

Grundlagen der Informatik

- Computerinterne Informationsdarstellung -

Prof. Dr. Klaus Volbert



Hochschule für angewandte Wissenschaften
Fakultät Informatik und Mathematik

Wintersemester 2010/11
Regensburg, 12. Oktober 2010

- Wie verarbeiten Computer die für den Menschen verständliche Darstellung von Informationen?
 - Sprache, Texte, Bilder, Zahlen
 - Umwandlung in **systeminterne Darstellung**
- Information kann drei wesentliche Aspekte haben
 - Syntaktischer Aspekt: Auswertbarkeit
 - Semantischer Aspekt: Bedeutungsinhalt
 - Pragmatischer Aspekt: Verhaltensauswahl
- Hardware und Software bestimmen die Art und Darstellung der Informationen
- Heutige Rechner verwenden intern das Dualsystem
 - 0: kein Strom, keine Spannung 1: Strom, Spannung (**Einheit: 1 Bit**)

Übersicht abgeleiteter Einheiten

- 1 Bit (Binary Digit)
- 1 Byte = 1 Oktett = 2 Nibble = 2 Tetraden = 8 Bit
- 1 Maschinenwort = 1 Wort = (8 Bit | 16 Bit | 32 Bit | 64 Bit) (typische Werte)
- 1 Maschinenwort → 2 Halbworte, ½ Doppelwort, ¼ Quadwort
- Weitere Größenangaben im Dualsystem (Faktor 2^{10} statt 10^3):
 - 1 Kilobyte (KByte) = 1.024 Byte
 - 1 Megabyte (MByte) = 1.024 KByte
 - 1 Gigabyte (GByte) = 1.024 MByte
 - 1 Terabyte (TByte) = 1.024 GByte
 - 1 Petabyte (PByte) = 1.024 TByte
 - 1 Exabyte (EByte) = 1.024 PByte
 - = $(2^{10})^6$ Byte
 - = 1.152.921.504.606.846.976 Byte
- Vorsicht: Hersteller von Festplatten verwenden häufig GB oder TB in ihren Produktbezeichnungen und Rechnen nicht im Dualsystem, sondern im Dezimalsystem (Verlust bei 1 TB ggü. TByte liegt bei 10 % !)

Zahlensysteme

- Informationen (Sprache, Text, Bilder, ...) werden auf Zahlen abgebildet
- Zahlen können unterschiedlich repräsentiert werden
- Beispiel
 - Zahlen in unterschiedlichen Zahlensystemen
- Repräsentation als konkrete Darstellungsform
- Probleme
 - Rechner kann nur **endlich** viele Repräsentationen darstellen
 - Informationen können oft nur approximiert werden
- Beispiel:
 - Mengen der natürlichen, ganzen, rationalen und reellen Zahlen

Römisch	Dual	Oktal	Dezimal	Hexadezimal
	0	0	0	0
I	1	1	1	1
II	10	2	2	2
III	11	3	3	3
IV	100	4	4	4
V	101	5	5	5
VI	110	6	6	6
VII	111	7	7	7
VIII	1000	10	8	8
IX	1001	11	9	9
X	1010	12	10	A
XI	1011	13	11	B
XII	1100	14	12	C
XIII	1101	15	13	D
XIV	1110	16	14	E
XV	1111	17	15	F
XVI	10000	20	16	10
XVII	10001	21	17	11
XVIII	10010	22	18	12
XIX	10011	23	19	13
XX	10100	24	20	14
XXI	10101	25	21	15
L	110010	62	50	32
C	1100100	144	100	64
D	111110100	764	500	1F4
M	1111101000	1750	1000	3E8

Zahldarstellung von gebrochenen Zahlen

- Gebrochene Zahlen haben einen ganzzahligen Vor- und Nachkommanteil und können wie folgt beschrieben werden:

B-adisches Zahlensystem
$$n = \sum_{i=-M}^{N-1} b_i \cdot B^i$$

- B Basis des Zahlensystems ($B \in \mathbb{IN}, B \geq 2$)
- b_i Ziffern ($b_i \in \mathbb{IN}_0, 0 \leq b_i < B$)
- N Anzahl der Stellen vor dem Komma
- M Anzahl der Stellen nach dem Komma

Kurzschreibweise
$$n = (b_{N-1}b_{N-2} \dots b_1b_0, b_{-1}b_{-2} \dots b_{-M+1}b_{-M})_B$$

O.B.d.A. wird im Folgenden Komma statt Punkt verwendet!

Beispiele

- Dualzahl:

$$- (1011,101)_2 = 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 + 1 \cdot 2^{-1} + 0 \cdot 2^{-2} + 1 \cdot 2^{-3}$$

- Oktalzahl:

$$- (107,45)_8 = 1 \cdot 8^2 + 7 \cdot 8^0 + 4 \cdot 8^{-1} + 5 \cdot 8^{-2}$$

- Dezimalzahl:

$$- (4702,123)_{10} = 4 \cdot 10^3 + 7 \cdot 10^2 + 2 \cdot 10^0 + 1 \cdot 10^{-1} + 2 \cdot 10^{-2} + 3 \cdot 10^{-3}$$

- Hexadezimalzahl:

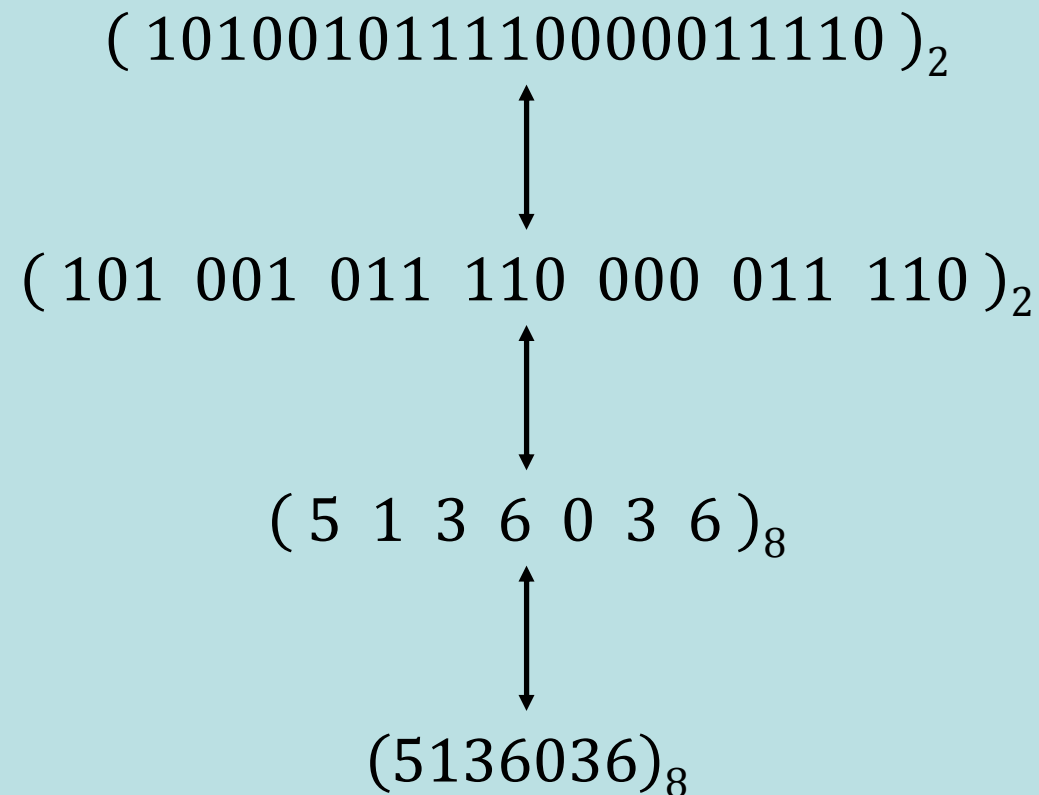
$$- (342,28)_{16} = 3 \cdot 16^2 + 4 \cdot 16^1 + 2 \cdot 16^0 + 2 \cdot 16^{-1} + 8 \cdot 16^{-2}$$

Bemerkungen zu Zahlensystemen

- B-adisches Zahlensystem wird auch **Positionssystem** genannt
- Das römische Zahlensystem ist kein Positionssystem, sondern ein **Additionssystem**
- Je größer die Basis in einem Positionssystem...
 - desto **weniger Ziffern** benötigt die Darstellung
 - desto **schwieriger** ist das Kopfrechnen („kleine Einmaleins“)
- Hexadezimaldarstellung einer Zahl ist lesbarer als die Dualdarstellung der Zahl und findet breite Anwendung
 - Programmieren in C
 - Adressierung von Speicher
 - Inhalte von Maschinenwörter (Debugger)
 - Bitmanipulationen

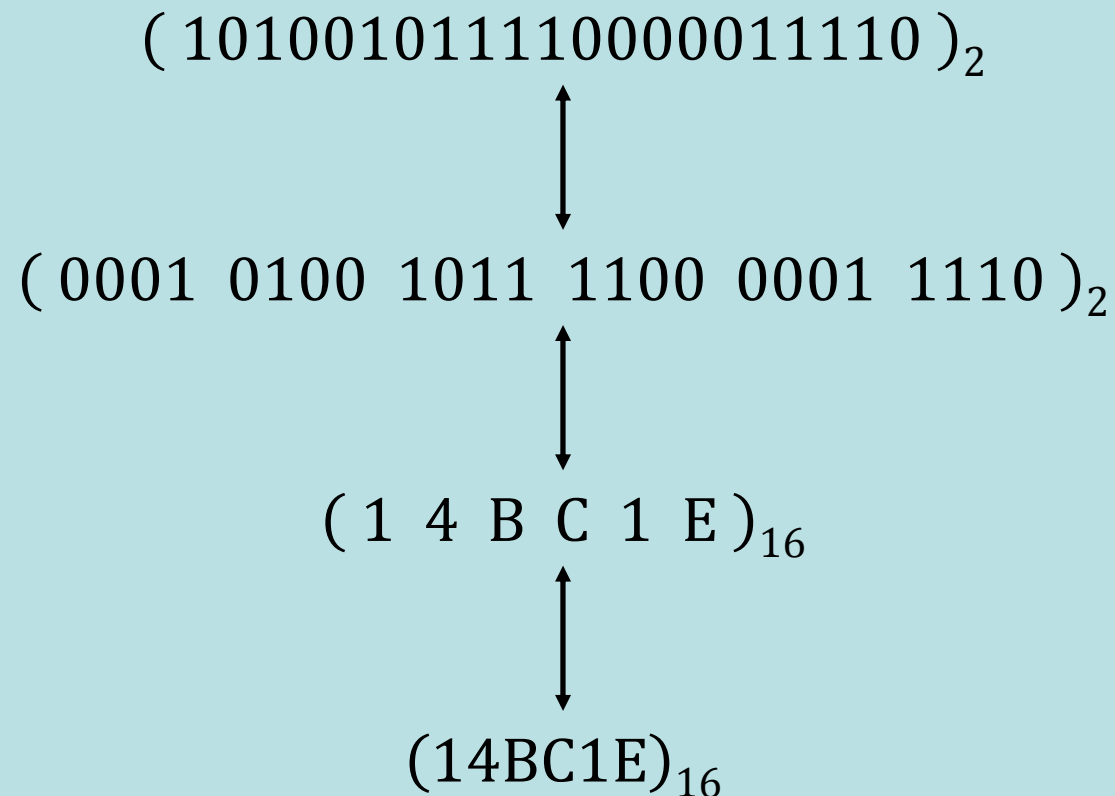
Umrechnung Dualzahl \leftrightarrow Oktalzahl

- Basis des Oktalsystems: $B = 8 = 2^3$ ermöglicht einfache Umrechnung über **Dualtriaden** (Dreiergruppen):



Umrechnung Dualzahl \leftrightarrow Hexadezimalzahl

- Basis des Hexadezimalsystems: $B = 16 = 2^4$ ermöglicht einfache Umrechnung über **Dualtetraden** (Vierergruppen):



Umrechnung Zahl \rightarrow Dezimalzahl

- Beobachtung

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

Anwendung des **Hornerschemas** liefert:

$$= ((\dots ((b_{N-1} \cdot B + b_{N-2}) \cdot B + b_{N-3}) \cdot B + \dots + b_1) \cdot B + b_0)$$

$$+ B^{-1} \cdot (b_{-1} + B^{-1} \cdot (b_{-2} + B^{-1} \cdot (b_{-3} + \dots + B^{-1} \cdot (b_{-M+1} + B^{-1} \cdot b_{-M}) \dots))$$

- Beispiel:

$$- (1011,11)_2 \rightarrow (?,?)_{10}$$

Umrechnung Dezimalzahl \rightarrow Zahl

- Idee: Horner-Schema „von außen“ verwenden
 - Vorkommateil
 - Division mit Restbildung bis das Ergebnis 0 wird (Teiler ist die Zielbasis B)
 - Weitergerechnet wird jeweils mit dem Teilergebnis
 - Die Reste ergeben die Ziffern $b_0 \dots b_{N-1}$
 - Nachkommateil inkl. „0,“
 - Multiplikation mit der Zielbasis B bis 0 erreicht wird **oder** genügend Nachkommastellen ermittelt wurden
 - Weitergerechnet wird jeweils mit dem Teilergebnis **ohne** Vorkommateil
 - Die Vorkommateile sind der Überlauf und ergeben die Ziffern $b_{-1} \dots b_{-M}$
- Übung: Formulierung als Algorithmus (Programmieren?)
- Beispiele:
 - $(11,5)_{10} \rightarrow (?,?)_2$
 - $(27,1)_{10} \rightarrow (?,?)_2$