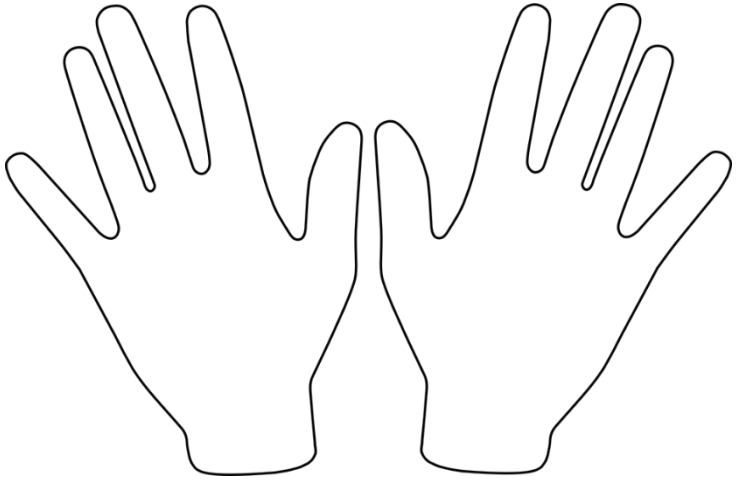
Zahlensysteme

Menschliches System: Dezimalsystem („Zehnersystem“)



- Mensch verwendet i. d. R. das Zehnersystem
  - Positionssystem mit Basis 10

- Darstellung einer Zahl:

$$n = \sum_{i=0}^{N} a_i \cdot 10^i$$

$$= a_n \cdot 10^n + \cdots + a_3 \cdot 1000 + a_2 \cdot 100 + a_1 \cdot 10 + a_0 \cdot 1$$

Aus der (Grund-)Schule:

Zehntausender, Tausender, Hunderter, Zehner, Einer

Positionssystem

- Ein Positionssystem mit **Basis** B ist ein Zahlensystem, in dem die Zahl n in Potenzen von B zerlegt wird

- Eine natürliche Zahl wird durch folgende Summe dargestellt:

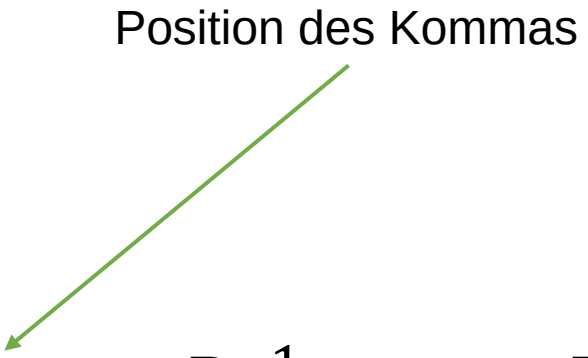
$$n = \sum_{i=0}^{N-1} b_i \cdot B^i$$

- Anmerkung:
  - $N :=$ Anzahl der Stellen
  - $b_i :=$ **Ziffern** der Zahl mit $b_i \in \mathbb{N}_0, 0 \leq b_i < B$

Erweiterung: Kommazahlen

- Das Komma einer Kommazahl trennt den ganzzahligen Teil der Zahl vom gebrochenen Teil „Nachkommastelle(n)“

- Darstellung als Summe:



$$x = \sum_{i=-m}^{n-1} b_i \cdot B^i = \cdots + x_0 + x_{-1} \cdot B^{-1} + x_{-2} \cdot B^{-2} + \cdots$$

- n und m: Anzahl der Stellen vor bzw. nach dem Komma

Beispiele: Kommazahlen im Dezimalsystem

$$375 = 300 + 70 + 5 = 3 \cdot 10^2 + 7 \cdot 10^1 + 5 \cdot 10^0$$

$$62.5 = 60 + 2 + 5 \cdot \frac{1}{10} = 6 \cdot 10 + 2 + 5 \cdot 10^{-1}$$

$$5.379 = 5 + 3 \cdot \frac{1}{10} + 7 \cdot \frac{1}{10^2} + 9 \cdot \frac{1}{10^3} = 5 + 3 \cdot 10^{-1} + 7 \cdot 10^{-2} + 9 \cdot 10^{-3}$$

Das Binärsystem

- Problem: Zehn verschiedene Zustände lassen sich technisch nur sehr schwer realisieren
- Einfacher: nur zwei Zustände (z.B. Strom ↔ kein Strom)

Typ	Darstellung 0	Darstellung 1
El. Ladung	Ungeladen	Geladen
El. Spannung	0V	5V
Magnetisierung	Nicht Magnetisiert	Magnetisiert

- Daher arbeiten Computer (*hardwaretechnisch*) nur mit zwei Zuständen, die meist als `0` und `1` („**Bits**“) dargestellt werden

- Bit = Einzelne Binärstelle (**BI**nary Digi**T**)

Das Binärsystem: Definition

- Das Binärsystem ist ein Positionssystem mit der Basis $B = 2$ und den beiden Ziffern $b = \{0, 1\}$

- Natürlich Zahlen werden mit folgender Summe dargestellt:

$$n = \sum_{i=0}^{N-1} b_i \cdot 2^i = b_0 + b_1 \cdot 2 + b_2 \cdot 4 + \cdots + \mathrm{b_{N-1}} \cdot 2^{N-1}$$

- Bei Binärzahlen nutzt man normalerweise eine feste Anzahl von Stellen (:= Bits, meiste Vielfache von 2 wie z. B. 4 oder 8) und gruppiert sie in Gruppen á 4 Bit, z. B. 0010 0011

Anzahl der darstellbaren Werte

- Anzahl der darstellbaren Werte hängt von der Anzahl der Stellen und der Basis ab

# Stellen	Binärsystem		Dezimalsystem	
	Maximum	# Werte	Maximum	# Werte
1	1	2	9	10
2	3	4	99	100
3	7	8	999	1000
…				
7	127	128	999 999	10^7
8	255	256	9 999 999	10^8

- Allgemeine Formel: $\#W = B^N$

Aufgabe: Binärsystem

- Welche Zahlen werden hier im Binärsystem dargestellt?

1011_2

$1011\ 0101_2$

$1111\ 1111_2$

Lösung: Binärsystem

- Welche Zahlen werden hier im Binärsystem dargestellt?

$$1011_2 = 1 \cdot 8 + 0 \cdot 4 + 1 \cdot 2 + 1 \cdot 1 = 11$$

$$1011\ 0101_2 = 128 + 32 + 16 + 4 + 1 = 181$$

$$1111\ 1111_2 = 128 + 64 + 32 + 16 + 8 + 4 + 2 + 1 = 255$$

Andere Zahlensysteme

- Um ein anderes Zahlensystem zu erhalten, tauschen wir nur die Basis B aus
- Damit ändern sich automatisch auch die verfügbaren Ziffern b_i
- Auswahl an Beispielen:

Basis	Name des Systems	Ziffern
2	Dual- / Binärsystem	0, 1
3	Ternärsystem	0, 1, 2
4	Quatenärsystem	0, 1, 2, 3
8	Oktalsystem	0, 1, …, 6, 7
16	Hexadezimalsystem	0, 1, …, 9, A, B, C, D, E, F
20	Vegesimalsytem	0, 1, …, I ($:= 18$), J ($:= 19$)

Andere Zahlensysteme

- Durch die Verwendung verschiedener Basen kann ein und die selbe Ziffernkombination für verschiedene Zahlen stehen
- Es gilt:
  - Gleiche Ziffern, aber andere Bedeutung
  - Gleiche Zahl, aber andere Ziffern
- Beispiele:

Basis	0	1	2	8	10	11	15	16	17
2	0	1	10	1000	1010	1011	1111	1 0000	1 0001
8	0	1	2	10	12	13	17	20	21
10	0	1	2	8	10	11	15	16	17
16	0	1	2	8	A	B	E	10	11

Zahlensysteme in der Informatik

In der Informatik sind vor allem drei Zahlensysteme wichtig:

- Binärsystem ($B = 2$)
  - Grundlegendes System in Rechnern (Strom, kein Strom)
- Oktalsystem ($B = 8$)
  - Verwendung in Unix-Systemen für Verwaltung der Dateizugriffsrechte (3 Bit, jeweils eines für Eigner, Gruppe und Andere)
  - Früher: Darstellung von Datenwörtern mit 24 Bit als 8-stellige Oktalzahl
- Hexadezimalsystem ($B = 16$)
  - Kürzere und menschenlesbarere Darstellung von heute üblichen Datenwörtern mit 32 und 64 Bit als 8- bzw. 16-stellige Hexadezimalzahl

Erkennen der Basis

- Wenn man also nun eine Ziffernfolge wie `1010` oder `1337` sieht, woher weiß man, für welche Zahl diese steht?
  - `1010` könnte als Binärzahl für 10, als Dezimalzahl für 1010 und in anderen Systemen für noch andere Zahlen stehen
  - `1337` kann keine Binärzahl, aber eine Oktalzahl (Wert: 735), eine Dezimalzahl, oder auch eine Hexadezimalzahl (Wert: 4919) sein

- Daher folgende Regel:
  - Die Basis wird tiefgestellt an die Zahl angehängt (Beispiel: 1010_2, 1337_8)
  - Fehlt die Basis, ist die Zahl automatisch eine Dezimalzahl ($1337 = 1337_{10}$)

Umgekehrtes Hornerschema

- Durch dieses Schema lassen sich Dezimalzahlen leicht in anderen Zahlensystemen darstellen

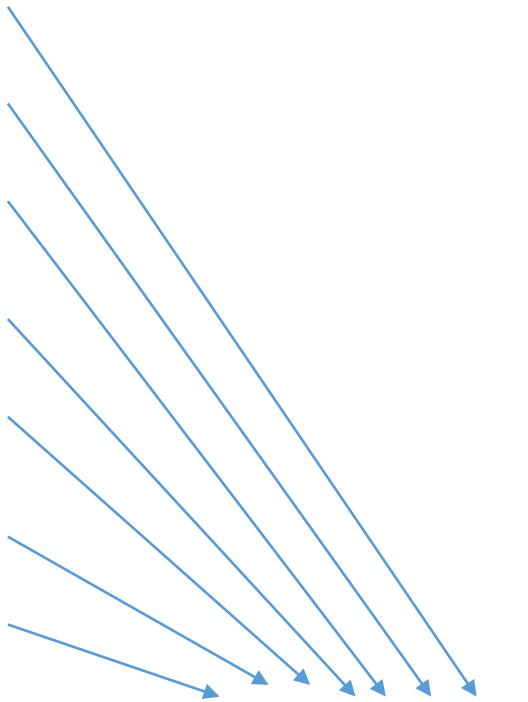
- Dazu wird die Dezimalzahl solange durch die Basis geteilt und der Rest der (Ganzzahl-)Division notiert, bis man auf 0 kommt

- Die Reste ergeben in umgekehrter Reihenfolge die Darstellung der Zahl im anderen System

Beispiele: Umgekehrtes Hornerschema

- Darstellung der Zahl 77 im Binärsystem

```
77 : 2 = 38 Rest 1
38 : 2 = 19 Rest 0
19 : 2 =  9 Rest 1
 9 : 2 =  4 Rest 1
 4 : 2 =  2 Rest 0
 2 : 2 =  1 Rest 0
 1 : 2 =  0 Rest 1
```



$$77_{10} = 1001101_2$$

Beispiele: Umgekehrtes Hornerschema

Wie lautet die Darstellung von 377 im Oktalsystem?

```
377 : 8 = 47 Rest 1
 47 : 8 =  5 Rest 7
  5 : 8 =  0 Rest 5
```

$377_{10} = 571_8$

Wie lautet die Darstellung von 984 im Hexadezimalsystem?

```
984 : 16 = 61 Rest  8
 61 : 16 =  3 Rest 13
  3 : 16 =  0 Rest  3
```

$984_{10} = 3D8_{16}$

Aufgabe: Umgekehrtes Hornerschema

- Wandeln Sie folgende Zahlen jeweils in ihre
  - Binär-,
  - Oktal- und
  - Hexadezimaldarstellung um
- Wie viele Bit brauchen Sie jeweils mindestens?

Dezimal	Binär	Oktal	Hexadezimal
11			
53			
155			
390			

Lösung: Umgekehrtes Hornerschema

- Wandeln Sie folgende Zahlen jeweils in ihre
  - Binär-,
  - Oktal- und
  - Hexadezimaldarstellung um
- Wie viele Bit brauchen Sie jeweils mindestens?

Dezimal	Binär	Oktal	Hexadezimal
11	$0000\ 1011_2$	13_8	B_{16}
53	$0011\ 0101_2$	65_8	35_{16}
155	$1001\ 1011_2$	233_8	$9B_{16}$
255	$1111\ 1111_2$	377_8	FF_{16}

Umwandlung ins Zehnersystem

- Zur „Rückkehr“ in das Dezimalsystem reicht es, den Wert der Zahl zu berechnen
- Beispiel:

$$1010\,0111_2 = 128 + 0 \cdot 64 + 32 + 0 \cdot 16 + 0 \cdot 8 + 4 + 2 + 1 = 167_{10}$$

$$257_8 = 2 \cdot 8^2 + 5 \cdot 8 + 7 \cdot 8^0 = 2 \cdot 64 + 40 + 7 = 128 + 47 = 175_{10}$$

$$EB_{16} = 14 \cdot 16^1 + 11 \cdot 16^0 = 224 + 11 = 253_{10}$$

Umwandlung zwischen Binär, Oktal und Hexadezimal

Zur Umwandlung zwischen Binär und Oktal bzw. Binär und Hexadezimal gibt es einfacherere Regeln:

- Binär & Oktal (Grund: $8 = 2^3$): jeweils 3 Bit der Binärdarstellung werden zu einer Ziffer der Oktaldarstellung
- Binär & Hexad. (Grund: $16 = 2^4$): jeweils 4 Bit der Binärdarstellung werden zu einer Ziffer der Hexadezimaldarstellung

- Beispiele:

$$(0)10\ 101\ 011_2 = 253_8$$

$$1010\ 1011_2 = AB_{16}$$

Rechnen in anderen Zahlensystemen

- Wir wissen, wie man im Dezimalsystem addiert, subtrahiert, multipliziert, …

- Aber:
  - Wie funktionieren diese Rechenarten in anderen Systemen, z. B. im Binärsystem oder im Oktalsystem?
  - Müssen wir immer eine Umwandlung in das Dezimalsystem vornehmen, dort die Rechnung vornehmen, und dann wieder zurück umwandeln?

Addition im Binärsystem

- Für die Addition im Binärsystem reichen prinzipiell folgende Regeln:

S 1	S 2	Summe	Übertrag
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

- Ansonsten wird wie bei der schriftlichen Addition vorgegangen:

```
  0000 0011        0000 0011        1001 0011
+ 0000 0010      + 0001 1011      + 0011 1010
       1                1 1          1 1    1
  ---------        ---------        ---------
  0000 0101        0001 1110        1100 1101
```

Negative Binärzahlen

- Wie wir wissen, kann ein Rechner nur 2 Zustände darstellen: 0 und 1
- Wie stellen wir also negative Zahlen dar?

- Dazu gibt es drei Möglichkeiten:
  - Darstellung des Vorzeichens mit einem zusätzlichen Bit (meist dem ersten Bit)
  - Einerkomplement: alle Bits werden invertiert
  - **Zweierkomplement:** alle Bits werden invertiert und 1 addiert

  ➢In allen Varianten erkennt man negative Zahlen daran, dass das erste Bit eine 1 ist

- Beispiele für Durchführung Einer- (EK) und Zweierkomplement (ZK):

```
 5 = 0101
Inv: 1010 == EK(-5)
 +1: 1011 == ZK(-5)
```

```
73 = 0010 1001
Inv: 1101 0110 == EK(-73)
 +1: 1101 0111 == ZK(-73)
```

Vorteile Zweierkomplement

- Nachteil Vorzeichenbit: Zwei Darstellungen für die 0:
  - 1000_2 = -0_{10} und 0000_2 = $+0_{10}$
- Nachteil Einerkomplement: Ebenfalls zwei Darstellungen für die 0:
  - 1111_2 = -0_{10} und 0000_2 = $+0_{10}$

- Vorteile Zweierkomplement:
  - Nur eine Darstellung für die 0: 0000_2 = $+0_{10}$
    - da Invertierung von 0000_2 + 1 = 1111 + 0001 = 0000
  - Subtraktion kann als Addition durchgeführt werden

Zweierkomplement (ZK)

- Im ZK können die Zahlen -2^{N-1} bis $2^{N-1}-1$ dargestellt werden
- Formel zur Rückrechnung: wenn das erste Bit eine 1 darstellt (→ negative Zahl), subtrahiere von den restlichen Bits die Zahl 2^{N-1}

$$n = -i_{N-1} \cdot 2^{N-1} + \sum_{i=0}^{N-2} b^i \cdot 2^i$$

$0001\,1010_2 = 16 + 8 + 2 = 26_{10}$
$1110\,0110_2 = -128 + 64 + 32 + 4 + 2 = -26_{10}$

$0110\,0110_2 = 64 + 32 + 4 + 2 = 96 + 6 = 102_{10}$
$1001\,1010_2 = -128 + 16 + 8 + 2 = -102_{10}$

Dezimal	Binär, ZK mit $N = 8$
127	0111 1111
4	0000 0100
1	0000 0001
0	0000 0000
−1	1111 1111
−4	1111 1100
−127	1000 0001
−128	1000 0000

Aufgabe: Umwandlung ins Zweierkomplement

Wandeln Sie diese Zahlen in Binärzahlen in ZK mit 8 Bit um:

5

-1

-13

-77

-127

Lösung: Umwandlung ins Zweierkomplement

Wandeln Sie diese Zahlen in Binärzahlen in ZK mit 8 Bit um:

$5 = 0000\ 0101_2$

$Inv_2(+1) = Inv_2(0000\ 0001) = 1111\ 1110_2$

$Inv_2(+13) = Inv_2(0000\ 1101_2) = 1111\ 0010_2$

$-1 = 1111\ 1110_2 + 1_2 = 1111\ 1111_2$

$-13 = 1111\ 0010_2 + 1_2 = 1111\ 0011_2$

$-77 = Inv_2(0100\ 1101_2) + 1_2 = 1011\ 0011_2$

$-127 = Inv_2(0111\ 1111_2) + 1_2 = 1000\ 0001_2$

Subtraktion von Binärzahlen

- Aufgrund der Eigenschaften des Zweierkomplements wird eine Subtraktion im Binärsystem als Addition im ZK durchgeführt:

4_{10}	0100_2
$-\ 5_{10}$	$+\ 1011_2$
---	---
$-\ 1_{10}$	1111_2

77_{10}	$0100\ 1101_2$
$-\ 90_{10}$	$+\ 1010\ 0110_2$
---	---
$-\ 13_{10}$	$1111\ 0011_2$

Kommazahlen

Mehrere Möglichkeiten, Kommazahlen darzustellen:

- Festkomma:
  - festgelegte Anzahl an Stellen vor (n) und nach (m) dem Komma
  - Multiplikation des Bits mit 2^{-k} für die k. Stelle hinter und mit 2^{i} für die i. Stelle vor dem Komma
- Gleitkomma:
  - Darstellung der Zahl als $n = V \cdot M \cdot 2^{E}$
  - V ≔ Vorzeichen, M ≔ Mantisse, E ≔ Exponent
  - Darstellung des Exponenten als Zweierkomplement
  - V, M und E werden im Rechner als ***eine*** Bitfolge mit fester Länge (Stichwort: 32 bzw. 64 Bit OS) dargestellt, Länge von M und E müssen per Konvention festgelegt werden (wobei $|V| = 1$, $|E| = |N| - |M| - 1$)

Beispiel für Festkommadarstellung

- Das Komma bei der Binärdarstellung dient nur dem Verständnis und wird im Rechner nicht dargestellt:

Binär-muster	$n = 2, m = 0$		$n = 1, m = 1$		$n = 0, m = 2$	
	Binär	Dezimal	Binär	Dezimal	Binär	Dezimal
00	00,	0	0,0	0,0	,00	0,00
01	01,	1	0,1	0,5	,01	0,25
10	10,	2	1,0	1,0	,10	0,50
11	11,	3	1,1	1,5	,11	,75

Beispiel für Gleitkommadarstellung

- Beispiel mit $|E| = 8, |M| = 23$

V	E	M	Z
0	-13	010 0101 0110 1011 0000 0000	299,34375
1	+44	101 0101 1000 0000 0000 0000	-98574788643885416448