

# Zusammenfassung der Wissenstruktur Computertechnik

Bodun Du

December 5, 2020

# Contents

|          |  |          |
|----------|--|----------|
| <b>1</b> | <b>Codierung</b>                                     | <b>3</b> |
| <b>2</b> | <b>Binärsystem</b>                                   | <b>4</b> |
| <b>3</b> | <b>Codierungs Verwirklichung auf dem Binärsystem</b> | <b>5</b> |
| 3.1      | Textcodierung . . . . .                              | 5        |
| 3.2      | Zahlencodierung . . . . .                            | 6        |
| 3.2.1    | Verwirklichung des Vorzeichens . . . . .             | 6        |
| 3.2.2    | Verwirklichung des Kommas . . . . .                  | 7        |

# 1 Codierung

Mapping der implizierten Elemente eines Zahlensystems durch konvention Entsprechungsregel, um die natursprachliche bzw. semantische Elemente zu vertreten, sodass die menschliche Informationen in bestimmter Form verarbeitet und gespeichert werden können.

## **Basis( $b$ ):**

Informationsvarität/Mächtigkeit eines einzelnen Zeichens.

## **Datenbreite( $n$ ):**

Zusammensetzung mehrerer Datenstellen.

Ein  $n$ -Stellen-Informationsstück als Menge hat die Informationsmächtigkeit (Anzahl verschiedener Kombinationsmöglichkeiten)

$$b^n$$

## **Überlauf**

Ein Informationsstück mit höherer Informationsmächtigkeit kann alle Elemente aus dem Informationsstück, der niedrigere Informationsmächtigkeit hat, vollständig implizieren.

Ist ein Element nicht in den Zielinformationsstück zu implizieren, so hat man einen Überlauf.

**Carryout:** Erweiterung des Zielvolums

## **Wertigkeit**

Der Wert( $v$ ) eines Informationsstück wird durch arithmetische Zählung der verschiedenen Zifferkombinationen definiert.

So gilt die Wertumrechnung ins dezimale System:<sup>1</sup>

$$v = \sum_{i=0}^{j=n-1} a_i \cdot b^i$$

---

<sup>1</sup> $a_i$ : Zifferinhalt an der Stelle  $i$

## 2 Binärsystem

In der Computertechnik wird hauptsächlich 2-Pegels-Hardware-Bauelemente benutzt, wird also  $b=2$  hier intensiv betrachtet.

**Bit:** eine Stelle im Binärsystem, entweder 0 oder 1

**Byte:** 1 Byte = 8 Bits

### 3 Codierungs Verwirklichung auf dem Binärsystem

Konventionell gibt es zwei Stufen der Codierung vom Binärdatenstück, die verschiedenen Verwendungszweck entsprechen. Der Anfang eines Informationstücks wird durch Prefix 0x gekennzeichnet.

#### Ziffer

|         |                          |                       |
|---------|--------------------------|-----------------------|
| $n = 3$ | oktale Darstellung       | entspricht $b=8;n=1$  |
| $n = 4$ | hexadezimale Darstellung | entspricht $b=16;n=1$ |

Weil die oktale Darstellung die dezimale Daten gleicher Breite nicht implizieren kann, wird heute standradmäßig hexadezimale Darstellung in der Codierung verwendet. Dazu werden die Werte 10 bis 15 durch A bis F repräsentiert.

#### Datenwort

Durch Kombination der Ziffer lässt sich die Mächtigkeit eines Informationstücks erweitern.

|                 |                |
|-----------------|----------------|
| $n = 3 \cdot 8$ | 24-Bits-System |
| $n = 4 \cdot 2$ | 8-Bits-System  |
| $n = 4 \cdot 4$ | 16-Bits-System |
| $n = 4 \cdot 8$ | 32-Bits-System |
| $\vdots$        | $\vdots$       |

Ein Datenwort kann sowohl mit übersetzter Ziffercodierung als auch mit Bits dargestellt werden.

#### 3.1 Textcodierung

Um Textinhalt, der enorme verschiedene Einheitszeichen hat, zu codieren, wird oft lange Datenwortbreite verwendet. Da wird das Mapping auf der Zifferbreite von 0 bis 9; A bis Z nach arithmetischer Folge geordnet.

#### ASCII-Standard(American Standards Code for Information Interchange):

Mapping des 7-Bits-Systems (mit Logik der Schreibmaschine) für englische Buchstaben mit Steuersymbolen

#### EASCII:

8-Bits-Erweiterung vom ASCII-Standard, dass europäisch-spezifische Buchstaben unterstützt werden.

### **UCS(Universal Charcter Set)/ISO 10646/ICE 10646**

Sammlung aller bekannten Zeichen mit Codierung(Nummerierung von U+0000 bis vorgesehen U+10FFFF)

#### **Unicode:**

heute als synchrone Zeichensammlung mit UCS, enthält aber zusätzlich noch sprachrelevante Algorithmen.

#### **UTF-8:**

Eine Codierng-Verwirklichung vom UCS bzw. Unicode auf 8- bis 32-Bits variierter Wortbreite, mit Kompatität zum EASCII

#### **UTF-16:**

Eine Codierng-Verwirklichung vom UCS bzw. Unicode auf 16- und 32-Bits Wortbreite

#### **UTF-32:**

Eine Codierng-Verwirklichung vom UCS bzw. Unicode auf 32-Bits fester Wortbreite

## **3.2 Zahlencodierung**

Bei Zahlen wird die hexadezimale Darstellung unmittelbar als Recheneinheit genommen.

Mit arithmetischen Zählung definierte Wertigkeit werden positive natürliche Zahlen impliziert. Will man die Zahlen mit Vorzeichen und Komma Berechnen, so muss die Wert-Codierung Modifiziert werden.

### **3.2.1 Verwirklichung des Vorzeichens**

#### **Separation von Zahl und dem Vorzeichen**

7. bis 16. Elemente werden von -0 bis -7 zugeordnet

#### **Einer-Kompliment**

7. bis 16. Elemente werden von -7 bis -0 zugeordnet

#### **Zweier-Kompliment**

7. bis 16. Elemente werden von -8 bis -1 zugeordnet

### 3.2.2 Verwirklichung des Kommas

#### Festkommazahl

Einführung vom Parameter  $r$ , sodass

$$v = b^r \cdot \sum_{i=0}^{j=n-1} a_i \cdot b^i$$

#### Gleitkommazahl

Darstellung der Zahlen durch wissenschaftliche Arithmetik mit Speicherung von  $s, e$  und  $f$ .<sup>2</sup>

Standardfall:

$$v = (-1^s \cdot 1, f \cdot 2^{e-K})$$

oder erweitert (für z.B. 0,  $\infty$  oder NaN<sup>3</sup>):

$$v = (-1^s \cdot 0, f \cdot 2^{1-K})$$

Da im 32-Bits-Informationsstück:  $K=127$ ;

im 64-Bits-Informationsstück (mit doppelter Genauigkeit):  $K=1023$

---

<sup>2</sup>Speicherungsverhältnis  $s/e/f$  im 32-Bits-Informationsstück: 1/8/23; im 64-Bits-  
Informationsstück: 1/11/52

<sup>3</sup>Not a Number