4. Digitale Datendarstellung

- ❑ Daten und Codierung
- ❑ Textcodierung
- ❑ Codierung natürlicher Zahlen
  - Stellenwertsysteme
  - Konvertierung
  - Elementare Rechenoperationen
- ❑ Codierung ganzer Zahlen
  - Komplementdarstellung
- ❑ Codierung rationaler Zahlen
  - Festpunktdarstellung
  - Gleitpunktdarstellung

Daten

- Daten repräsentieren Informationen (eines anwendungsbezogenen Ausschnitts) der realen Welt ("Miniwelt") in der Rechenanlage

- Sie stellen fundamentale Objekte dar, die erfaßt, gespeichert, verarbeitet, ausgegeben (angezeigt, gedruckt), übertragen und gelöscht werden können
  - Verarbeitung der Daten durch Programme modelliert Handlungen in der realen Welt

- Einfache Datenstrukturen
  - numerische Daten: natürliche Zahlen, ganze Zahlen, rationale Zahlen etc.
  - nicht-numerische Daten

- Zusammengesetzte Datenstrukturen
  - Listen, Bäume, . . .
  - Sammlungen von Datensätzen (Dateien)

Codierung

- Darstellung der Daten erfolgt mit *Symbolen* (Zeichen) eines bestimmten *Alphabets* (endliche Zeichenmenge)

{a, b, c, ..., A, B, C, ...}	Alphabet der Buchstaben
{rot, gelb, grün}	Alphabet der Verkehrsampelsignale
{Frühling, Sommer, Herbst, Winter}	Alphabet der Jahreszeiten
{0, 1, 2, ..9}	Alphabet der Dezimalziffern
{0, 1}	Alphabet der Dualziffern
{0, 1, 2, 3, 4, 5, 6, 7}	Alphabet der Oktalziffern
{0, 1, ... 9, A, B, C, D, E, F}	Alphabet der Sedezimalziffern (Hexadezimalziffern)

- *Codierung:* injektive Abbildung zwischen zwei Alphabeten (beschreibbar durch eine Code-Tabelle)

- Binärcodierung: Codierung durch Binärzeichen (Bits)
  - bei mehr als 2 Objekten sind mehrere Binärzeichen (Bitfolgen) zur Codierung erforderlich

  - mit Folgen aus n Binärzeichen können 2^n Symbole codiert werden

Binärcodierung

- ❑ Computer speichern alle Daten und Programme als Bitfolgen

- ❑ einfache Darstellungsmöglichkeit durch zweiwertige physikalische Größen
  - Bsp.: Spannung/keine Spannung, positive/negative Magnetisierung, Loch/kein Loch, heller/dunkler Bildpunkt etc.
  - geringe Störempfindlichkeit
  - Speicherbarkeit in integrierten Schaltungen, Magnetbändern, Platten, Lochkarten, etc.

- ❑ Alle zählbaren und meßbaren Daten können binärcodiert und damit durch Computer verarbeitet werden
  - Texte, Zahlen, logische Werte, Bilder, Sprache, Musik etc.

- ❑ Für analoge Daten (stetig veränderliche physikalische Größen) ist zunächst eine *Diskretisierung* (Rasterung) erforderlich, bevor eine Binärcodierung (*Digitalisierung*) erfolgt

- ❑ Digitalisierung kann durch geeignete Wahl der Bitanzahl beliebig genau gemacht werden

Darstellung von Texten

- Binärcodierung von Groß- und Kleinbuchstaben, Ziffern, Satzzeichen sowie einiger nicht-druckbaren Sonderzeichen (Carriage Return, Tabulatorzeichen, etc.)

- Verbreitetste Festlegung: ASCII (American Standard Code for Information Interchange)
  - Normierung umfaßt 7 Bit (128 Zeichen)
  - erweiterte ASCII-Codes (8 Bit) umfassen sprachspezifische Sonderzeichen wie Umlaute und Grafiksymbole (nicht einheitlich)

- Auszug aus der Code-Tabelle

Zeichen	ASCII (dezimal)	ASCII (dual)
0	48	00110000
1	49	00110001
9	57	00111001
A	65	01000001
B	66	01000010
Z	90	01011010
[	91	01011011
a	97	01100001
b	98	01100010
z	122	01111010

  - die Ziffern 0 bis 9 stehen in der natürlichen Reihenfolge
  - Klein- und Großbuchstaben sind in alphabetischer Reihenfolge durchnummeriert, jedoch in getrennten Bereichen (ihre Codierung unterscheidet sich nur im 3. Bit von links)

Codierung von natürlichen Zahlen

- Stellenwert-Codierung natürlicher Zahlen:

  Zu jeder natürlichen Zahl z und jeder Basis b >= 2 gibt es genau eine natürliche Zahl n > 0 (Stellenanzahl) und eine Ziffernfolge $x_{n-1}\ x_{n-2} \ldots x_1\ x_0$ mit

$$z = \sum_{i=0}^{n-1} x_i\, b^i = x_{n-1}\ b^{n-1} + \ldots + x_1\ b + x_0$$

- Beispiele:

  $321_{10} =$

  $1001110_2 =$

  $7E4_{16} =$

- Computerspeicherung natürlicher Zahlen in Dualdarstellung wesentlich kompakter als in Dezimaldarstellung

  Wortlänge 16 Bit unterstützt Zahlenbereich 0 bis 2^{16}-1 (=65535)

Umrechnung zwischen Zahlsystemen (Konvertierung)

❑ Horner -Algorithmus zur Berechnung von Polynomwerten

■ Polynom k-ten Grades ($k \in N$) mit reellem Koeffizienten $\mathbf{a_i}$:

$$p(x) = a_k x^k + a_{k-1} x^{k-1} + \ldots + a_1 x + a_0$$

■ sukzessives Ausklammern von x :

$$\begin{aligned} p(x) &= (a_k x^{k-1} + a_{k-1} x^{k-2} + \ldots + a_1)x + a_0 \\ &\;\vdots \\ &= (\ldots(a_k x + a_{k-1})x + a_{k-2})x + \ldots + a_1)x + a_0 \end{aligned}$$

❑ Algorithmus (HORNER-Schema)

$y_0 := a_k$
FOR i := 1 **TO** k **DO**
$y_i := y_{i-1} \cdot x + a_{k-i}$

END ; (* FOR *)
(* Ergebnis : $y_k = p(x)$ *)

❑ Konvertierung

A) beliebiges b-adisches Zahlsystem → Dezimalsystem

B) Dezimalsystem → beliebiges b-adisches Zahlsystem

C) zwischen beliebigen b-adischen Zahlsystemen

Fall A) 3 Teilfälle: Zahl $\mathbf{z \in N}$

$\mathbf{z \in P}$, $\mathbf{0 < z < 1}$

$\mathbf{z \in P}$, $\mathbf{1 < z}$

Voraussetzung: b-adische Zahldarstellung für $z \in N$

$c_b(z) = a_k a_{k-1} \ldots a_0$ ist endlich!

(anlog für $z \in P$!)

❑ z natürlich (1.Teilfall) $\mathbf{z \in N}$

$$z = \sum_{i=0}^{k} a_i b^i$$

$$=(\ldots(a_k b + a_{k-1})b + a_{k-2})b + \ldots + a_1)b + a_0$$

Berechnung von $c_{10}(z)$: gemäß HORNER-Schema

Klammern von innen nach außen ausmultiplizieren,
d.h. folgendes **Rechenschema :**

Koeff. von $c_b(z)$	0
a_k	y_0
a_{k-1}	y_1
.	.
.	.
.	.
a_1	y_{k-1}
a_0	$y_k = c_{10}(z)$

* b + „linker Spaltenwert“

alle Rechnungen im Dezimalsystem ausführen, vorher Koeffizienten a_i ins Dezimalsystem umwandeln!

❑ z echter endlicher Bruch (2.Teilfall) **$z \in P, \quad 0 < z < 1$**

$c_b(z) = 0.\, a_{-1}\, a_{-2} \ldots a_{-m}$, $\quad z = \sum_{i=1}^{m} a_{-i}\, b^{-i}$

$z = b^{-1}(a_{-1} + b^{-1}(a_{-2} + b^{-1}(a_{-3} + \ldots + b^{-1}(a_{-m+1} + b^{-1}a_{-m}) \ldots)$

Rechenschema:

Koeff. von $c_b(z)$	0
a_{-m}	z_0
a_{-m+1}	z_1
a_{-m+2}	z_2
.	.
.	.
.	.
a_{-1}	z_{m-1}
0	$z_m = c_{10}(z)$

$* \, b^{-1}$ + „linker Spaltenwert“

$z_i = a_{-m+i} + b^{-1}(\ldots b^{-1}(a_{-m+1} + b^{-1}a_{-m}) \ldots)$

mit

$z_i/b < 1$ für $0 \leq i \leq m-1$

❑ z unechter endlicher Bruch (3.Teilfall) **$z \in P, \quad 1 < z$**

$$z = \bar{z} + \hat{z}$$

$$c_b(z) = \underbrace{a_k a_{k-1} \ldots a_1 a_0}_{\bar{z} \in N} \,.\, \underbrace{a_{-1} a_{-2} \ldots a_{-m}}_{\hat{z} \in P}$$

$\bar{z} \in N \qquad \hat{z} \in P, \qquad 0 < \hat{z} < 1$

Fall B) 3 Teilfälle: Zahl $\mathbf{z \in N}$

$\mathbf{z \in P}$, $\mathbf{0 < z < 1}$

$\mathbf{z \in P}$, $\mathbf{1 < z}$

- ❑ z natürlich (1.Teilfall) $\mathbf{z \in N}$

$$c_{10}(z) = d_k\, d_{k-1} \ldots d_0$$

$$z = \sum_{i=0}^{k} d_i \cdot 10^i$$

gesucht:

$$z = \sum_{i=0}^{k} a_i \cdot b^i$$

$$= (\ldots(a_k\, b + a_{k-1})b + a_{k-2})b + \ldots + a_1)b + a_0$$

Ziffern a_i: beginnend bei a_0 als Rest der sukzessiven Division von z durch b

: b

$c_{10}(z)$	Rest
y_1	a_0
y_2	a_1
.	.
.	.
.	.
y_k	a_{k-1}
0	a_k

$y_i = (\ldots(a_k b + a_{k-1})\, b + \ldots + a_{i+1})\, b + a_i)$

$y_i \in N$

$c_b(z) = a_k a_{k-1} \ldots a_1 a_0$

❑ z echter endlicher Bruch (2.Teilfall) **$z \in P, \quad 0 < z < 1$**

$c_{10}(z) \quad = 0.\, d_{-1}\, d_{-2} \ldots d_{-j}$

$$z = \sum_{i=1}^{j} d_{-i} \cdot 10^{-i}$$

$$z = \sum_{i=1}^{m} a_{-i} \cdot b^{-i} \qquad \text{(gesuchte Darstellung)}$$

$$z = b^{-1}\,(a_{-1} + b^{-1}(a_{-2} + \ldots + b^{-1}(\, a_{-m+1} + b^{-1}a_{-m}) \ldots)$$

→ Koeffizienten $a_{-1}, \ldots, a_{-m}$ durch fortlaufende Multiplikation der Dezimaldarstellung von z mit b

Rechenschema:

ganzzahliger Anteil	$c_{10}(z)$	
a_{-1}	y_1	$* b$
a_{-2}	y_2	$y_i = b^{-1}(a_{-i-1} + \ldots + b^{-1}(a_{-m+1} + b^{-1}a_{-m})\ldots)$
.	.	
.	.	$a_{-i-1} \in N \quad , \quad a_{-i-1} < b$
.	.	
a_{-m}	y_m	$0 \le y_i < 1 \quad \text{für} \quad 1 \le i \le m$

↓

$c_b(z) = 0.\, a_{-1}a_{-2} \ldots a_{-m}$

ist die gesuchte Darstellung

❑ z unechter endlicher Bruch (3.Teilfall) **$z \in P, \quad 1 < z$**

→ analog zu Fall A

Rechnen mit natürlichen Dualzahlen

- Addition

  0 + 0 = 0
  1 + 0 = 1
  0 + 1 = 1
  1 + 1 = 10 (0 Übertrag 1)
  Addition mehrstelliger Dualzahlen analog zum Dezimalsystem

- Multiplikation

  0 • 0 = 0
  0 • 1 = 0
  1 • 0 = 0
  1 • 1 = 1

  Multiplikation mehrstelliger Dualzahlen analog zum Dezimalsystem

- Subtraktion kann auf Addition negativer Zahlen, Division auf Folge von Additionen, Multiplikationen und Vergleiche zurückgeführt werden

Darstellung von ganzen Zahlen

- Standard-Datentyp INTEGER

- Unterscheidung zwischen positiven und negativen Zahlen durch Vorzeichenbit (z.B. 0 = +, 1 = -)

  - Zahlenbereich für n Bit: $-2^{n-1}+1$ bis $+2^{n-1}-1$
  - Beispiel für n=4 (-7 bis +7):

0000 = +0	0100 = +4	1000 = -0	1100 = -4
0001 = +1	0101 = +5	1001 = -1	1101 = -5
0010 = +2	0110 = +6	1010 = -2	1110 = -6
0011 = +3	0111 = +7	1011 = -3	1111 = -7

Nachteile: zwei Repräsentationen für 0; komplizierte Rechenoperationen

- Alternative: (Zweier-) Komplementdarstellung

Komplementdarstellung

- Negative Zahl -z wird durch Zweierkomplement von z repräsentiert, nämlich $2^n - z$
(entspricht Stellenkomplement der Dualzahl + 1)

- Binäre Ziffernfolge $x_{n-1}\ x_{n-2}\ ..\ x_1\ x_0$ repräsentiert die Zahl

$$x_{n-1} \bullet (-2^{n-1}) + x_{n-2} \bullet 2^{n-2} + ... + x_1 \bullet 2 + x_0$$

- Darstellbarer Zahlenbereich: $-\ 2^{n-1}$ bis $+2^{n-1}\ -1$
  - n=4: Bereich -8 bis +7
  - n=32: Bereich -2 147 483 648 bis +2 147 483 647

1000 = -8 1001 = -7 1010 = -6 1011 = -5	1100 = -4 1101 = -3 1110 = -2 1111 = -1	0000 = 0 0001 = 1 0010 = 2 0011 = 3	0100 = 4 0101 = 5 0110 = 6 0111 = 7

- Vorzeichen wird weiterhin durch erstes Bit bestimmt

- Addition negativer Dualzahlen kann wie für natürliche Zahlen erfolgen

Darstellung von rationalen Zahlen

- Da nur 2^n Werte exakt darstellbar sind, müssen alle anderen der unendlich vielen rationalen Zahlen durch einen benachbarten Wert approximiert werden

- Jede nichtnegative rationale Zahl z kann eindeutig in ganzzahligen Anteil z1 >= 0 und

  gebrochenen Anteil   0 <= z2 < 1 zerlegt werden mit

  $$z = z1 + z2$$

- *Festpunktdarstellung* für z im Stellenwertsystem mit n (=k+m) Stellen und Basis b

  $x_{k-1}\ x_{k-2} \ldots x_1\ x_0 \cdot y_1\ y_2 \ldots y_m$ bedeutet

  $$z = \sum_{i=0}^{k-1} x_i \times b^i + \sum_{i=1}^{m} y_i \times b^{-i}$$

- Konversionen analog zu ganzen Zahlen

  $0.110101_2 =$

  $0.5_{10} =$

  Einfache Dezimalbrüche ergeben jedoch oft periodische Dualbrüche:

  $0.1_{10} = 0.0\ 0011\ 0011 \ldots_2$

Darstellung logischer Werte

- 2 Wahrheitswerte (Boolesche Werte):
  TRUE (wahr) und
  FALSE (falsch)
- Codierung durch 1 Bit möglich
- Boolesche Werte dienen zur Festlegung der Wahrheitswerte von Aussagen (z.B. Beziehungen)

  Relation 5 < 7 hat Wahrheitswert

  Relation 7 < 5 hat Wahrheitswert

  Aussage "Der Mars ist bewohnt" hat Wahrheitswert

  Relation x = 0 hat Wahrheitswert
- Operationen der Aussagelogik anwendbar: Negation (NOT), Konjunktion (AND), Disjunktion (OR), Exklusives Oder (XOR), etc.

NOT	
T	
F	

AND	T	F
T		
F		

OR	T	F
T		
F		

XOR	T	F
T		
F		

Beobachtung

- Da sämtliche Daten und Programme (Befehle) binärcodiert werden, kann eine bestimmte Bitfolge unterschiedliche Dinge bedeuten

  Bitmuster: 01100001 01100010 01100011 01100100

ASCII-Zeichen	abcd
ganze Zahl	1 633 837 924
Gleitpunktzahl	$0.7608150709 \ldots \bullet 2^{-30}$
Befehl	LOAD 6447972

  • • •

- Interpretation v.a. abhängig vom Ort der Verarbeitung: Steuerwerk, Rechenwerk für ganze Zahlen, Rechenwerk für Gleitpunktzahlen, ...

- Programmierer mitverantwortlich für Bereitstellung der "richtigen" Eingabedaten