrat:
* Entscheidungsgehalt: Mass für den Aufwand, der zur Bildung einer Nachricht notwendig ist.
* Entscheidungsfluss
* Chart, line chart

  Description automatically generatedInformationsgehalt: sagt aus, wie viele Elementarentscheidungen zur Bestimmung dieses Zeichens zu treffen sind (je kleiner die Wahrscheinlichkeit eines Zeichens, desto grösser der Informationsgehalt)
* Mittlerer Informationsgehalt / Entropie
* Tatsächliche Codewortlänge
* Entropie des Codes/Mittelwert der Codewortlänge

## Shannon’sches Codierungstheorem

* Redundanz der Quelle:
* Redundanz des Codes:

## Bedingte Wahrscheinlichkeiten (mit gedächtnis)

In der Sprache sind gewisse Zeichen nach gewissen Zeichen wahrscheinlicher als andere.

Wir haben folgendes Markov-Diagramm:

A diagram of a network

Description automatically generated with low confidence

Daraus lässt sich folgendes ablesen:

A picture containing text, screenshot, number, font

Description automatically generated

Daraus lässt sich mithilfe eines GLS berechnen:

A picture containing text, screenshot, font, number

Description automatically generated

Daraus lässt sich dann berechnen:

A picture containing text, screenshot, number, font

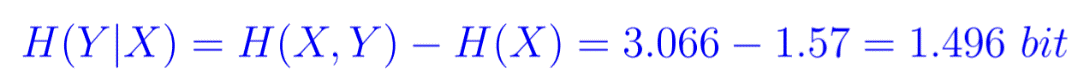
Description automatically generated

Usw.

A picture containing text, font, white, handwriting

Description automatically generated

Verbundentropie:



Bedingte Entropie:

A picture containing font, typography, white, text

Description automatically generated ist von der Sprache gegeben

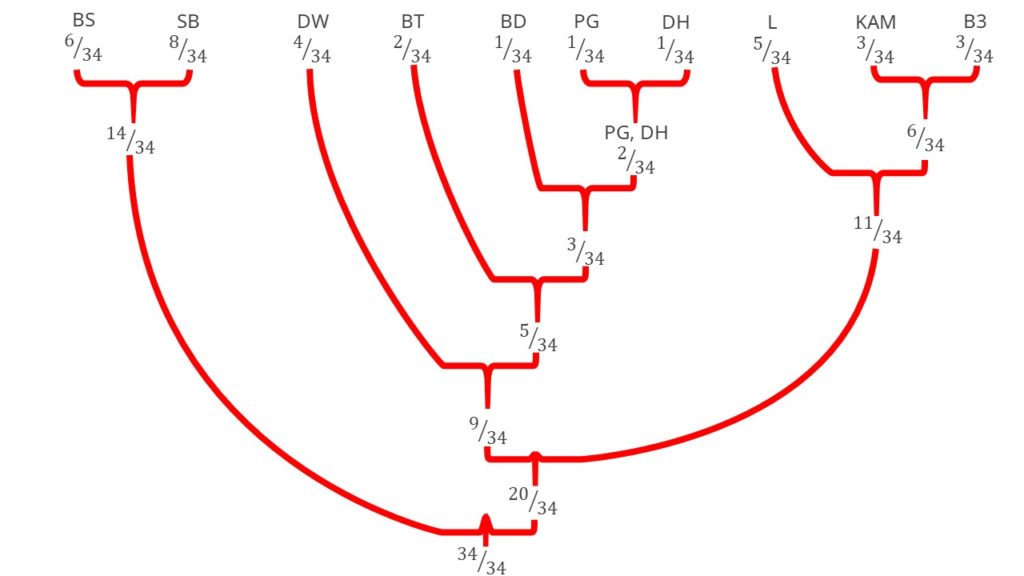


# Komprimierung

Das Ziel der Datenkomprimierung ist, den Aufwand der Datenspeicherung und Datenübertragung zu reduzieren. Das heisst Entfernen von Redundanz und Irrelevanz.

## Huffman-Codierung

Bei dieser Codierung werden oft vorkommenden Zeichen kurze Codeworte zugewiesen und seltener vorkommenden Zeichen längere Codeworte. Die Präfixeigenschaft (kein anderes CW beginnt mit einer Zeichenfolge, die ein anderes CW definiert) wird erfüllt.

1. Die Zeichen aufsteigend nach Wahrscheinlichkeit aufschrieben  
   
2. Die zwei Zeichen mit der niedrigsten Auftretenswahrscheinlichkeit auswählen und zu einem Zeichen verschmelzen. Die Auftretenswahrscheinlichkeit wird addiert und das Zeichen neu einsortiert. Diesen Schritt wiederholen, bis es nur noch einen Knoten mit der Wahrscheinlichkeit 1 gibt.  
   
3. Nun muss der Baum umgedreht werden, allen linken Kanten eine 0 und allen rechten Kanten eine 1 zugewiesen werden. Die Codewörter können dann von oben her abgelesen werden.  
   A picture containing diagram, plan, design

   Description automatically generated

## Lauflängenkomprimierung

Lauflängenkomprimierung wird auch Run Length Encoding (RLE) oder Run Length Coding (RLC) genannt. Sie wird von Bildformaten wie BMP, PCX oder TIFF verwendet. Sie basiert auf der Verkürzung von Wiederholungen aufeinanderfolgender Elemente.

Beispiel bei einem Wort: Agggbbehfffgggg 🡪 A3g2beh3f4g

Bei der Codierung von Bitfolgen ist es etwas einfacher, es existieren ja nur Folgen von 0 und 1. Der Codierer und Decodierer müssen sich anfänglich darauf einigen, ob mit 0 oder 1 begonnen wird.

Beispiel: 1111 1110 0000 1000 0001 11111111 1110 0000 1000 0001 1111 🡪 7 5 1 6 5 (Start mit 1)  
Da die grösste Zahl im Code 7 ist, werden 3 Bits benötigt pro Stelle ().  
Das Codewort ist also: 111 101 001 110 101

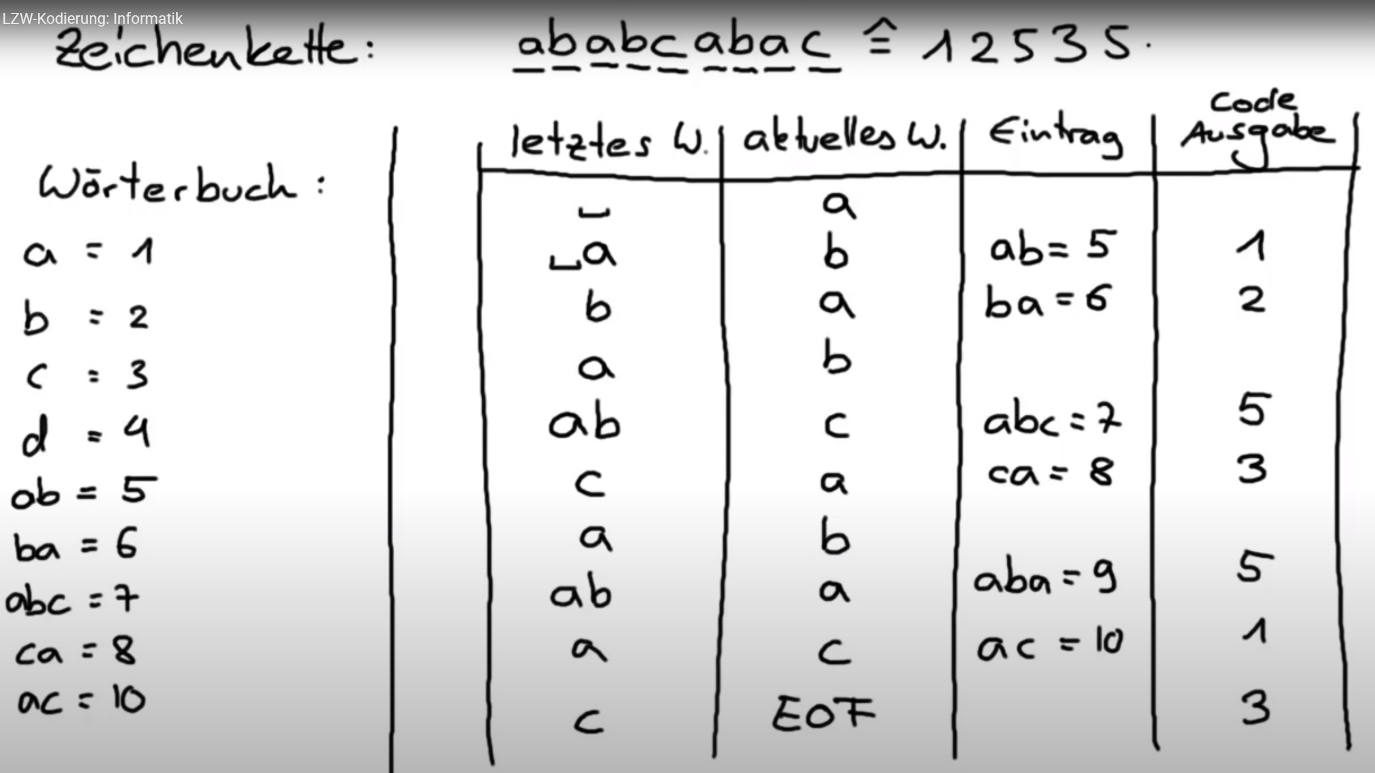
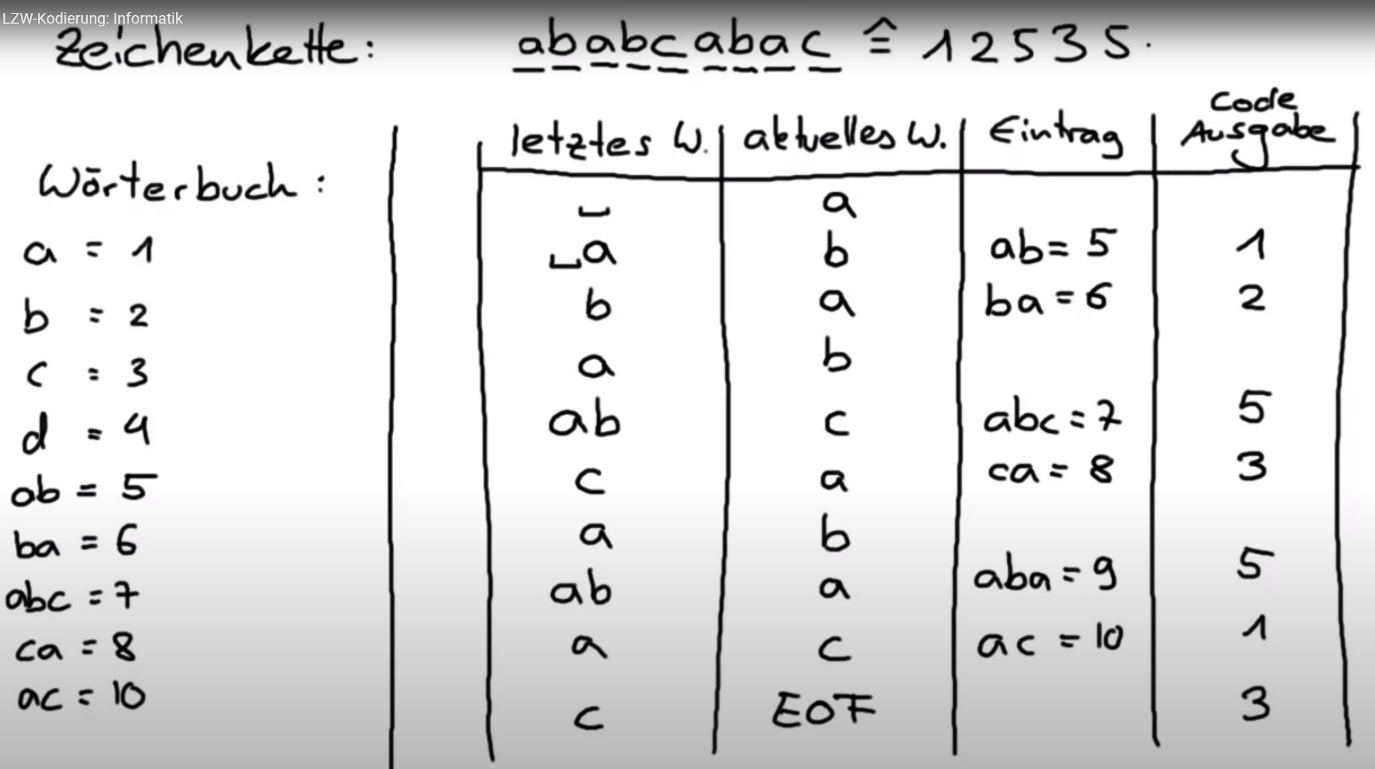
## Lempel-Ziv

Dieses Verfahren ist gut geeignet, wenn der zu komprimierende Code wiederkehrende Muster oder Phrasen hat.

### Codierung

A picture containing table

Description automatically generated



### Decodierung

Text, letter

Description automatically generated

# Verschlüsselung

## Symmetrische Verfahren

### Substitionsverfahren/Caesar Chiffre

Table

Description automatically generated

Zwar sieht der Chiffretext unlesbar aus, jedoch sind die statistischen Eigenschaften von Klar- und Chiffriertext unverändert. Wenn wir die Sprache kennen und die Probe gross genug ist, kann der Schlüssel leicht ermittelt werden. Ausserdem ist die Schlüsselanzahl recht übersichtlich, es braucht nicht mal einen Computer, um alle auszuprobieren. Sowohl Sprache als auch Textlänge können einen entscheidenden Einfluss haben auf den Erfolg der Analyse.

### Transpositionsverfahren

Text

Description automatically generated with medium confidence

### Vigenère-Chiffre

A picture containing background pattern

Description automatically generatedIn diesem Beispiel soll «beispielklartext» mit dem Schlüssel «APFELSTRUDEL» verschlüsselt werden. Der Chiffretext ist «BTNWAAXCEOECTTCX». Man sucht den Klartextbuchstaben links in der 1. Spalte und geht in dieser Zeile so weit nach rechts, bis man in der obersten Zeile den Buchstaben des Schlüssels gefunden hat. Nun kann man dort den Chiffre-Buchstaben ablesen.

### Data Encryption Standard (DES) / Advanced Encryption Standard (AES)

A diagram of a block diagram

Description automatically generated with low confidence

DES wurde als offizieller Standard für die US-Regierung im Jahr 1977 bestätigt und wird seither international vielfach eingesetzt. Die Beteiligung der NSA am Design des Algorithmus hat immer wieder Anlass zu Spekulationen über seine Sicherheit gegeben. Heute wird DES aufgrund der verwendeten Schlüssellänge von nur 56 Bits für viele Anwendungen als nicht ausreichend sicher erachtet.

AES ist der Nachfolger von DES und gilt derzeit als sicher.

## Asymmetrische Verfahren

### RSA

#### Public und private Key bestimmen

1. Zwei beliebige Primzahlen, *p* und *q* bestimmen
2. bestimmen
3. bestimmen
4. Eine beliebige Zahl bestimmen,
5. Multiplikatives Inverses *b* von *a* bestimmen:
6. Öffentliche Schlüssel *e* und *n* veröffentlichen
7. Der private Schlüssel bleibt geheim

#### Verschlüsselung

#### Entschlüsselung

1. Ergebnis der Entschlüsselung mithilfe der Buchstabentablle wieder in Text umwandeln

#### Multiplikatives Inverses

Wenn , dann ist b das multiplikative Inverse von a

Gibt es nur, wenn

#### Eulersche Phi-Funktion

Beispiel:

Liste:

gilt aber nur, wenn

#### Erweiterter Euklidischer Algorithmus

Vorgabe:

Tabelle:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Ablauf | x | y | q | r | u | s | v | t |
| Initialisieren |  |  |  |  | 1 | 0 | 0 | 1 |
| Wiederholen |  |  |  |  |  |  |  |  |

Beispiel:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| x | y | q | r | u | s | v | t |
| 99 | 79 | 1 | 20 | 1 | 0 | 0 | 1 |
| 79 | 20 | 3 | 19 | 0 | 1 | 1 | -1 |
| 20 | 19 | 1 | 1 | 1 | -3 | -1 | 4 |
| 19 | 1 (ggT) | 19 | 0 | -3 | 4 | 4 | -5 (mod 99 = Multiplikat. Inverses) |

### Probleme Handhabung asymmetrisches Verfahren

* Performance: Wie Sie sich nach den bisher gelösten Aufgaben vorstellen können, sind asymmetrische Algorithmen sehr langsam (ca. 10’000x langsamer als symmetrische Verfahren).
* Infrastruktur: Der öffentliche Schlüssel muss über eine vertrauenswürdige Stelle abrufbar sein. Die Sicherstellung der Vertrauenswürdigkeit und die Bestimmung der Herkunft des öffentlichen Schlüssels sind mit grossem Aufwand verbunden.
* Mehrere Empfänger: Bei mehreren Empfängern muss die Nachricht mit dem öffentlichen Schlüssel jedes einzelnen Empfängers verschlüsselt werden.
* Man-In-The-Middle: Wenn es ein Angreifer schafft, seinen öffentlichen Schlüssel als den des eigentlichen Empfängers anzupreisen, kann er mit seinem privaten Schlüssel die Nachricht entschlüsseln. Um nicht aufzufallen, kann er zudem die Nachricht weiter mit dem öffentlichen Schlüssel des eigentlichen Empfängers wieder verschlüsseln und dann die Nachricht weiterschicken. Um das zu verhindern, muss die Authentizität des öffentlichen Schlüssels gewährleistet sein.

### Kombination Komprimierung/Verschlüsselung

Da eine gute Verschlüsselung ihre Daten pseudozufällig macht, hat eine anschliessende Datenkompression keine Wirkung mehr - die Zeichen werden gleich wahrscheinlich. Sie müssen daher die Daten zuerst komprimieren und erst anschliessend verschlüsseln. Zudem maximiert die Komprimierung die Entropie der Nachricht und reduziert so etwas die Angriffsfläche für statistische Angriffe auf die Verschlüsselung.

# Kanalmodell

Diagram

Description automatically generated

## Kanalmatrix

Diagram

Description automatically generated

T=Transinformation

Nicht gestört (Einheitsmatrix, T=1): Vollständig gestört (T=0):

## Maximum-Likelihood-Verfahren

A picture containing text, diagram, screenshot

Description automatically generated



Pro Spalte den höchsten Wert einkreisen, wenn alle gleich sind das Zeichen nehmen, das noch nicht abgedeckt wurde. Der Entscheider nimmt an, wenn er ein empfangen hat, dass das mit der höchsten Wahrscheinlichkeit gesendet wurde.

Table

Description automatically generated with low confidence

Text

Description automatically generated

A picture containing text, font, handwriting, line

Description automatically generated

## Transinformation

Wir können feststellen, dass bei der Datenübertragung über einen gestörten Kanal «Informationen» verloren gehen. Das bedeutet, der mittlere Informationsgehalt (die Entropie) nimmt ab.

Diagram

Description automatically generatedEin Bild, das Text enthält.

Automatisch generierte Beschreibung

* Transinformation beschreibt den maximalen, fehlerfreien Informationsfluss über einen Kanal. (Es gibt noch keine Aussage wie dies zu erreichen ist)
* Verändert sich die Entropie der Quelle, verändert sich auch die Transinformation.
* Nimmt die Fehlerwahrscheinlichkeit zu, so verringert sich die Transinformation.
* Transinformation wird durch die Quelle bestimmt
* Sind alle Positionen der Kanalmatrix gleich besetzt, so wird die Transinformation d.h. und zwar unabhängig von der Entropie am Kanaleingang.
* Beispiel:

## Verbundentropie

Das paarweise Auftreten aller Möglichkeiten am Kanaleingang und Kanalausgang.

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

## Äquivokation/Verlust/Rückschlussentropie

Beschreibt die Ungewissheit über ein gesendetes Zeichen bei bekannten Empfangszeichen.

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

bei fehlerfreiem Kanal

## Irrelevanz/Rauschen/Streuentropie

Beschreibt die Ungewissheit der empfangenen Zeichen bei bekanntem Sendezeichen.

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Beispiel eines verlustfreien und rauschbehafteten Kanals: A picture containing font, line, electric blue, design

Description automatically generated

Die Summe der Zeilen muss sein. Die Matrix ist asymmetrisch, weil nicht existiert.

# Blockcodes

## Begriffe und Formeln

* Anzahl Worte = 2 bei Binärcode
* n = Anzahl Codestellen
* m = Anzahl Nachrichtenstellen
* k = Anzahl Kontrollstellen, wenn man eine Prüfmatrix hat, kann k aus der Einheitsmatrix am Ende der Prüfmatrix abgelesen werden, z.B. für :  
  

Ausserdem ist kann k aus dem Generatorpolynom abgelesen werden, es ist die höchste Potenz.

* Gültige Codeworte:
* Mögliche CW:

## Hammingcode

* Text

  Description automatically generated with low confidenceHammingdistanz h: beschreibt den minimalen Abstand zwischen zwei gültigen Codeworten im gesamten Coderaum. Wenn man eine Liste von Codeworten hat, kann man auch schauen, in wie vielen Bits sich die Codeworte unterscheiden, z.B. bei :
* ist die Anzahl Prüfvektoren, die addiert werden müssen, um den Nullvektor zu erhalten
* meistens
* Anzahl erkennbare Fehler
* Anzahl korrigierbare Fehler bei geradem h:
* Anzahl korrigierbare Fehler bei ungeradem h:
* Treten mehr Fehler auf als korrigierbar sind, so wird entweder falsch korrigiert oder der Fehler wird nicht erkannt.

### Kontrollstellen eines Codeworts berechnen

Es wird die Matrix aus [8.1 Begriffe und Formeln](#_Begriffe_und_Formeln) verwendet, CW = 10110000010 und h = 4

Aus dem Codewort lesen wir ab, dass relevant sind (alle Einsen).

Aus der Matrix lesen wir folgende Formeln ab (alle Einsen der Zeilen von oben nach unten):

Text

Description automatically generated

Die Kontrollstellen sind somit und das komplette Codewort

### Fehlersyndrome

Fehlersyndrome können aus der Prüfmatrix abgelesen werden. Es sind jeweils die Spalten mit dem Fehler. Bei mehreren Fehlern müssen die Spalten addiert und mod 2 gerechnet werden.

Beispiel aus der Matrix in [8.1 Begriffe und Formeln](#_Begriffe_und_Formeln), Fehler bei :

oder bei und :

### Mehrfachaddition zur Bestimmung aller gültigen Codeworte

Benötigt: Generatorpolynom binär & Anzahl Nachrichtenstellen m (siehe [8.1 Begriffe und Formeln](#_Begriffe_und_Formeln))

Zuerst müssen alle möglichen Nachrichten erstellt werden, es gibt Nachrichten. Bei wären es z.B. und .

Text

Description automatically generatedDanach wird für jede mögliche Nachricht das Generatorpolynom mehrfach addiert, bis vor dem vertikalen Strich nur noch Nullen stehen. Beispiel:

Icon

Description automatically generated



Das erste mögliche Codewort sind immer alles Nullen und das Letzte alle Einsen.

### Mehrfachaddition zur Bestimmung der Prüfmatrix

Benötigt: Generatorpolynom binär & Anzahl Nachrichtenstellen m (siehe [8.1 Begriffe und Formeln](#_Begriffe_und_Formeln))

Zuerst müssen Codewörter erstellt werden, bei dem jeweils genau eine Nachrichtenstelle 1 ist, z.B. und .

Danach wird für jedes Codewort das Generatorpolynom mehrfach addiert, bis vor dem vertikalen Strich nur noch Nullen stehen. Beispiel:

Das Ergebnis ist dann eine Prüfmatrix mit Zeilen und Spalten. Am Ende befindet sich die Einheitsmatrix. Die Additionsbeispiel ist in der Matrix die zweite Spalte, weil das zweite Bit auf 1 gesetzt wurde:

Text, whiteboard

Description automatically generatedIcon

Description automatically generated



### Polynomdivision zur Ermittlung eines Fehlersyndroms an Stelle

Benötigt: Stelle (hier: ) & Generatorpolynom (hier: )

A picture containing text, font, line, handwriting

Description automatically generated

## Korrigierkugel

Das Zentrum der Korrigierkugel liegt auf jedem gültigen Codewort (weiss) und schliesst alle umliegenden, ungültigen Codeworte (grau), die sicher korrigiert werden können, ein. Die Anzahl der sicher korrigierbaren Fehler kann deshalb als Radius der Korrigierkugel betrachtet werden. Die Hammingdistanz ist die kürzeste Distanz zwischen zwei gültigen Codeworten unter Betrachtung aller Paare von gültigen Codeworten im gesamten Coderaum. Die Anzahl der sicher erkennbaren Fehler ist deshalb .

A picture containing diagram, circle, line, plot

Description automatically generated

## Dichtgepackter Code

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

Der Coderaum ist dichtgepackt wenn:

Beispiel für :

## Zyklische Abramson Codes / CRC Codes

Hammingdistanz

Wird gebildet durch Multiplikation des primitiven Polynoms mit

Rückgekoppeltes Schiebregister für :

A picture containing line, diagram, font, plot

Description automatically generated

Um ein reduzibles Generatorpolynom zu reduzieren, teilt man es durch

# A picture containing text, screenshot, number, calendar Description automatically generatedA picture containing handwriting, text, drawing, sketch Description automatically generatedFaltungscodes

A diagram of a block diagram

Description automatically generated with low confidence

Beispiel für Codierung von

Ausgabe ist:

In der Eingabe sind 3 Tailbits enthalten (6 in der Ausgabe).   
Die Anzahl entspricht der höchsten Potenz des Generatorpolynoms.  
Für die Impulsantwort gibt man dem Faltungscodierer eine 1 und dann nur noch Nullen. Die Impulsantwort ist dann das, was rauskommt dabei.

Coderate = Eingangsbits/Ausgangsbits

## Zustandsdiagramm

A picture containing diagram, technical drawing, line, plan

Description automatically generatedA picture containing handwriting, font, line, calligraphy

Description automatically generated

* Die 4 Boxen im Zustandsdiagramm sind jeweils alle möglichen Kombinationen der x in der Encoder-Schaltung
* Die Zahl vor dem / ist die Zahl, welche man vorne in die Schaltung reingibt
* Die Zahl nach dem / ist die Zahl, welche hinten an der Schaltung herauskommt mit Einbezug der reingegebenen Zahl und der aktuellen Belegung der x

## Decodierung mithilfe des Netzdiagramms

Verw. Encoder-Schaltung und Zustandsdiagramm:

Empfangene Bits:

Netzdiagramm:

A picture containing line, diagram, text, map

Description automatically generated

* Die orangen Zahlen sind jeweils die gemachten Fehler
* Wenn zwei Pfeile auf eine Kugel zeigen, wird die tiefere Zahl gewählt und der andere Weg geblockt
* Es ist definiert, dass Start und Ende bei liegen
* Wenn alle orangen Einträge gemacht sind, geht man vom Ende aus und sucht den Weg mit den geringsten Kosten.
* Das gesendete Codewort (Lösung) sind dann die Zahlen vor dem / auf dem Weg abzüglich der Tailbits, in diesem Fall also oder
* Talbots sind immer die Anzahl Quadrate im Schieberegister

# Binärzahlen

Computer arbeiten mit dem Binärsystem, weil technisch zwei Zustände einfach unterscheidbar sind.

A screenshot of a number

Description automatically generated with low confidenceDezimalsystem: Binärsystem:

A picture containing font, text, line, design

Description automatically generated

Most Significant Bit (MSB) Least Significant Bit (LSB)

Nibble: 4 Bit Notation einzelner Bits von : oder

Byte: 8 Bit Notation von Unterbereichen von : oder

A picture containing text, screenshot, font, number

Description automatically generatedPräfixe:

## Hexadezimalsystem

Ein Nibble (4 Bit) können jeweils mit einer Hexadezimalstelle beschrieben werden. Daher ist es sehr gut geeignet, um Binärzahlen zu verkürzen.

A close-up of numbers

Description automatically generated with low confidence

### Grösste darstellbare Zahl

Die grösste Hex-Zahl mit Bit hat an der Most Significant Stelle eine .

usw.

## Darstellung von Zahlen

## Berechnungen

### Umwandlung Dezimal – Binär

Man teilt jeweils durch 2. Wenn das Ergebnis eine ganze Zahl ist, wird eine 0 geschrieben. Wenn es eine .5 Zahl ist, wird eine 1 geschrieben und mit der Abrundung weitergerechnet. Abgebrochen wird bei 0.5. Die Binärzahl wird dann von unten nach oben abgelesen. Beispiel:

|  |  |  |  |
| --- | --- | --- | --- |
| ***Dezimalzahl*** | ***Operation*** | ***Ergebnis*** | ***Bit*** |
| 22 | /2 | 11 | 0 |
| 11 | /2 | 5.5 | 1 |
| 5 | /2 | 2.5 | 1 |
| 2 | /2 | 1 | 0 |
| 1 | /2 | 0.5 | 1 |

in binär:

### Addition

Bei der Addition von Binärzahlen wird von rechts begonnen. Es wird jeweils das Bit des ersten Summanden, des zweiten Summanden und das Carry Bit addiert und dann gerechnet. Das Carry Bit wird auf 1 gesetzt, wenn das vorherige Ergebnis mehr als 1 war vor . Beispiel:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| *Binär* | | | | | | *Dezimal* |
| 1. Summand |  | 1 | 1 | 0 | 1 | 13 |
| 2. Summand |  | 1 | 0 | 0 | 1 | 9 |
| Carry-Bit | 1 | 0 | 0 | 1 | - | = |
| Ergebnis | **1** | **0** | **1** | **1** | **0** | **22** |

### Subtraktion

Bei der Subtraktion von Binärzahlen wird von rechts begonnen. Es wird jeweils das Bit des Subtrahenden und Carry-Bits vom Minuenden abgezogen und dann gerechnet. Das Carry Bit wird auf 1 gesetzt, wenn das Ergebnis kleiner als 0 war, vor . Beispiel:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***Binär*** | | | | | ***Dezimal*** |
| Minuend | 1 | 1 | 0 | 1 | 13 |
| Subtrahend | 1 | 0 | 0 | 1 | 9 |
| Carry-Bit | 0 | 0 | 0 | - | = |
| Ergebnis | **0** | **1** | **0** | **0** | **4** |

### Multiplikation

Bei Präfixschreibweise:

A screenshot of a computer

Description automatically generated with medium confidenceBei der binären Multiplikation gelten die folgenden Regeln:

Danach wird jeweils der ganze erste Multiplikator mit einer Stelle (angefangen von rechts) des zweiten Multiplikators multipliziert. Nach jeder Stelle wird eins nach links eingerückt. Danach werden die Zahlen normal addiert.

### Division

Bei Präfixschreibweise:

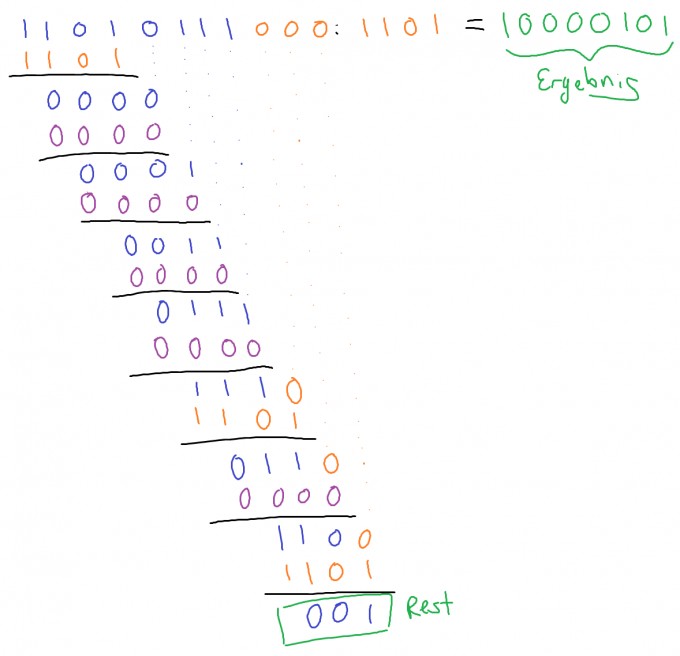
Bei der binären Division gelten folgende Regeln (Division durch 0 nicht definiert):

Danach probiert man zuerst, die Stelle ganz links der zu teilenden Zahl durch den ganzen Teiler zu teilen. Solange es nicht geht, nimmt man noch eine Stelle eins weiter rechts der zu teilenden Zahl dazu. Dann schreibt man beim Ergebnis auf, wie oft der Teiler in die Stellen reinpasst (beim ersten Durchlauf immer ).

Nun subtrahiert man den Teiler von den Stellen und schreibt noch die nächste Stelle vom Dividenden runter ganz rechts vom Ergebnis. Nun gibt es zwei Möglichkeiten:

* Der Teiler passt ins erweiterte Subtraktions-Ergebnis. Es kommt eine 1 ins Ergebnis und der Teiler wird erneut subtrahiert.
* Der Teiler passt nicht ins erweiterte Subtraktions-Ergebnis. Es kommt eine 0 ins Ergebnis und 0 subtrahiert.

Das wird wiederholt, bis ganz unten eine Zahl kleiner als der Teiler befindet (Rest), es kann auch 0 sein.





## Signed Integers

Dabei wird das MSB als Vorzeichen interpretiert, 0 ist positiv und 1 ist negativ.

Der Wertebereich reicht von bis , wobei die Anzahl Bits ist.

Die negativen Zahlen werden dabei als Zweierkomplement gespeichert.

Der Wert einer vorzeichenbehafteten, negativen Binärzahl mit n Bit entspricht der Differenz aus vorzeichenlosem Wert und .

Schreibweise Zweierkomplement:

### Umwandlung negative Dezimalzahl ins Zweierkomplement (Inversionsverfahren)

Beispiel mit Zahl :

1. In die positive Binärzahl umwandeln (muss als MSB haben):
2. Alle Bits invertieren:
3. Eins addieren:

### Umwandlung Zweierkomplement in negative Dezimalzahl (Inversionsverfahren)

Bespiel mit :

1. Alle Bits invertieren:
2. Eins addieren:
3. Zahl ablesen und minus davorsetzen:

# Boolsche Logik

## Arität von Funktionen

A picture containing text, screenshot, font

Description automatically generated

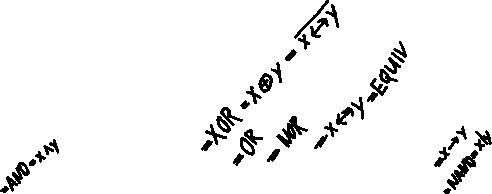
## Unäre Funktionen

Es gibt genau 4 unäre Funktionen, wobei nur die Identität (id) und Negation (not) von der Eingabe abhängen.

A picture containing font, number, screenshot, line

Description automatically generated

## Binäre Funktionen



Es gibt genau 16 binäre Funktionen.

A picture containing font

Description automatically generated

NAND sind die Grundbausteine der Computertechnik, da sie technisch leicht als Transistoren zu realisieren sind. Alle Basisoperationen können damit dargestellt werden.

De Morgan Law:

## Terme

A picture containing text, font, screenshot, algebra

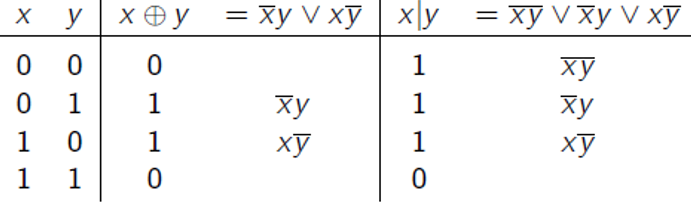
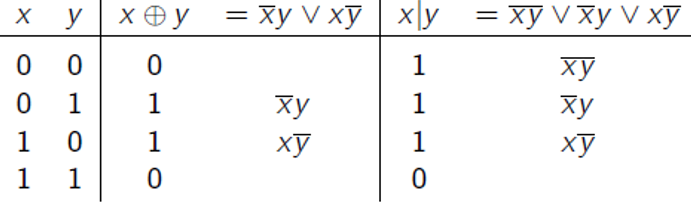
Description automatically generated

## Disjunktive Normalformen

Disjunktive Normalform (DNF): Disjunktion von Konjunktionstermen, z.B.

Kanonische disjunktive Normalform (KDNF): Disjunktion von Mintermen, z.B.

Die KDNF kann schnell aus einer Wahrheitstabelle abgeleitet werden:



Die KDNF kann zu einer DNF vereinfacht werden:

# ASCII

Alle Zeichen eines Texts stammen aus einer endlichen Menge Z, dem Zeichensatz. Jedem Zeichen kann eindeutig eine natürliche Zahl zugeordnet werden (Encoding). Sender und Empfänger müssen das Encoding miteinander vereinbaren, damit sie mit 0 und 1 Text übertragen können. Ein verbreitetes Encoding ist der «American Standard Code for Information Interchange» (ASCII).

A picture containing text, screenshot

Description automatically generated

Bekannte, nicht obsolete Steuerzeichen:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| \0 | Null | Ende des Texts | \n | Line Feed | Cursor eine Zeile nach unten |
| \b | Backspace |  | \r | Carriage Return | Cursor an Anfang der Zeile |
| \t | Tabulator |  |  |  |  |

# Fixkommazahlen

Bei der Fixkommazahl wird an einer beliebigen Stelle einer Binärzahl ein Komma gesetzt. Viele Dezimalkommazahlen können nicht exakt dargestellt werden.

Beispiel für :

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Stelle |  |  |  |  |  |  |  |  |
| Zweierpotenz |  |  |  |  | . |  |  |  |
| Dezimal |  |  |  |  | . |  |  |  |
| Binäre Fixkommazahl |  |  |  |  | . |  |  |  |

Bezogen auf unsigned Fixkommazahl mit 3 Bit vor dem Komma und 5 Bit nach dem Komma:

* Kleinste Zahl > 0:
* Grösste Zahl:

Gleich wie bei ganzen Binärzahlen sind: Addition, Subtraktion, Zweierkomplement und Shift

## Multiplikation

Bei der Multiplikation verschiebt sich das Komma, das Ergebnis hat so viele Nachkommastellen wie die beiden Multiplikatoren zusammen.

A picture containing text, handwriting, blackboard, font

Description automatically generated

Bei der Division wird ebenfalls das Komma verschoben.

# Gleitkommazahlen

Gleitkommazahlen werden immer als normalisierte Zahlen geschrieben, das heisst mit genau einer 1 vor dem Komma. Beispiel:

Die führende 1 kann beim Codieren weggelassen werden, sie ist ein Hidden Bit. Somit hat man eine Stelle mehr für die Präzision.



Wert einer Float-Zahl an diesem Beispiel:



Das Bias beträgt bei 32-bit Float und bei 64-bit Double

A picture containing text, font, screenshot, white

Description automatically generated

* Grösste Float-Zahl:
* Kleinste Float-Zahl > 0:

## Dezimalzahl als Float darstellen

Beispiel: als Single-Float

1. Umwandlung der Zahl vor dem Komma zur Binärzahl:
2. Umwandlung der Zahl hinter dem Komma zur Binärzahl:

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

1. Nun fügen wir die beiden Binärzahlen zu einer Fixkommazahl zusammen, ausserdem haben wir ein Muster entdeckt:
2. Nun wir das Komma so weit nach links geschoben, bis genau eine links vom Komma steht  
   Die Anzahl Stellen nach rechts werden als Exponent von 2 malgerechnet
3. Bestimmung von mithilfe des Vorzeichens:
4. Bestimmung des Exponenten : Exponent von Schritt 4 plus 127:
5. Bestimmung der Mantisse : Die Bits nach dem Komma von Schritt 4:
6. Ergebnis:

## Addition von Floats

Beispiel Addition von

1. Dezimalzahlen im Float-Format aufschreiben
2. Das Hidden Bit (immer 1 ganz links der Mantisse einfügen)
3. Den grösseren Exponenten in beide Floats schreiben
4. Die Differenz der beiden Exponenten berechnen
5. Beim Float mit dem ursprünglich kleineren Exponenten so viele Nullen einfügen, wie bei Schritt 4 berechnet wurden. Die gleiche Anzahl wie eingefügt wurde fliegt hinten raus.
6. Nun wird die Mantisse normal binär addiert. Jedoch nur die vordersten 23 Bit. Falls das Carry Bit 1 ist nach der Mantisse, wird der Exponent um 1 erhöht. Das Vorzeichen bleibt.



## Runden

* Round to nearest: es gibt zwei Varianten für .1
  + Ties to even: auf gerade Zahl, z.B. 0.1 auf 0 und 1.1 auf 10.0
  + Ties away from zero: immer auf die grössere Zahl
* Round toward 0
* Round toward
* Round toward

Rundungsfehler bei kleinster Zahl grösser 1:

A picture containing text, font, handwriting, line

Description automatically generated

Maschinen-Epsilon:

A picture containing text, font, screenshot, white

Description automatically generated