UR

Zahlensysteme

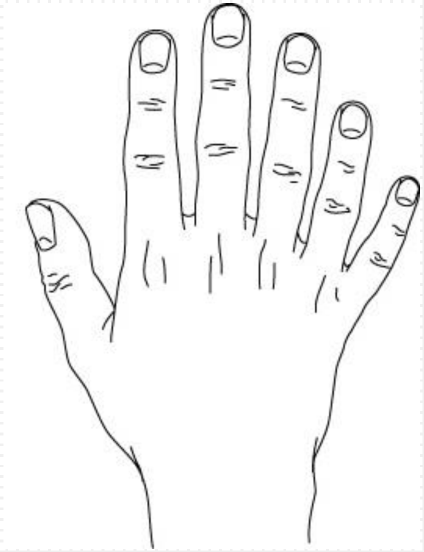
Informationswissenschaft
Universität Regensburg
Jürgen Reischer

Einführung

- Überblick:
  - Zahlensysteme im Alltag und in der Historie;
  - Zahlensysteme zu verschiedenen Zahlenbasen (Positionssysteme);
  - Zahlenkonvertierungen zwischen verschiedenen Zahlenbasen;
  - Zahlenoperatoren für arithmetische und logische Operationen für Ganz- und Bruchzahlen;
  - Zahlenkodierungen für Ganz- und Bruchzahlen im Rechner.

Einführung

- Im Laufe der Menschheitsgeschichte wurden verschiedene Zahlendarstellungs- und Rechen-systeme entwickelt:
  - Zahlensystem der Römer vs. Chinesen;
  - Rechnen mit Händen vs. Abakus:



http://upload.wikimedia.org/wikipedia/commons/e/e6/Polydactyly_postaxial.gif (6.10.2006)

http://upload.wikimedia.org/wikipedia/commons/2/27/Chinese-abacus.jpg (15.10.2008)

Einführung

- Die meisten der Systeme basieren auf der Zahl Zehn, was allein dem Abzählen mit unseren zehn Fingern geschuldet ist.
- In manchen Bereichen werden jedoch unbewusst auch andere Zahlensysteme verwendet:
  - 12er-System (Doudezimalsystem):
    1 Jahr = 12 Monate, 1 Dutzend = 12, 1 Gros = 12 Dutzend;
  - 24er-System (Quadrovigesimalsystem/Tetravigesimalsystem):
    1 Tag = 24 Stunden;
  - 60er-System (Hexagesimalsystem/Sexagesimalsystem):
    1 Stunde = 60 Minuten, 1 Minute = 60 Sekunden.

Einführung

- Begriff der Ziffer:
  - *einstellige* Zahl in einem bestimmten Zahlensystem (0 bis 9 im Dezimalsystem);
  - für jede Ziffer existiert ein eigenes (einstelliges/einfaches) Ziffernsymbol (s. arabische/römische Ziffern);
  - im Zwölfersystem etwa ist aber auch Ⅺ und Ⅻ *eine Ziffer* (z. B. *Ziffer*blatt!).

http://commons.wikimedia.org/wiki/File:BadSalzdetfurthBadenburgerStr060529.jpg (9.3.2012)

Einführung

- Wichtig ist die Unterscheidung zwischen Zahl und Zahldarstellung:
  - Eine *Zahl* ist ein *einzelnes abstraktes* Objekt:
    - Jede Zahl existiert genau *ein* Mal (Zahl als Einzelobjekt oder *Individuum*);
    - eine Zahl kann nicht mit den Sinnen, sondern nur mit dem Verstand erfasst werden.
  - Eine *Zahldarstellung* ist eine *mehrfach realisierbare konkrete* Repräsentation einer Zahl durch Symbole:
    - Eine Zahl kann auf unterschiedliche Weisen mehrfach dargestellt werden;
    - die Darstellung ist sinnlich wahrnehmbar.

Einführung

- Verschiedene Darstellungsmittel für Zahlen:
  - Alltagssprache:
    - Ziffernsymbole: "1", "10", "100", "I", "X", "C", "1/10" "3.14159265358979323846264338327950..." usw.;
    - Zahlwörter (Eigennamen für Zahlen): "eins", "zehn", "(ein)hundert", "(ein) Zehntel", "Pi" bzw. "π".
  - Formalsprache (Mathematik):
    - formalisiert/normalisiert: "1.0", "$1.0 \bullet 10^0$" (Darstellungen auf Taschenrechner);
    - operational/algorithmisch: $0+1$, $2-1$, $1 \div 1$, $1 \bullet 1$, $\sqrt{1}$, 1^2;
    - periodisch: $0.\overline{9} = 1$ (da $1 = 9 \bullet 1/9 = 9 \bullet 0.\overline{1} = 0.\overline{9}$!).

Einführung

- Grob können Zahlen in ganze und/oder gebrochene Zahlen eingeteilt werden:
  - Ganze Zahlen mit Vorkomma- ohne Nachkomma-Stellen: alle nicht-gebrochenen Zahlen von $-\infty$ bis 0 bis $+\infty$.
  - Gebrochene Zahlen mit (un)endlich vielen Nachkomma- und/oder Vorkomma-Stellen ('reelle Zahlen'):
    - *rationale Zahlen* (lat. "ratio" = 'Verhältnis' oder 'Verstand'): alle Zahlen, die durch einen Bruch X/Y aus Ganzzahlen darstellbar sind (z. B. 1/3, 3/8, 3/1, 8/3);
    - *irrationale Zahlen*: alle Zahlen mit unendlich vielen Stellen, die nicht durch einen Bruch erzeugbar (und nicht durch den Verstand fassbar) sind (z. B. π oder $\sqrt{2}$).

Römisches Zahlensystem

- Im Römischen Zahlensystem gibt es insgesamt sieben Grundeinheiten (Ziffern) zweierlei Art, durch die alle Zahlen darstellbar sind:
  - Ziffern, die einer- oder zehner-basierend sind (1, 10, 100, 1000);
  - Ziffern, die fünfer-basierend sind (5, 50, 500).
  - Für die Null gibt es kein eigenes Symbol (0 als Leerstelle).

Bereich	Ziffer	Wert	Potenz
Nuller		0	$0 \cdot 10^0$
Einer	I	1	$1 \cdot 10^0$
	V	5	$5 \cdot 10^0$
Zehner	X	10	$1 \cdot 10^1$
	L	50	$5 \cdot 10^1$
Hunderter	C	100	$1 \cdot 10^2$
	D	500	$5 \cdot 10^2$
Tausender	M	1000	$1 \cdot 10^3$

Römisches Zahlensystem

- Regeln zur Bildung Römischer Zahlen:
  - *Allgemeine Verwendungsregeln*:
    - Die einer-/zehner-basierten Werte dürfen mehrmals nebeneinander verwendet werden, wobei die einzelnen Ziffernwerte zu einem Gesamtwert addiert werden (Beispiele Tabelle rechts):
      - I, X, C: 1–4 Mal nebeneinander;
      - M: beliebig oft nebeneinander.
    - Die fünfer-basierten Werte dürfen nur *ein* Mal in einer Zahl verwendet werden (da VV = X, VVV = XV, VVVV = XX, VVVVV = XXV usw.).

Beispiel	Wert
II	2
XXX	30
CCCC	400
MMMMM	5000

Römisches Zahlensystem

- *Spezielle Kombinationsregeln*:
  - Additionsverfahren (global):
    - Grundsätzlich werden Zahlen von links nach rechts durch Aneinanderfügen von Grundeinheiten (Ziffern) gebildet, wobei größere Ziffernwerte links vor kleineren Werten stehen;
    - der Gesamtwert wird durch Aufaddieren aller Werte errechnet.
  - Subtraktionsverfahren (lokal):
    - Zur Vermeidung von vier gleichen Ziffern kann eine kleinere Einheit auch links vor einer der beiden *nächstgrößeren* Einheiten stehen (IIII = IV, VIIII = IX, XXXX = XL, LXXXX = XC, CCCC = CD, DCCCC = CM);
    - der kleinere Wert wird von der größeren Einheit lokal abgezogen, wobei das Ergebnis global wieder zum Gesamtwert addiert wird.

Römisches Zahlensystem

- Das Römische Zahlensystem weist einige Schwachpunkte auf:
  - Die Zahl 0 ist nicht explizit darstellbar bzw. implizit nur durch Abwesenheit einer Zahl realisierbar;
  - die Darstellung benutzt verschiedene Arten von Grundwerten: einer-/zehner- vs. fünferbasierte Werte;
  - ein und die selbe Zahl kann auf unterschiedliche Weisen dargestellt werden:
    - 99 = LXXXXVIIII (50+40+5+4);
    - 99 = XCVIIII (90+5+4);
    - 99 = XCIX (90+9);
    - 99 = ~~IC~~ (falsch!).

Positionssystem

- Im Gegensatz zum Römischen Zahlensystem weisen die Systeme der Chinesen u. a. eine andere Darstellungslogik auf:
  - Sie nutzen das *Positionssystem* (Stellenwertsystem) zur Darstellung aller Zahlen einschließlich der Null.
  - Der Wert einer Zahl wird durch zwei Faktoren genau festgelegt:
    - der absolute Ziffernwert (Ziffer = einstellige Zahl);
    - der relative Stellenwert einer Ziffer gemäß ihrer Position in der gesamten Zahl (Zahl = Abfolge von Ziffern).

Positionssystem – Ganzzahlen

- Im uns bekannten Dezimalsystem (Zehnersystem) werden Ganzzahlen durch Multiplikation jeder Ziffer mit ihrem Stellenwert und anschließender Addition aller Stellen dargestellt:

Stellen-wert	1000er 10^3	100er 10^2	10er 10^1	1er 10^0	Summe Zahl
Bsp. 1:	$9 \cdot 10^3$ +	$8 \cdot 10^2$ +	$7 \cdot 10^1$ +	$6 \cdot 10^0$	9876
Bsp. 2:	$0 \cdot 10^3$ +	$1 \cdot 10^2$ +	$3 \cdot 10^1$ +	$5 \cdot 10^0$	135
Bsp. 3:	$0 \cdot 10^3$ +	$0 \cdot 10^2$ +	$4 \cdot 10^1$ +	$2 \cdot 10^0$	42
Bsp. 4:	$0 \cdot 10^3$ +	$0 \cdot 10^2$ +	$0 \cdot 10^1$ +	$1 \cdot 10^0$	1
Bsp. 5:	$0 \cdot 10^3$ +	$0 \cdot 10^2$ +	$0 \cdot 10^1$ +	$0 \cdot 10^0$	0

Positionssystem – Ganzzahlen

- Allgemein kann jede *Ganzzahl* GZ mit N Stellen (*vor* dem Dezimalpunkt) als Summe aller Ziffern Z_K, multipliziert mit dem entsprechenden Stellenwert 10^K, verstanden werden ($K \geq 0$):

$$\begin{aligned} GZ &= Z_{N-1} \bullet 10^{N-1} + Z_{N-2} \bullet 10^{N-2} + \ldots + Z_1 \bullet 10^1 + Z_0 \bullet 10^0 \\ &= \Sigma_{K=N-1\ldots0}\, Z_K \bullet 10^K \end{aligned}$$

  - Z_K ist jeweils eine der Ziffern 0 bis 9;
  - 10^K ist der Stellenwert der jeweiligen Ziffer Z_K, ausgedrückt als nicht-negative Zehnerpotenz von 10^{N-1} bis hinab zu 10^0 (= 1).

Positionssystem – Ganzzahlen

- Vorteile dieser Darstellung im Positionssystem (gegenüber dem Römischen Zahlensystem):
  - Ziffern- und Stelligkeitswerte sind klar getrennt;
  - jede Zahl kann eindeutig dargestellt werden;
  - die Zahl 0 kann regulär dargestellt werden;
  - einfache Regeln für Grundrechenarten sind möglich;
  - die Systematik kann erweitert werden:
    - auf gebrochene Zahlen zwischen 0 und 1;
    - auf Systeme mit anderen Basen als 10.

Positionssystem – Bruchzahlen

- Bruchzahlen zwischen 0 und 1 können analog den Ganzzahlen dargestellt werden, wobei die Zehner-Potenzen einfach in den negativen Bereich erweitert werden (Zehntel, Hundertstel, Tausendstel usw.):

Stellen-wert	0.1er 10^{-1}	0.01er 10^{-2}	0.001er 10^{-3}	0.0001er 10^{-4}	Summe Zahl
Bsp. 1:	$9 \cdot 10^{-1}$ +	$8 \cdot 10^{-2}$ +	$7 \cdot 10^{-3}$ +	$6 \cdot 10^{-4}$	0.9876
Bsp. 2:	$0 \cdot 10^{-1}$ +	$1 \cdot 10^{-2}$ +	$3 \cdot 10^{-3}$ +	$5 \cdot 10^{-4}$	0.0135
Bsp. 3:	$0 \cdot 10^{-1}$ +	$0 \cdot 10^{-2}$ +	$4 \cdot 10^{-3}$ +	$2 \cdot 10^{-4}$	0.0042
Bsp. 4:	$0 \cdot 10^{-1}$ +	$0 \cdot 10^{-2}$ +	$0 \cdot 10^{-3}$ +	$1 \cdot 10^{-4}$	0.0001
Bsp. 5:	$0 \cdot 10^{-1}$ +	$0 \cdot 10^{-2}$ +	$0 \cdot 10^{-3}$ +	$0 \cdot 10^{-4}$	0.0000

Positionssystem – Bruchzahlen

- Allgemein kann jede *Bruchzahl* BZ mit M Stellen (*nach* dem Dezimalpunkt) als Summe aller Ziffern Z_K, multipliziert mit dem jeweiligen Stellenwert 10^K, verstanden werden ($K < 0$):

$$\begin{aligned} BZ &= Z_{-1} \cdot 10^{-1} + Z_{-2} \cdot 10^{-2} + \ldots + Z_{-M+1} \cdot 10^{-M+1} + Z_{-M} \cdot 10^{-M} \\ &= \Sigma_{K=-1\ldots-M}\, Z_K \cdot 10^K \end{aligned}$$

  - Z_K ist jeweils eine der Ziffern 0 bis 9;
  - 10^K ist der Stellenwert der jeweiligen Ziffer Z_K, ausgedrückt als negative Zehnerpotenz von 10^{-1} bis hinab zu 10^{-M}.

Positionssystem – Ganz-/Bruchzahlen

- Zusammengenommen lassen sich alle Zahlen als Kombination aus *Ganz- und Bruchzahlen* darstellen, wobei entweder der Ganz- oder Bruchzahl-Anteil fehlen kann (aber nicht beides zugleich):

Stellen-wert	10er 10^1	1er 10^0	0.1er 10^{-1}	0.01er 10^{-2}	Summe Zahl
Bsp. 1:	$9 \bullet 10^1 +$	$8 \bullet 10^0 +$	$7 \bullet 10^{-1} +$	$6 \bullet 10^{-2}$	98.76
Bsp. 2:	$0 \bullet 10^1 +$	$1 \bullet 10^0 +$	$3 \bullet 10^{-1} +$	$5 \bullet 10^{-2}$	1.35
Bsp. 3:	$0 \bullet 10^1 +$	$0 \bullet 10^0 +$	$4 \bullet 10^{-1} +$	$2 \bullet 10^{-2}$	0.42
Bsp. 4:	$0 \bullet 10^1 +$	$0 \bullet 10^0 +$	$0 \bullet 10^{-1} +$	$1 \bullet 10^{-2}$	0.01
Bsp. 5:	$0 \bullet 10^1 +$	$0 \bullet 10^0 +$	$0 \bullet 10^{-1} +$	$0 \bullet 10^{-2}$	0.0

Positionssystem – Ganz-/Bruchzahlen

- Allgemein kann jede Dezimalzahl DZ mit N+M Stellen (*vor und nach* dem Dezimalpunkt) als Summe aller Ziffern Z_K, multipliziert mit dem jeweiligen Stellenwert 10^K, verstanden werden:

$$\begin{aligned} DZ = \; & Z_{N-1}\bullet 10^{N-1} + Z_{N-2}\bullet 10^{N-2} + \ldots + Z_1\bullet 10^1 + Z_0\bullet 10^0 + \ldots + \\ & Z_{-1}\bullet 10^{-1} + Z_{-2}\bullet 10^{-2} + \ldots + Z_{-M+1}\bullet 10^{-M+1} + Z_{-M}\bullet 10^{-M} \\ = \; & \Sigma_{K=N-1\ldots 0\ldots -M}\, Z_K\bullet 10^K \end{aligned}$$

Für Zahlen mit unendlich vielen Stellen wie π oder 1/3 werden auch noch andere Darstellungen verwendet.

Positionssystem – Ganz-/Bruchzahlen

- Eine weitere Verallgemeinerung betrifft die Basis des gewählten Zahlensystems: Anstelle der Basis 10 kann auch jede andere ganzzahlige Basis verwendet werden:

Zahlensystem	Alternative Bezeichnung	Basis
Dualsystem	Binärsystem	2
Trialsystem	Trinärsystem	3
Quaternalsystem	Quaternärsystem	4
Hexalsystem	Senärsystem	6
Oktalsystem	—	8
Dezimalsystem	Denärsystem	10
Duodezimalsystem	—	12
Hexadezimalsystem	Sedezimalsystem	16

Positionssystem – Ganz-/Bruchzahlen

- Die Anzahl verschiedener Ziffern im jeweiligen System ist abhängig von der Größe der Basis (ab dem Elfersystem auch *Buchstaben* als *Ziffern*!):

Basis	**Inventar an Ziffern im jeweiligen Zahlensystem (0 bis Basis−1)**															
2	0	1														
3	0	1	2													
4	0	1	2	3												
6	0	1	2	3	4	5										
8	0	1	2	3	4	5	6	7								
10	0	1	2	3	4	5	6	7	8	9						
12	0	1	2	3	4	5	6	7	8	9	A	B				
16	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

Positionssystem – Ganz-/Bruchzahlen

- Dadurch kann die Darstellung einer Zahl Z nicht nur von der Anzahl Stellen N+M, sondern auch von der gewählten Basis B unabhängig gemacht werden:

$$
\begin{aligned}
Z &= Z_{N-1} \bullet B^{N-1} + Z_{N-2} \bullet B^{N-2} + \ldots + Z_{1} \bullet B^{1} + Z_{0} \bullet B^{0} + \ldots + \\
&\quad Z_{-1} \bullet B^{-1} + Z_{-2} \bullet B^{-2} + \ldots + Z_{-M+1} \bullet B^{-M+1} + Z_{-M} \bullet B^{-M} \\
&= \Sigma_{K=N-1\ldots0\ldots-M}\, Z_{K} \bullet B^{K}
\end{aligned}
$$

Für informationstechnische Zwecke relevant sind vor allem die Zahlensysteme zu den Basen 2, 4, 8 und 16.

Positionssystem – Dualsystem

- Beispiel Dualsystem:
  - Ganzzahlige Werte werden analog zum Dezimal-system ausschließlich unter Verwendung der zwei Ziffern 0 und 1 gebildet:

Stellen-wert	8er 2^3	4er 2^2	2er 2^1	1er 2^0	Summe Dual	Dezi-mal
Bsp. 1:	$1 \cdot 2^3 +$	$0 \cdot 2^2 +$	$1 \cdot 2^1 +$	$1 \cdot 2^0$	1011	11
Bsp. 2:	$0 \cdot 2^3 +$	$1 \cdot 2^2 +$	$1 \cdot 2^1 +$	$0 \cdot 2^0$	110	6
Bsp. 3:	$0 \cdot 2^3 +$	$0 \cdot 2^2 +$	$1 \cdot 2^1 +$	$1 \cdot 2^0$	11	3
Bsp. 4:	$0 \cdot 2^3 +$	$0 \cdot 2^2 +$	$0 \cdot 2^1 +$	$1 \cdot 2^0$	1	1
Bsp. 5:	$0 \cdot 2^3 +$	$0 \cdot 2^2 +$	$0 \cdot 2^1 +$	$0 \cdot 2^0$	0	0

Positionssystem – Dualsystem

- Entsprechend werden Bruchzahlen dargestellt:

Stellen-wert	0.5er 2^{-1}	0.25er 2^{-2}	0.125er 2^{-3}	0.0625er 2^{-4}	Summe Dual	Dezi-mal
Bsp. 1:	$1 \cdot 2^{-1}$ +	$0 \cdot 2^{-2}$ +	$1 \cdot 2^{-3}$ +	$1 \cdot 2^{-4}$	0.1011	0.6875
Bsp. 2:	$0 \cdot 2^{-1}$ +	$1 \cdot 2^{-2}$ +	$1 \cdot 2^{-3}$ +	$0 \cdot 2^{-4}$	0.0110	0.3750
Bsp. 3:	$0 \cdot 2^{-1}$ +	$0 \cdot 2^{-2}$ +	$1 \cdot 2^{-3}$ +	$1 \cdot 2^{-4}$	0.0011	0.1875
Bsp. 4:	$0 \cdot 2^{-1}$ +	$0 \cdot 2^{-2}$ +	$0 \cdot 2^{-3}$ +	$1 \cdot 2^{-4}$	0.0001	0.0625

- Zahlen mit Ganz- und Bruchzahlanteil:

Stellen-wert	2er 2^{1}	1er 2^{0}	0.5er 2^{-1}	0.25er 2^{-2}	Summe Dual	Dezi-mal
Bsp. 1:	$1 \cdot 2^{1}$ +	$0 \cdot 2^{0}$ +	$1 \cdot 2^{-1}$ +	$1 \cdot 2^{-2}$	10.11	2.75
Bsp. 2:	$0 \cdot 2^{1}$ +	$1 \cdot 2^{0}$ +	$1 \cdot 2^{-1}$ +	$0 \cdot 2^{-2}$	1.1	1.5

Positionssystem – Hexadezimalsystem

- Beispiel Hexadezimalsystem:
  - Ganzzahlige Werte werden analog zum Dezimal-system unter Verwendung der sechzehn Ziffern 0 bis 9 und A bis F gebildet:

Stellen-wert	4096er 16^3	256er 16^2	16er 16^1	1er 16^0	Summe Hex	Dezi-mal
Bsp. 1:	A • 16^3 +	F • 16^2 +	F • 16^1 +	E • 16^0	AFFE	45054
Bsp. 2:	0 • 16^3 +	8 • 16^2 +	8 • 16^1 +	8 • 16^0	888	2184
Bsp. 3:	0 • 16^3 +	0 • 16^2 +	C • 16^1 +	D • 16^0	CD	205
Bsp. 4:	0 • 16^3 +	0 • 16^2 +	0 • 16^1 +	1 • 16^0	1	1
Bsp. 5:	0 • 16^3 +	0 • 16^2 +	0 • 16^1 +	0 • 16^0	0	0

Positionssystem – Hexadezimalsystem

- Entsprechend werden Bruchzahlen dargestellt:

Stellen-wert	0.0625er 16^{-1}	~$3.9 \cdot 10^{-3}$er 16^{-2}	~$2.4 \cdot 10^{-4}$er 16^{-3}	Σ Hex	Dezimal
Bsp. 1:	A • 16^{-1} +	B • 16^{-2} +	C • 16^{-3}	0.ABC	0.6708984375
Bsp. 2:	0 • 16^{-1} +	8 • 16^{-2} +	8 • 16^{-3}	0.088	0.033203125
Bsp. 3:	0 • 16^{-1} +	0 • 16^{-2} +	1 • 16^{-3}	0.001	~$2.4414 \cdot 10^{-4}$

- Zahlen mit Ganz- und Bruchzahlanteil:

Stellen-wert	16er 16^{1}	1er 16^{0}	0.0625er 16^{-1}	Σ Hex	Dezimal
Bsp. 1:	A • 16^{1} +	B • 16^{0} +	C • 16^{-1}	AB.C	171.75
Bsp. 2:	0 • 16^{1} +	8 • 16^{0} +	8 • 16^{-1}	8.8	8.5
Bsp. 3:	0 • 16^{1} +	0 • 16^{0} +	4 • 16^{-1}	0.4	0.25

Positionssystem

- Anmerkungen:
  - Die Darstellung einer Zahl ist umso kürzer, je größer die Basis des verwendeten Zahlensystems ist:
    - Je mehr verschiedene Werte eine Stelle bzw. Ziffer annehmen kann, desto später benötigt man eine weitere Stelle für größere Zahlen (vgl. 2er- vs. 16er-System).
    - Das Dualsystem zur Basis 2 stellt dabei einen Extremfall dar:
      - Hier werden die meisten Stellen zur Anzeige einer Zahl benötigt;
      - dafür benötigt man lediglich die zwei Ziffern 0 und 1, um jede beliebige, ganze und/oder gebrochene Zahl darstellen zu können.

Positionssystem

- Bei der Darstellung von Zahlen können am Anfang und/oder Ende (aber nicht in der Mitte!) beliebig viele *Nullen* hinzugefügt oder entfernt werden, ohne dass sich der Wert der Zahl ändert:
  - *führende* Nullen: 0ABC = ABC, 0011 = 11, 000 = 0;
  - *folgende* Nullen: 0.ABC0 = 0.ABC, 0.1100 = 0.11, 0.0 = 0.
- Um eindeutig kenntlich zu machen, in welchem Zahlensystem eine Zahl dargestellt wird, sollte die Basis als Subskript hinzugefügt werden: $10000_2 = 16_{10} = 10_{16}$.

Übungsaufgaben

- Stellen Sie folgende Zahlen in Summenschreibweise dar und geben Sie jeweils die dezimalen Werte an:

System	Zahl 1	Zahl 2	Zahl 3
Dual	10	0.1	10.01
Quaternal	10	0.2	32.10
Hexal	10	0.3	54.32

Übungsaufgaben

System	Zahl 1	Zahl 2	Zahl 3
Oktal	10	0.4	75.31
Dezimal	10	0.5	96.30
Duodezimal	10	0.6	AB.BA
Hexadezimal	10	0.8	CAFE.AFFE

Übungsaufgaben

- Analysieren Sie folgende Beispiele und überlegen Sie, warum die gegebenen Zusammenhänge immer gelten müssen:

	$10_B = B_{10}$ (beliebige Basen B)	$100_B = 10_B \bullet 10_B$ (beliebige B)
Beisp.e	$10_2 = 2_{10}$, $10_3 = 3_{10}$, $10_4 = 4_{10}$, $10_8 = 8_{10}$, $10_{12} = 12_{10}$, $10_{16} = 16_{10}$	$100_2 = 10_2 \bullet 10_2$ $(2 \bullet 2 = 4)$, $100_4 = 10_4 \bullet 10_4$ $(4 \bullet 4 = 16)$, $100_{16} = 10_{16} \bullet 10_{16}$ $(16 \bullet 16 = 256)$
Begründung		

Übungsaufgaben

- Lösen Sie folgende beide Aufgaben:

Aufgabe 1:	**Aufgabe 2:**
Ermitteln Sie, für welche Basis die Gleichung $42_B + 242_B = 16_B^{\ 2}$ aufgeht.	Finden Sie eine Lösung für die Gleichung $X^Y = Y^X$ ($X \neq Y$!) in einem beliebigen Zahlensystem.

Positionssystem – Konversion

- Zur Konversion von Zahlendarstellungen mit verschiedenen Basen sind grundsätzlich zwei Methoden möglich:
  - *Umschreibung*: bestimmte Basen mit gemeinsamen Eigenschaften erlauben eine einfache Transkription von Zahlen (z. B. alle Basen, die Zweierpotenzen sind: 2, 4, 8, 16);
  - *Umrechnung*: Zahlendarstellungen zu beliebigen Basen können durch einen Algorithmus ineinander transformiert werden (z. B. 10er- ins 2er-System).

Positionssystem – Konversion – Transkription

- Umschreibung (Transkription):
  - Bei Zahlendarstellungen zu Basen, die Potenzen von 2 sind ($2^2 = 4$, $2^3 = 8$, $2^4 = 16$), können mehrere Dualstellen zu einer einzelnen Quaternal-, Oktal- oder Hexadezimalziffer zusammengefasst werden:

Dual	Quat
00	0
01	1
10	2
11	3

Dual	Oktal
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

Dual	Hex	Dual	Hex
0000	0	1000	8
0001	1	1001	9
0010	2	1010	A
0011	3	1011	B
0100	4	1100	C
0101	5	1101	D
0110	6	1110	E
0111	7	1111	F

Positionssystem – Konversion – Transkription

- Umgekehrt lassen sich durch die selbe Methode hexadezimale, oktale oder quaternale Ziffern zu Dualstellen expandieren.
- Beispiele:
  - Ganzzahlen:

<table>
<tr><th>2^N</th><th>2^{11}</th><th>2^{10}</th><th>2^9</th><th>2^8</th><th>2^7</th><th>2^6</th><th>2^5</th><th>2^4</th><th>2^3</th><th>2^2</th><th>2^1</th><th>2^0</th></tr>
<tr><td>Dual</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td></tr>
<tr><td>Quat</td><td colspan="2">3</td><td colspan="2">0</td><td colspan="2">1</td><td colspan="2">2</td><td colspan="2">0</td><td colspan="2">3</td></tr>
<tr><td>Oktal</td><td colspan="3">6</td><td colspan="3">1</td><td colspan="3">4</td><td colspan="3">3</td></tr>
<tr><td>Hex</td><td colspan="4">C</td><td colspan="4">6</td><td colspan="4">3</td></tr>
</table>

Dezimal: $2^{11}+2^{10}+2^6+2^5+2^1+2^0 = 2048+1024+64+32+2+1 = 3171$;
$C\bullet16^2+6\bullet16^1+3\bullet16^0 = 3072+96+3 = 3171$.

Positionssystem – Konversion – Transkription

* Ganz- und Bruchzahlen:

2^N	2^5	2^4	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}
Dual	0	0	1	1	0	1	1	0	1	1	0	0
4^N	4^2		4^1		4^0		4^{-1}		4^{-2}		4^{-3}	
Quat	0		3		1		2		3		0	
8^N	8^1			8^0			8^{-1}			8^{-2}		
Oktal	1			5			5			4		
16^N	...		16^0				16^{-1}				...	
Hex	...		D				B				...	

Dezimal: $2^3+2^2+2^0+2^{-1}+2^{-3}+2^{-4} = 13.6875$;
$3 \bullet 4^1+1 \bullet 4^0+2 \bullet 4^{-1}+3 \bullet 4^{-2} = 13.6875$;
$1 \bullet 8^1+5 \bullet 8^0+5 \bullet 8^{-1}+4 \bullet 8^{-2} = 13.6875$;
$D \bullet 16^0+B \bullet 16^{-1} = 13.6875$.

Positionssystem – Konversion – Transformation

- Umrechnung (Transformation):
  - Zur Umrechnung einer Zahlendarstellung A der Ausgangsbasis B_A in die Zahlendarstellung Z der Zielbasis B_Z werden zwei verschiedene Verfahren (Algorithmen) eingesetzt:
    - *Divisions(rest)methode* zur Ganzzahl-Konvertierung;
    - *Multiplikations(rest)methode* zur Bruchzahl-Konvertierung.
  - Für eine Zahl mit Ganz- und Bruchanteil müssen beide Verfahren nacheinander eingesetzt werden.

Positionssystem – Konversion – Transformation

- Algorithmus zur Konversion von *Ganzzahlen*:
  1. Die nach B_Z zu konvertierende Ausgangszahl A wird durch die Zielbasis B_Z der Zielzahl Z dividiert und dabei der Divisionsrestwert < 1 ermittelt:
     a. Divisionsergebnis: $Y := A \div B_Z$ (Vorkomma-Anteil);
     b. Divisionsrest (Modulo): $X := A \% B_Z$ (Nachkomma-Anteil).
  2. Das neu errechnete X wird die nächst*höher*wertige (linke) Ziffer der Zahl Z zur Basis B_Z, in die A gerade konvertiert wird; Y wird das neue A des nächsten Berechnungsschrittes:
     a. $Z = X_{neu}[X_{alt} \ldots X_{alt}]$;
     b. $A := Y$.
  3. Gehe wieder zu 1., solange A (= Y) noch nicht 0 ist, ansonsten ist Z fertig konvertiert und der Algorithmus endet.

Positionssystem – Konversion – Transformation

- Beispiel 1: Die Zahl 77_{10} im Dezimalsystem soll ins Dualsystem konvertiert werden:

Durchlauf	A	Y (1.a.)	X (1.b.)	Z (2.a.)	A (2.b.)
I	77	$A \div 2 = 38.\cancel{5}$	A % 2 = 1	1.	A := Y = 38
II	38	$A \div 2 = 19.\cancel{0}$	A % 2 = 0	01.	A := Y = 19
III	19	$A \div 2 = 9.\cancel{5}$	A % 2 = 1	101.	A := Y = 9
IV	9	$A \div 2 = 4.\cancel{5}$	A % 2 = 1	1101.	A := Y = 4
V	4	$A \div 2 = 2.\cancel{0}$	A % 2 = 0	01101.	A := Y = 2
VI	2	$A \div 2 = 1.\cancel{0}$	A % 2 = 0	001101.	A := Y = 1
VII	1	$A \div 2 = 0.\cancel{5}$	A % 2 = 1	1001101.	A := Y = 0
VIII	0	Beendigung des Algorithmus, da A = 0			

Positionssystem – Konversion – Transformation

- Beispiel 2: Die Zahl BCD_{16} im Hexadezimal-system soll ins Quaternalsystem konvertiert werden (statt einer einfacheren Transkription):

Durchlauf	A	Y (1.a.)	X (1.b.)	Z (2.a.)	A (2.b.)
I	BCD	A ÷ 4 = 2F3.4	A % 4 = 1	1.	A := Y = 2F3
II	2F3	A ÷ 4 = BC.~~C~~	A % 4 = 3	31.	A := Y = BC
III	BC	A ÷ 4 = 2F.~~0~~	A % 4 = 0	031.	A := Y = 2F
IV	2F	A ÷ 4 = B.~~C~~	A % 4 = 3	3031.	A := Y = B
V	B	A ÷ 4 = 2.~~C~~	A % 4 = 3	33031.	A := Y = 2
VI	2	A ÷ 4 = 0.~~8~~	A % 4 = 2	233031.	A := Y = 0
VII	0	Beendigung des Algorithmus, da A = 0			

Positionssystem – Konversion – Transformation

- Algorithmus zur Konversion von *Bruchzahlen*:
  1. Die nach B_Z zu konvertierende Ausgangszahl A wird mit der Zielbasis B_Z der Zielzahl Z multipliziert und dabei der Multiplikations-Überlauf ≥ 1 ermittelt:
     a. Multiplikationsergebnis: $Y := A \bullet B_Z$ (Nachkomma-Anteil);
     b. Multiplikationsüberlauf: $X := A ¤ B_Z$ (Vorkomma-Anteil).
  2. Das neu errechnete X wird die nächst*nieder*wertige (rechte) Ziffer der Zahl Z zur Basis B_Z, in die A gerade konvertiert wird; Y wird das neue A des nächsten Berechnungsschrittes:
     a. $Z = [X_{alt}...X_{alt}]X_{neu}$;
     b. $A := Y$.
  3. Gehe wieder zu 1., solange A (= Y) noch nicht 0 ist; ansonsten ist Z fertig konvertiert und der Algorithmus endet.

Positionssystem – Konversion – Transformation

- Beispiel 1: Die Zahl 0.6953125_{10} im Dezimal-system soll ins Dualsystem konvertiert werden:

#	A	Y (1.a.)	X (1.b.)	Z (2.a.)	A (2.b.)
I	0.6953125	A • 2 = ~~1~~.390625	A ¤ 2 = 1	.1	A := Y = 0.390625
II	0.390625	A • 2 = ~~0~~.78125	A ¤ 2 = 0	.10	A := Y = 0.78125
III	0.78125	A • 2 = ~~1~~.5625	A ¤ 2 = 1	.101	A := Y = 0.5625
IV	0.5625	A • 2 = ~~1~~.125	A ¤ 2 = 1	.1011	A := Y = 0.125
V	0.125	A • 2 = ~~0~~.25	A ¤ 2 = 0	.10110	A := Y = 0.25
VI	0.25	A • 2 = ~~0~~.5	A ¤ 2 = 0	.101100	A := Y = 0.5
VII	0.5	A • 2 = ~~1~~.0	A ¤ 2 = 1	.1011001	A := Y = 0
VIII	0	Beendigung des Algorithmus, da A = 0			

Positionssystem – Konversion – Transformation

- Beispiel 2: Die Zahl $0.BCD_{16}$ im Hexadezimal-system soll ins Quaternalsystem konvertiert werden (statt einer einfacheren Transkription):

Durchlauf	A	Y (1.a.)	X (1.b.)	Z (2.a.)	A (2.b.)
I	0.BCD	A • 4 = ~~2~~.F34	A ¤ 4 = 2	.2	A := Y = 0.F34
II	0.F34	A • 4 = ~~3~~.CD	A ¤ 4 = 3	.23	A := Y = 0.CD
III	0.CD	A • 4 = ~~3~~.34	A ¤ 4 = 3	.233	A := Y = 0.34
IV	0.34	A • 4 = ~~0~~.D	A ¤ 4 = 0	.2330	A := Y = 0.D
V	0.D	A • 4 = ~~3~~.4	A ¤ 4 = 3	.23303	A := Y = 0.4
VI	0.4	A • 4 = ~~1~~	A ¤ 4 = 1	.233031	A := Y = 0
VII	0	Beendigung des Algorithmus, da A = 0			

Positionssystem – Konversion – Transformation

- Hinweise zur Konvertierung von Zahlen:
  - Bei der Transformation einer Ganzzahl mittels Division durch die Zielbasis B_Z können immer nur Restwerte zwischen 0 und Zielbasis–1 entstehen, die dem Ziffernumfang der Zielbasis entsprechen.
  - Nicht jede Konversion zwischen Zahlendarstellungen zu beliebigen Basen terminiert: Darstellungen zur Basis B_A können endlich viele Stellen, zur Basis B_Z aber unendlich viele Ziffern besitzen (oder umgekehrt):
    - Dezimal- nach Trialsystem: 1/3 = $0.\overline{3}33\ldots$ dezimal (unendlich) = 0.1 trial (endlich);
    - Dezimal- nach Dualsystem: 2/5 = 0.4 dezimal (endlich) = $0.\overline{0110}01100110\ldots$ dual (unendlich).

Positionssystem – Konversion – Transformation

- Die Konversion ganzer und/oder gebrochener Zahlen mittels Transformation führt logischerweise zum selben Ergebnis wie eine Transkription (sofern diese überhaupt möglich ist):

<table>
<tr><th>Basis</th><th colspan="12">Vorkommastellen</th><th colspan="12">Nachkommastellen</th></tr>
<tr><td>2</td><td>1</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td></tr>
<tr><td>4</td><td colspan="2">2</td><td colspan="2">3</td><td colspan="2">3</td><td colspan="2">0</td><td colspan="2">3</td><td colspan="2">1</td><td colspan="2">2</td><td colspan="2">3</td><td colspan="2">3</td><td colspan="2">0</td><td colspan="2">3</td><td colspan="2">1</td></tr>
<tr><td>8</td><td colspan="3">5</td><td colspan="3">7</td><td colspan="3">1</td><td colspan="3">5</td><td colspan="3">5</td><td colspan="3">7</td><td colspan="3">1</td><td colspan="3">5</td></tr>
<tr><td>16</td><td colspan="4">B</td><td colspan="4">C</td><td colspan="4">D</td><td colspan="4">B</td><td colspan="4">C</td><td colspan="4">D</td></tr>
</table>

- Vorkomma-Anteil: $BCD.0_{16} = 3021.0_{10}$;
- Nachkomma-Anteil: $0.BCD_{16} = 0.737548828125_{10}$.

Übungsaufgaben

- Wandeln Sie folgende Zahlen des vorgegebenen Ausgangssystems mit einem geeigneten Verfahren in das angegebene Zielsystem um:

Konversion von Dezimal nach Dual			
$99_{10} = ?_2$	$666_{10} = ?_2$	$0.9375_{10} = ?_2$	$15.9375_{10} = ?_2$

Übungsaufgaben

Konversion Dezimal nach Trial		Konversion Dezimal nach Quat.	
$99_{10} = ?_3$	$666_{10} = ?_3$	$111_{10} = ?_4$	$2222_{10} = ?_4$
Konversion Dezimal nach Pental		**Konversion Dezimal nach Hexal**	
$333_{10} = ?_5$	$5555_{10} = ?_5$	$555_{10} = ?_6$	$8888_{10} = ?_6$

Übungsaufgaben

Konversion Dezimal nach Oktal		Konversion Dez. nach Duodez.	
$64_{10} = ?_8$	$30064_{10} = ?_8$	$131_{10} = ?_{12}$	$19006_{10} = ?_{12}$

Konversion Dezimal nach Hexadezimal			
$205_{10} = ?_{16}$	$45054_{10} = ?_{16}$	$12.8125_{10} = ?_{16}$	$0.9375_{10} = ?_{16}$

Operatoren

- Durch Operatoren werden Zahlenwerte zu einem neuen Wert verrechnet:
  - 1-stellige (unäre) Operatoren: z. B. '−', '+' (−5, +7);
  - 2-stellige (binäre) Operatoren: z. B. '−', '+', '•', '÷' (8+7, 6−5, 4•3, 2÷1).
- Dabei kommen verschiedene Arten von Operatoren zum Einsatz:
  - *arithmetische* Operatoren: Verknüpfung von Ziffern als Zahlenwerte;
  - *logische* Operatoren: Verknüpfung von Ziffern als Quasi-Wahrheitswerte (vor allem die dualen Werte 0 und 1).

Operatoren – Arithmetik

- Arithmetische Operatoren umfassen u. a. die aus dem Dezimalsystem bekannten vier Grundrechenarten, die auch auf Zahlendarstellungen zu anderen Basen anwendbar sind:
  - *'Strich'-Operatoren*: Addition und Subtraktion;
  - *'Punkt'-Operatoren*: Multiplikation und Division.
- Alle Operatoren in beliebigen Zahlensystemen funktionieren analog zum vertrauten Dezimalsystem.

Operatoren – Arithmetik – Addition

- Vorschrift zur Addition zweier Dezimalzahlen Zahl_1 und Zahl_2 mit jeweils einer bestimmten Anzahl Ziffern:
  - Ziffern gleicher Stellenwerte werden paarweise von rechts nach links einschließlich Übertrag von der vorangegangenen (rechteren) Stelle addiert (Ziffer_1+Ziffer_2+Übertrag).
  - Bei jeder Stelle gibt es zwei mögliche Arten von Additionswerten:
    - *Additionswert* < *10*: Der Wert wird in die entsprechende Stelle des Ergebnisses übernommen (kein Übertrag);
    - *Additionswert* ≥ *10*: Der Wert−10 (Einerstelle der Summe) wird in die entsprechende Stelle des Ergebnisses übernommen und zusätzlich ein Übertrag von 1 auf die nächsthöhere (linkere) Stelle addiert.

Operatoren – Arithmetik – Addition

- Beispiel:
  - Addition der beiden Dezimalzahlen 890.1234 und 2345.678 = 3235.8014 (Dezimalpunkt unten nicht dargestellt):

Wert	10^4	10^3	10^2	10^1	10^0	10^{-1}	10^{-2}	10^{-3}	10^{-4}
$Zahl_1$			8	9	0	1	2	3	4
$Zahl_2$		2	3	4	5	6	7	8	0
Übertrag	–	1	1	–	–	1	1	–	–
Summe		**3**	**2**	**3**	**5**	**8**	**0**	**1**	**4**

  - Die Vorgehensweise ist unabhängig davon, ob es sich um eine Ganz- und/oder Bruchzahl handelt.

Operatoren – Arithmetik – Addition

- Addition im Dualsystem:
  - Bei der Addition zweier Stellenwerte (Ziffern) im Dualsystem gibt es genau vier mögliche Ergebnisse.
  - Ein Übertrag für die nächste Stelle findet nur beim Summenwert $1_2+1_2 = 10_2$ statt (4. Zeile):

Ziffer_1	+	Ziffer_2	Ergebnis-Summe	Ergebnis-Ziffer	Ergebnis-Übertrag
0	+	0	00 (= 0_{10})	0	0
0	+	1	01 (= 1_{10})	1	0
1	+	0	01 (= 1_{10})	1	0
1	+	1	10 (= 2_{10})	0	1

Bei jeder Addition der Ziffern ist noch der jeweilige Übertrag als 3. Wert miteinzuberechnen!

Operatoren – Arithmetik – Addition

- Beispiel:
  - Addition der beiden Dualzahlen 111.0011_2 (7.1875_{10}) und 1011.101_2 (11.625_{10}) = 10010.1101_2 (18.8125_{10}):

Wert	2^4	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}
$Zahl_1$			1	1	1	0	0	1	1
$Zahl_2$		1	0	1	1	1	0	1	0
Übertrag	1	1	1	1	–	–	1	–	–
Summe	**1**	**0**	**0**	**1**	**0**	**1**	**1**	**0**	**1**

  - Maximal kann Summenziffer 1 und Summenübertrag 1 für die nächstlinke Stelle entstehen (s. 2^1-Stelle).

Operatoren – Arithmetik – Addition

- Addition im Hexadezimalsystem:
  - Bei der Addition zweier Stellenwerte (Ziffern) im Hexadezimalsystem gibt es insgesamt 256 Möglichkeiten (16•16 wegen 0…F•0…F).
  - Ein Übertrag entsteht immer dann, wenn die Summe zweier addierter (Hexadezimal-)Ziffern nicht mehr mit einer einstelligen Zahl dargestellt werden kann (d. h. der Wert größer als die Zahlenbasis 16 ist).
  - Der Übertrag im Hexadezimalsystem findet dabei erst ab dezimal 16 (hexadezimal 10) statt.

Operatoren – Arithmetik – Addition

- Beispiel:
  - Addition der beiden Hexadezimalzahlen $FED.4321_{16}$ ($4077.26222229003906250_{10}$) und $765D.CBA_{16}$ (30301.79541015625_{10}) = $864B.0EC1_{16}$ ($34379.05763244628906250_{10}$):

Wert	16^4	16^3	16^2	16^1	16^0	16^{-1}	16^{-2}	16^{-3}	16^{-4}
$Zahl_1$			F	E	D	4	3	2	1
$Zahl_2$		7	6	5	D	C	B	A	0
Übertrag		1	1	1	1	–	–	–	–
Summe		**8**	**6**	**4**	**B**	**0**	**E**	**C**	**1**

  - Der Stellenwert einer Ziffer ergibt sich durch Subtraktion von 16 von der Stellensumme (falls Summe ≥ 16 wird).

Operatoren – Arithmetik – Subtraktion

- Vorschrift zur Subtraktion zweier Dezimalzahlen $Zahl_1$ und $Zahl_2$ mit jeweils einer bestimmten Anzahl Ziffern ($Zahl_1 \geq Zahl_2$):
  - Ziffern gleicher Stellenwerte werden paarweise von rechts nach links einschließlich Untertrag von der vorangegangenen (rechteren) Stelle subtrahiert ($Ziffer_1 - Ziffer_2 - Untertrag$).
  - Bei jeder Stelle gibt es zwei Arten von Subtraktionswerten:
    - *Subtraktionswert* ≥ *0*: Der Wert wird in die entsprechende Stelle des Ergebnisses übernommen (kein Untertrag);
    - *Subtraktionswert* < *0*: Der Wert+10 (Einerstelle der Differenz) wird in die entsprechende Stelle des Ergebnisses übernommen und zusätzlich ein Untertrag von 1 von der nächsthöheren (linken) Stelle subtrahiert.

Operatoren – Arithmetik – Subtraktion

- Beispiel:
  - Subtraktion der beiden Dezimalzahlen 2345.678 und 789.0123 = 1556.6657 (Dezimalpunkt unten nicht dargestellt):

Wert	10^4	10^3	10^2	10^1	10^0	10^{-1}	10^{-2}	10^{-3}	10^{-4}
Zahl 1		2	3	4	5	6	7	8	0
Zahl 2			7	8	9	0	1	2	3
Untertrag		1	1	1	–	–	–	1	–
Differenz		**1**	**5**	**5**	**6**	**6**	**6**	**5**	**7**

  - Die Vorgehensweise ist unabhängig davon, ob es sich um eine Ganz- und/oder Bruchzahl handelt.

Operatoren – Arithmetik – Subtraktion

- Subtraktion im Dualsystem:
  - Bei der Subtraktion zweier Stellenwerte (Ziffern) im Dualsystem gibt nur vier mögliche Ergebnisse.
  - Der Untertrag für die nächste Stelle findet nur beim Differenzwert $0_2 - 1_2 = -1_2$ statt (2. Zeile).

$Ziffer_1$	–	$Ziffer_2$	Ergebnis-Differenz	Ergebnis-Ziffer	Ergebnis-Untertrag
0	–	0	+0	0	0
0	–	1	−1	1	1
1	–	0	+1	1	0
1	–	1	+0	0	0

Bei jeder Subtraktion der Ziffern ist noch der jeweilige Untertrag als 3. Wert einzuberechnen!

Operatoren – Arithmetik – Subtraktion

- Beispiel:
  - Subtraktion der beiden Dualzahlen 1011.011_2 (11.375_{10}) und 101.1101_2 (5.8125_{10}) = 101.1001_2 (5.5625_{10}):

Wert	2^4	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}
$Zahl_1$		1	0	1	1	0	1	1	0
$Zahl_2$			1	0	1	1	1	0	1
Untertrag		1	–	1	1	–	–	1	–
Summe		0	1	0	1	1	0	0	1

  - Maximal kann Differenzziffer 1 und Differenzuntertrag 1 für die nächstlinke Stelle entstehen.

Operatoren – Arithmetik – Subtraktion

- Subtraktion im Hexadezimalsystem:
  - Bei der Subtraktion zweier Stellenwerte (Ziffern) im Hexadezimalsystem gibt es insgesamt wiederum 256 Möglichkeiten (16•16 wegen 0…F•0…F).
  - Ein Untertrag entsteht wiederum nur dann, wenn die Differenz zweier subtrahierter Hexadezimalziffern kleiner als 0 wird, d. h. nicht mehr innerhalb einer positiven Hexadezimalstelle darstellbar ist.
  - Zur Vermeidung könnte auch schon vorher 16 addiert werden (statt danach), falls sich ein negatives Ergebnis für die Stelle abzeichnet.

Operatoren – Arithmetik – Subtraktion

- Beispiel:
  - Subtraktion der beiden Hexadezimalzahlen $765D.CBA_{16}$ (30301.79541015625_{10}) und $FED.4321_{16}$ ($4077.2622222900390625_{10}$) = $6670.887F_{16}$ ($26224.533187866210937 5_{10}$):

Wert	16^4	16^3	16^2	16^1	16^0	16^{-1}	16^{-2}	16^{-3}	16^{-4}
$Zahl_1$		7	6	5	D	C	B	A	0
$Zahl_2$			F	E	D	4	3	2	1
Untertrag		1	1	–	–	–	–	1	–
Summe		**6**	**6**	**7**	**0**	**8**	**8**	**7**	**F**

  - Der Stellenwert einer Ziffer ergibt sich durch Addition von 16 zur Stellensumme (falls Summe < 0 wird).

Operatoren – Arithmetik – Subtraktion

- Die Vorgehensweisen bei der Addition und Subtraktion von Zahlen im Positionssystem zur Basis B sind vollständig analog:

Operation	Grenzwert für Über-/Untertrag von 1 auf nachfolgende (linkere) Stelle	Korrekturwert zur Ermittlung des aktuellen Ziffern-wertes, falls Grenzwert über-/unterschritten	Verrechnung des Über-/Untertrags von vorangegan-gener (rechtere) auf aktuelle Stelle
Addition	B (Summe ≥ Basis)	−B	+1
Subtraktion	0 (Differenz < 0)	+B	−1

Operatoren – Arithmetik – Multiplikation

- Vorschrift zur Multiplikation zweier Dezimalzahlen $Zahl_1$ und $Zahl_2$ mit jeweils einer bestimmten Anzahl Ziffern:
  - Jede Ziffer Z_K aus $Zahl_2$ wird sukzessive von der niedrigsten zur höchsten Stelle (von rechts nach links) mit der gesamten $Zahl_1$ unter Berücksichtigung des Stellenwertes von Z_K multipliziert (da Z_K nur die Werte von 0 bis 9 annehmen kann, ist die Multiplikation auch durch eine Addition ersetzbar).
  - Alle Teilmultiplikationen $Zahl_1 \bullet Z_K$ werden schließlich gemäß ihrer Stellenwerte aufaddiert.

Operatoren – Arithmetik – Multiplikation

- Anders als bei der Addition und Subtraktion, wo die Zahlen bereits vor der Operation *am Dezimalpunkt ausgerichtet* werden, muss bei der Multiplikation die Anzahl der Nachkommastellen (NK) berechnet werden:
  - Die Anzahl Nachkommastellen des Multiplikationsergebnisses für beliebige Basen ist die *addierte* Anzahl der Nachkommastellen der beiden multiplizierten Zahlen (unter Einbezug sich zufällig durch die Multiplikation ergebender nachfolgender Nullen).
  - Beispiele für das Dezimalsystem:

•	16 (0 NK)	1.6 (1 NK)	0.16 (2 NK)
16 (0 NK)	256 (0 NK)	25.6 (1 NK)	2.56 (2 NK)
1.6 (1 NK)	25.6 (1 NK)	2.56 (2 NK)	0.256 (3 NK)
0.16 (2 NK)	2.56 (2 NK)	0.256 (3 NK)	0.0256 (4 NK)

Operatoren – Arithmetik – Multiplikation

- Beispiel:
  - Schrittweise Multiplikation der beiden dezimalen Ganzzahlen 5678 und 4321 = 24534638, wobei die Teilergebnisse (in der linken Spalte) bereits stellenwert-korrigiert sind:

Teilergebnisse								5678•4321
5678•1 +					5	6	7	8
5678•20 +			1	1	3	5	6	0
5678•300 +		1	7	0	3	4	0	0
5678•4000	2	2	7	1	2	0	0	0
Gesamtergebnis	**2**	**4**	**5**	**3**	**4**	**6**	**3**	**8**

Operatoren – Arithmetik – Multiplikation

- Schrittweise Multiplikation der beiden dezimalen Ganz- und Bruchzahlen 43.21 und 56.78 = 2453.4638:

Teilergebnisse								43.21•56.78
43.21•0.08 +				3	4	5	6	8
43.21•0.7 +			3	0	2	4	7	0
43.21•6 +		2	5	9	2	6	0	0
43.21•50	2	1	6	0	5	0	0	0
Gesamtergebnis	**2**	**4**	**5**	**3**	**4**	**6**	**3**	**8**

Der Dezimalpunkt braucht bei der Ermittlung der einzelnen Stellenziffern des Ergebnisses nicht berücksichtigt zu werden.

Operatoren – Arithmetik – Multiplikation

- Beispiel:
  - Schrittweise Multiplikation der beiden dualen Ganzzahlen 1011_2 (11_{10}) und 1101_2 (13_{10}) = 10001111_2 (143_{10}):

Teilergebnisse								1011•1101
1011•1 +					1	0	1	1
1011•00 +				0	0	0	0	0
1011•100 +			1	0	1	1	0	0
1011•1000		1	0	1	1	0	0	0
Gesamtergebnis	**1**	**0**	**0**	**0**	**1**	**1**	**1**	**1**

Operatoren – Arithmetik – Multiplikation

- Schrittweise Multiplikation der beiden dualen Ganz- und Bruchzahlen 10.11_2 (2.75_{10}) und 11.01_2 (3.25_{10}) = 1000.1111_2 (8.9375_{10}):

Teilergebnisse								10.11•11.01
10.11•00.01 +					1	0	1	1
10.11•00.00 +				0	0	0	0	0
10.11•01.00 +			1	0	1	1	0	0
10.11•10.00		1	0	1	1	0	0	0
Gesamtergebnis	**1**	**0**	**0**	**0**	**1**	**1**	**1**	**1**

Auch hier sind die Ziffernfolgen *im Dualen* identisch (im Dezimalen nicht!).

Operatoren – Arithmetik – Multiplikation

- Beispiel:
  - Schrittweise Multiplikation der beiden hexadezimalen Ganzzahlen $89AB_{16}$ (35243_{10}) und $FE01_{16}$ (65025_{10}) = $889833AB_{16}$ (2291676075_{10}):

Teilergebnisse								89AB•FE01
89AB•1 +					8	9	A	B
89AB•00 +				0	0	0	0	0
89AB•E00 +		7	8	7	5	A	0	0
89AB•F000	8	1	1	0	5	0	0	0
Gesamtergebnis	**8**	**8**	**9**	**8**	**3**	**3**	**A**	**B**

Operatoren – Arithmetik – Multiplikation

- Schrittweise Multiplikation der beiden hexadezimalen Ganz- und Bruchzahlen $FE.01_{16}$ (254.00390625_{10}) und $89.AB_{16}$ (137.66796875_{10}) = $8898.33AB_{16}$ ($34968.201828002929687 5_{10}$):

Teilergebnisse								FE.01•89.AB
FE.01•0.0B +				A	E	A	0	B
FE.01•0.A +			9	E	C	0	A	0
FE.01•9 +		8	E	E	0	9	0	0
FE.01•80	7	F	0	0	8	0	0	0
Gesamtergebnis	**8**	**8**	**9**	**8**	**3**	**3**	**A**	**B**

Operatoren – Arithmetik – Multiplikation

- In bestimmten Fällen kann die Multiplikation stark vereinfacht werden:
  - Bei Multiplikation mit der Basis B muss lediglich der Dezimal-punkt um eine Stelle nach *rechts* verschoben werden.
  - Der Basiswert B wird in *jedem* Zahlensystem (zur Basis B) durch die Ziffernfolge 10_B dargestellt (da $B = 1 \bullet B^1 + 0 \bullet B^0$)!
  - Beispiele:

Zahlensystem zur Basis B	Multiplikation mit B	Multiplikation mit B
Dezimalsystem	$45.67 \bullet 10_{10} = 456.7_{10}$	$456.7 \bullet 10_{10} = 4567_{10}$
Dualsystem	$11.01 \bullet 10_2 = 110.1_2$	$110.1 \bullet 10_2 = 1101_2$
Hexadezimalsystem	$89.AB \bullet 10_{16} = 89A.B_{16}$	$89A.B \bullet 10_{16} = 89AB_{16}$

Operatoren – Arithmetik – Division

- Vorschrift zur Division zweier Dezimalzahlen $Zahl_1$ und $Zahl_2$ mit jeweils einer bestimmten Anzahl Ziffern (mit $Zahl_1 \geq Zahl_2$):
  - $Zahl_2$ wird ausgehend von der höchsten Stelle in $Zahl_1$ Z_K Mal ($0 \leq Z_K \leq 9$) so abgezogen, dass der verbleibende Subtraktionsrest (von der aktuell höchsten Stelle aus betrachtet) gerade nicht negativ wird; Z_K wird die nächsthöchste Ergebnisziffer der Division.
  - Dies wird mit dem verbleibenden Differenzrest unter Heranziehung der nächsthöchsten Ziffer aus $Zahl_1$ so lange weiter durchgeführt (wiederholt), bis der Rest 0 wird oder die gewünschte Stellenzahl/Genauigkeit erreicht ist.

Operatoren – Arithmetik – Division

- Da die Division auch als Multiplikation mit dem Kehrbruch verstanden werden kann, ergibt sich die Anzahl der Nachkommastellen entsprechend:
  - Die Anzahl Nachkommastellen des Divisionsergebnisses für beliebige Basen ist die *addierte* Anzahl der Nachkommastellen aus der ersten Zahl und dem Kehrbruch der zweiten Zahl (unter Einbezug sich zufällig durch die Multiplikation ergebender nachfolgender Nullen).
  - Beispiele für das Dezimalsystem:

÷ bzw. •	4 bzw. 0.25 (2 NK)	0.4 bzw. 2.5 (1 NK)	0.04 bzw. 25 (0 NK)
15 (0 NK)	3.75 (2 NK)	37.5 (1 NK)	375 (0 NK)
1.5 (1 NK)	0.375 (3 NK)	3.75 (2 NK)	37.5 (1 NK)
0.15 (2 NK)	0.0375 (4 NK)	0.375 (3 NK)	3.75 (2 NK)

Operatoren – Arithmetik – Division

- Beispiel:
  - Schrittweise Division der beiden dezimalen Ganzzahlen 65536÷16 = 4096:

Teilergebnisse										**65536÷16**
65÷16 = 4 Rest 1 **64 = 16•4**	 –	6 6	5 4	5	3	6	÷	16	=	**4**
15÷16 = 0 Rest 15 **0 = 16•0**	 –		1 0	5 0			÷	16	=	**0**
153÷16 = 9 Rest 9 **144 = 16•9**	 –		1 1	5 4	3 4		÷	16	=	**9**
96÷16 = 6 Rest 0 **96 = 16•6**	 –				9 9	6 6	÷	16	=	**6**

Operatoren – Arithmetik – Division

- Schrittweise Division der beiden dezimalen Ganz- und Bruchzahlen 65.536 und 1.6 = 40.96 (bzw. Multiplikation von 65.536 und 0.625 = 40.960000):

Teilergebnisse										65.536÷1.6
65÷16 = 4 Rest 1 64 = 16•4	 –	6 6	5 4	5	3	6	÷	16	=	4
15÷16 = 0 Rest 15 0 = 16•0	 –		1 0	5 0			÷	16	=	0
153÷16 = 9 Rest 9 144 = 16•9	 –		1 1	5 4	3 4		÷	16	=	9
96÷16 = 6 Rest 0 96 = 16•6	 –				9 9	6 6	÷	16	=	6

Operatoren – Arithmetik – Division

- Beispiel:
  - Schrittweise Division der beiden dualen Ganzzahlen 11110_2 $(30_{10}) \div 11_2$ $(3_{10}) = 1010_2$ (10_{10}):

Teilergebnisse										11110÷11
11÷11 = 1 Rest 0 11 = 11•1	 –	1 1	1 1	1	1	0	÷	11	=	1
01÷11 = 0 Rest 1 0 = 11•0	 –		0 0	1 0			÷	11	=	0
11÷11 = 1 Rest 0 11 = 11•1	 –			1 1	1 1		÷	11	=	1
0÷11 = 0 Rest 0 0 = 11•0	 –				0 0	0 0	÷	11	=	0

Operatoren – Arithmetik – Division

- Schrittweise Division der beiden dualen Ganz- und Bruchzahlen 11.110_2 $(3.75_{10}) \div 1.1_2$ $(1.5_{10}) = 10.10_2$ (2.5_{10}):

Teilergebnisse										**11.110÷1.1**
11÷11 = 1 Rest 0 **11 = 11•1**	 –	1 1	1 1	1	1	0	÷	11	=	**1**
01÷11 = 0 Rest 1 **0 = 11•0**	 –		0 0	1 0			÷	11	=	**0**
11÷11 = 1 Rest 0 **11 = 11•1**	 –			1 1	1 1		÷	11	=	**1**
0÷11 = 0 Rest 0 **0 = 11•0**	 –				0 0	0 0	÷	11	=	**0**

Operatoren – Arithmetik – Division

- Beispiel:
  - Schrittweise Division der beiden hexadezimalen Ganzzahlen $E45DC_{16}$ (935388_{10}) ÷ 12_{16} (18_{10}) = $CAFE_{16}$ (51966_{10}):

Teilergebnisse										E45DC÷12
E4÷12 = C Rest C D8 = 12•C	 –	E D	4 8	5	D	C	÷	12	=	C
C5÷12 = A Rest 11 B4 = 12•A	 –		C B	5 4			÷	12	=	A
11D÷12 = F Rest F 10E = 12•F	 –		1 1	1 0	D E		÷	12	=	F
FC÷12 = E Rest 0 FC = 12•E	 –				F F	C C	÷	12	=	E

Operatoren – Arithmetik – Division

- Schrittweise Division der beiden hexadezimalen Ganz- und Bruchzahlen $E4.5DC_{16}$ (228.3662109375_{10}) ÷ 1.2_{16} (1.125_{10}) = $CA.FE_{16}$ (202.9921875_{10}):

Teilergebnisse										**E4.5DC÷1.2**
E4÷12 = C Rest C **D8 = 12•C**	 –	E D	4 8	5	D	C	÷	12	=	**C**
C5÷12 = A Rest 11 **B4 = 12•A**	 –		C B	5 4			÷	12	=	**A**
11D÷12 = F Rest F **10E = 12•F**	 –		1 1	1 0	D E		÷	12	=	**F**
FC÷12 = E Rest 0 **FC = 12•E**	 –				F F	C C	÷	12	=	**E**

Operatoren – Arithmetik – Division

- In bestimmten Fällen kann die Division stark vereinfacht werden:
  - Bei Division durch die Basis B muss lediglich der Dezimalpunkt um eine Stelle nach *links* verschoben werden.
  - Der Basiswert B wird in *jedem* Zahlensystem (zur Basis B) durch die Ziffernfolge 10_B dargestellt (da $B = 1 \bullet B^1 + 0 \bullet B^0$)!
  - Beispiele:

Zahlensystem zur Basis B	Division mit B	Division mit B
Dezimalsystem	$4567 \div 10_{10} = 456.7_{10}$	$456.7 \div 10_{10} = 45.67_{10}$
Dualsystem	$1101 \div 10_2 = 110.1_2$	$110.1 \div 10_2 = 11.01_2$
Hexadezimalsystem	$89AB \div 10_{16} = 89A.B_{16}$	$89A.B \div 10_{16} = 89.AB_{16}$

Übungsaufgaben

- Führen Sie die angegebenen Operationen im vorgegebenen Zahlensystem *ohne Konversionen* durch:

System	Addition	Subtraktion	Multiplikation	Division
Dual	11100100+ 10010011	11100100− 10010011	11•11	110÷10
	1110.0100+ 1001.0011	1110.0100− 1001.0011	101•110	100011÷101
	1011.01+ 10.1101	1011.01− 10.1101		

Übungsaufgaben

System	Multiplikation	Division
Dual	1001•1001	1010001÷1001
	101•11011	11111111÷10001

Übungsaufgaben

System	Addition		Subtraktion	
Trial	2220+ 1211	21011220+ 12121021	2220− 1211	21011220− 12121021
Qua- ternal	3201+ 1320	23010132+ 00332211	3201− 1320	23010132− 00332211
Hexal	5204+ 3415	41203542+ 32415520	5204− 3415	41203542− 32415520

Übungsaufgaben

System	Addition		Subtraktion	
Oktal	7513+ 4620	70615243+ 52437061	7513− 4620	70615243− 52437061
Duo- dezi- mal	AB05+ 9637	6420A93B+ 50873142	AB05− 9637	6420A93B− 50873142
Hexa- dezi- mal	FADE+ AFFE	9A8B7C6D+ 0F1E2D3C	FADE− AFFE	9A8B7C6D− 0F1E2D3C

Operatoren – Logik

- Logische Operatoren umfassen die in der Logik verwendeten Operationen für zwei Wahrheits-werte (anstelle von Ziffern):
  - Wahrheitswert 'wahr' (kurz 'w', engl. 'true' oder 't');
  - Wahrheitswert 'falsch' (kurz 'f', engl. 'false' oder 'f').
- Im informationstechnischen Bereich werden anstelle von Wahrheitswerten 'w' und 'f' die beiden Dualwerte 0 und 1 verwendet:
  - Dualwert 1 wird als 'wahr' interpretiert;
  - Dualwert 0 wird als 'falsch' interpretiert.

Operatoren – Logik

- Üblicherweise werden logische Operationen nur auf Zahlendarstellungen im *Dualsystem* mit den Ziffern 0=falsch und 1=wahr angewendet.
- Da jedoch jede Zahl auch immer im Dualsystem darstellbar ist, sind logische Operatoren indirekt auf alle Zahlen anwendbar.
- Insbesondere die 2^N-basierten Zahlensysteme (Quaternal-/Oktal-/Hexadezimalsystem) eignen sich hierfür nach vorheriger Transkription.

Operatoren – Logik

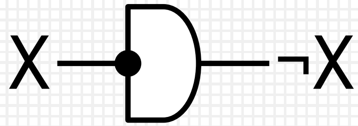
- Die logischen Operationen umfassen vor allem die folgenden Operatoren:

Operation/Name	Definition (s. u.)	Kurzzeichen	Interpretation
Negation	Z := NOT X	¬ oder –	NICHT
Konjunktion	Z := X AND Y	∧ oder &	UND, ABER
Disjunktion	Z := X OR Y	∨ oder |	UND/ODER
Kontrajunktion	Z := X XOR Y	⊕	ENTWEDER-ODER
Subjunktion	Z := X SBJU Y	→	WENN-DANN
Bi(sub)junktion	Z := X BIJU Y	↔	NUR-DANN-WENN
Negatkonjunktion	Z := X NAND Y	⊼	?
Negatdisjunktion	Z := X NOR Y	⊽	WEDER-NOCH

Operatoren – Logik – Definition

- Logische Negation (Einer-Komplement):
  - Beschreibung:
    - Z := 1, wenn X = 0;
    - Z := 0, wenn X = 1.
  - Definition: Z := NOT X.
  - Logisches Gatter:
  - Beispiele mit mehreren Dualziffern pro Zahl:

Kombination	1	2
X	0	1
(kein Y)		
Z := NOT X	1	0

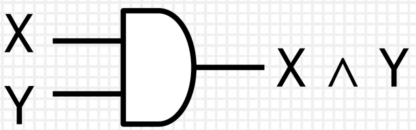


Beispiele	Bsp. 1 Dual:	Bsp. 2 Hex:
X	1001	B3 = 10110011
(kein Y)		
Z := NOT X	0110	4C = 01001100

Operatoren – Logik – Definition

- Logische Konjunktion:
  - Beschreibung:
    - Z := 1, wenn X = Y = 1;
    - Z := 0, sonst.
  - Definition: Z := X AND Y.
  - Logisches Gatter:
  - Beispiele mit mehreren Dualziffern pro Zahl:

Kombination	1	2	3	4
X	0	0	1	1
Y	0	1	0	1
Z := X AND Y	0	0	0	1

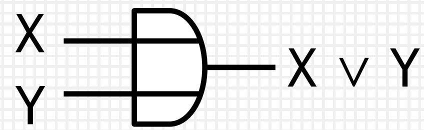


Beispiele	Bsp. 1 Dual:	Bsp. 2 Hex:
X	1001 (= 9_{10})	B3 = 10110011
Y	1100 (= 12_{10})	69 = 01101001
Z := X AND Y	1000 (= 8_{10})	21 = 00100001

Operatoren – Logik – Definition

- Logische Disjunktion:
  - Beschreibung:
    - Z := 0, wenn X = Y = 0;
    - Z := 1, sonst.
  - Definition: Z := X OR Y.
  - Logisches Gatter:
  - Beispiele mit mehreren Dualziffern pro Zahl:

Kombination	1	2	3	4
X	0	0	1	1
Y	0	1	0	1
Z := X OR Y	0	1	1	1

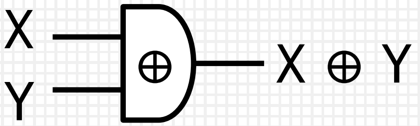


Beispiele	Bsp. 1 Dual:	Bsp. 2 Hex:
X	1001 (= 9_{10})	B3 = 10110011
Y	1100 (= 12_{10})	69 = 01101001
Z := X OR Y	1101 (= 13_{10})	FB = 11111011

Operatoren – Logik – Definition

- Logische Kontrajunktion (Kontra-/Antivalenz):
  - Beschreibung:
    - Z := 1, wenn X ≠ Y;
    - Z := 0, wenn X = Y.
  - Definition: Z := X XOR Y.
  - Logisches Gatter:
  - Beispiele mit mehreren Dualziffern pro Zahl:

Kombination	1	2	3	4
X	0	0	1	1
Y	0	1	0	1
Z := X XOR Y	0	1	1	0



Beispiele	Bsp. 1 Dual:	Bsp. 2 Hex:
X	1001 (= 9_{10})	B3 = 10110011
Y	1100 (= 12_{10})	69 = 01101001
Z := X XOR Y	0101 (= 5_{10})	DA = 11011010

Operatoren – Logik – Definition

- Logische Subjunktion:
  - Beschreibung:
    - Z := 0, wenn X = 1 und Y = 0;
    - Z := 1, sonst.
  - Definition: Z := X SBJU Y.
  - Logisches Gatter: (kein Gattersymbol!)
  - Beispiele mit mehreren Dualziffern pro Zahl:

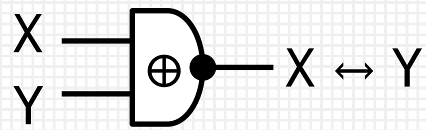
Kombination	1	2	3	4
X	0	0	1	1
Y	0	1	0	1
Z := X SBJU Y	1	1	0	1

Beispiele	Bsp. 1 Dual:	Bsp. 2 Hex:
X	1001 (= 9_{10})	B3 = 10110011
Y	1100 (= 12_{10})	69 = 01101001
Z := X SBJU Y	1110 (= E_{10})	6D = 01101101

Operatoren – Logik – Definition

- Logische Bi(sub)junktion:
  - Beschreibung:
    - Z := 1, wenn X = Y;
    - Z := 0, wenn X ≠ Y.
  - Definition: Z := X BIJU Y.
  - Logisches Gatter:
  - Beispiele mit mehreren Dualziffern pro Zahl:

Kombination	1	2	3	4
X	0	0	1	1
Y	0	1	0	1
Z := X BIJU Y	1	0	0	1

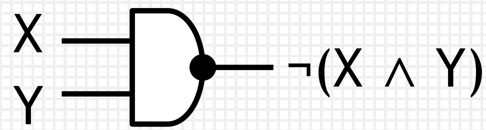


Beispiele	Bsp. 1 Dual:	Bsp. 2 Hex:
X	1001 (= 9_{10})	B3 = 10110011
Y	1100 (= 12_{10})	69 = 01101001
Z := X BIJU Y	1010 (= 10_{10})	25 = 00100101

Operatoren – Logik – Definition

- Logische Negatkonjunktion:
  - Beschreibung:
    - Z := 0, wenn X = Y = 1;
    - Z := 1, sonst.
  - Definition: Z := X NAND Y.
  - Logisches Gatter:
  - Beispiele mit mehreren Dualziffern pro Zahl:

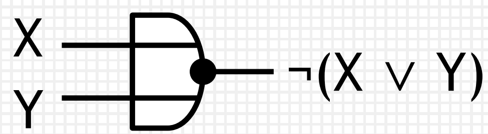
Kombination	**1**	**2**	**3**	**4**
X	0	0	1	1
Y	0	1	0	1
Z := X NAND Y	1	1	1	0



Beispiele	**Bsp. 1 Dual:**	**Bsp. 2 Hex:**
X	1001 (= 9_{10})	B3 = 10110011
Y	1100 (= 12_{10})	69 = 01101001
Z := X NAND Y	0111 (= 7_{10})	DE = 11011110

Operatoren – Logik – Definition

- Logische Negatdisjunktion:
  - Beschreibung:
    - Z := 1, wenn X = Y = 0;
    - Z := 0, sonst.
  - Definition: Z := X NOR Y.
  - Logisches Gatter:



Kombination	1	2	3	4
X	0	0	1	1
Y	0	1	0	1
Z := X NOR Y	1	0	0	0

  - Beispiele mit mehreren Dualziffern pro Zahl:

Beispiele	Bsp. 1 Dual:	Bsp. 2 Hex:
X	1001 (= 9_{10})	B3 = 10110011
Y	1100 (= 12_{10})	69 = 01101001
Z := X NOR Y	0010 (= 2_{10})	04 = 00000100

Operatoren – Logik – Übersicht

- Zusammenfassung:

X	0	0	1	1
Y	0	1	0	1
Z := NOT X	1	1	0	0
Z := X AND Y	0	0	0	1
Z := X OR Y	0	1	1	1
Z := X XOR Y	0	1	1	0
Z := X SBJU Y	1	1	0	1
Z := X BIJU Y	1	0	0	1
Z := X NAND Y	1	1	1	0
Z := X NOR Y	1	0	0	0

Bei NOT als logischem Operator für nur einen Operanden X ist der Wert von Y unerheblich.

Bis auf SBJU (!) sind alle Operatoren *kommutativ*, d. h. die Werte von X und Y sind vertauschbar, ohne dass sich das Ergebnis ändert.

Operatoren – Logik – Anwendung

- Anwendungsbeispiele für logische Operatoren:
  - Rekonstruktion der Addition und Subtraktion:

X	0	0	1	1
Y	0	1	0	1
Summenziffer Z := X XOR Y	0	1	1	0
Übertrag Z := X AND Y	0	0	0	1

X	0	0	1	1
Y	0	1	0	1
Differenzziffer Z := X XOR Y	0	1	1	0
Untertrag Z := NOT (Y SBJU X)	0	1	0	0

Berechnung der Ziffern für Addition und Subtraktion identisch

Operatoren – Logik – Anwendung

- Umwandlung von Groß- in Kleinbuchstaben und umgekehrt im ASCII-Kode:
  - Kodierungen (Unterschied in 2^5-Stelle):
    - Großbuchstaben: 'A' = $01\underline{0}00001_2$ = 65 bis 'Z' = $01\underline{0}11010_2$ = 90;
    - Kleinbuchstaben: 'a' = $01\underline{1}00001_2$ = 97 bis 'z' = $01\underline{1}11010_2$ = 122.
  - Umwandlung durch Verknüpfung des ASCII-Kodes mit dem jeweils passenden Wert durch OR bzw. AND:

'A' nach 'a'	ASCII-Kode
'A' ∨ (OR)	01000001
2^5 =	00100000
'a'	01100001

'a' nach 'A'	ASCII-Kode
'a' ∧ (AND)	01100001
NOT 2^5 =	11011111
'A'	01000001

Übungsaufgaben

- Verknüpfen Sie folgende Zahlen mit den angegebenen logischen Operatoren im vorgegebenen Zahlensystem (*vorübergehende* Konversion nach Dual erlaubt):

Operator	Dual	Quaternal	Oktal
Negation	¬10001101	¬3210	¬7205
Kon-junktion	10011110 ∧ 11100100	3102 ∧ 2031	7531 ∧ 2460

Übungsaufgaben

Operator	Dezimal	Hexadezimal
Negation	¬2782	¬CAFE
Kon- junktion	2782∧ 1620	CAFE∧ 0815

Übungsaufgaben

Operator	Dual	Quaternal	Oktal
Dis-junktion	10011110∨ 11100100	3102∨ 2031	7531∨ 2460
Kontra-junktion	10011110⊕ 11100100	3102⊕ 2031	7531⊕ 2460
Sub-junktion	10011110→ 11100100	3102→ 2031	7531→ 2460

Übungsaufgaben

Operator	Dezimal	Hexadezimal
Dis-junktion	2782 ∨ 1620	CAFE ∨ 0815
Kontra-junktion	2782 ⊕ 1620	CAFE ⊕ 0815
Sub-junktion	2782 → 1620	CAFE → 0815

Übungsaufgaben

Operator	Dual	Quaternal	Oktal
Bi-junktion	10011110↔ 11100100	3102↔ 2031	7531↔ 2460
Negat-kon-junktion	10011110⊼ 11100100	3102⊼ 2031	7531⊼ 2460
Negat-dis-junktion	10011110⊽ 11100100	3102⊽ 2031	7531⊽ 2460

Übungsaufgaben

Operator	Dezimal	Hexadezimal
Bi-junktion	2782↔ 1620	CAFE↔ 0815
Negat-kon-junktion	2782⊼ 1620	CAFE⊼ 0815
Negat-dis-junktion	2782⊽ 1620	CAFE⊽ 0815

Operatoren – Logik – Gesetze

- In der Logik gelten eine Reihe von Gesetzen (teilweise analog zur Arithmetik), durch die die Gleichheit logischer Aussagen expliziert wird:

Anzahl Operanden	Gesetze
1: X	Negationsgesetz, Idempotenzgesetz, Komplementgesetz, Identitätsgesetz, Dominanzgesetz
2: X und Y	Kommutativgesetz, Absorptionsgesetz, DeMorgan-Gesetz
3: X, Y und Z	Assoziativgesetz, Distributivgesetz

Operatoren – Logik – Gesetze

- Gesetze für eine Variable X:

Gesetz	Formulierung	Beispiele
Negationsgesetz	$\neg(\neg X) = X$	$\neg(\neg 0) = 0$, $\neg(\neg 1) = 1$
Idempotenzgesetz	$X \wedge X = X$ $X \vee X = X$	$1 \wedge 1 = 1$, $0 \wedge 0 = 0$ $1 \vee 1 = 1$, $0 \vee 0 = 0$
Komplementgesetz	$X \wedge \neg X = 0$ $X \vee \neg X = 1$	$1 \wedge \neg 1 = 0$, $0 \wedge \neg 0 = 0$ $1 \vee \neg 1 = 1$, $0 \vee \neg 0 = 1$
Identitätsgesetz	$X \wedge 1 = X$ $X \vee 0 = X$	$1 \wedge 1 = 1$, $0 \wedge 1 = 0$ $1 \vee 0 = 1$, $0 \vee 0 = 0$
Dominanzgesetz	$X \wedge 0 = 0$ $X \vee 1 = 1$	$1 \wedge 0 = 0$, $0 \wedge 0 = 0$ $1 \vee 1 = 1$, $0 \vee 1 = 1$

Operatoren – Logik – Gesetze

- Gesetze für zwei unterschiedliche Variablen X und Y:

Gesetz	Formulierung	Beispiele
Kommutativgesetz	$X \wedge Y = Y \wedge X$ $X \vee Y = Y \vee X$	$0 \wedge 1 = 1 \wedge 0$ $0 \vee 1 = 1 \vee 0$
Absorptionsgesetz	$X \wedge (X \vee Y) = X$ $X \vee (X \wedge Y) = X$	$0 \wedge (0 \vee 1) = 0$ $1 \wedge (1 \vee 0) = 1$ $0 \vee (0 \wedge 1) = 0$ $1 \vee (1 \wedge 0) = 1$
DeMorgan-Gesetz	$\neg(X \wedge Y) = \neg X \vee \neg Y$ $\neg(X \vee Y) = \neg X \wedge \neg Y$	$\neg(0 \wedge 1) = \neg 0 \vee \neg 1$ $\neg(0 \vee 1) = \neg 0 \wedge \neg 1$

Operatoren – Logik – Gesetze

- Gesetze für drei unterschiedliche Variablen X, Y und Z:

Gesetz	Formulierung	Beispiele
Assoziativgesetz	$X \land (Y \land Z) =$ $(X \land Y) \land Z$ $X \lor (Y \lor Z) =$ $(X \lor Y) \lor Z$	$1 \land (0 \land 1) =$ $(1 \land 0) \land 1$ $1 \lor (0 \lor 1) =$ $(1 \lor 0) \lor 1$
Distributivgesetz	$X \land (Y \lor Z) =$ $(X \land Y) \lor (X \land Z)$ $X \lor (Y \land Z) =$ $(X \lor Y) \land (X \lor Z)$	$1 \land (1 \lor 0) =$ $(1 \land 1) \lor (1 \land 0)$ $1 \lor (1 \land 0) =$ $(1 \lor 1) \land (1 \lor 0)$

Operatoren – Logik – Gesetze

- Der Nachweis für die Richtigkeit der Gesetze kann im einfachsten Falle durch so genannte Wertetabellen erfolgen:
  - Nachweis für das Komplementgesetz (1 Operand X):

Nachweis	**X**	**¬X**	**X ∧ ¬X**	**0 (erwartet)**
X ∧ ¬X = 0	0 1	1 0	0 ∧ 1 = 0 1 ∧ 0 = 0	0 ✓ 0 ✓
Nachweis	**X**	**¬X**	**X ∨ ¬X**	**1 (erwartet)**
X ∨ ¬X = 1	0 1	1 0	0 ∨ 1 = 1 1 ∨ 0 = 1	1 ✓ 1 ✓

Das Gesetz ist bewiesen, wenn sich für *alle möglichen Belegungen* der Variablen (hier X = 0 oder X = 1) der erwartete Wert rechts von '=' ergibt.

Operatoren – Logik – Gesetze

- Nachweis für das Absorptionsgesetz (2 Operanden X und Y):

Nachweis	X	Y	(X ∨ Y)	X ∧ (X ∨ Y)	X (erwartet)
X ∧ (X ∨ Y) = X	0 0 1 1	0 1 0 1	0 1 1 1	0 ∧ 0 = 0 0 ∧ 1 = 0 1 ∧ 1 = 1 1 ∧ 1 = 1	0 (= X) ✓ 0 (= X) ✓ 1 (= X) ✓ 1 (= X) ✓
Nachweis	**X**	**Y**	**(X ∧ Y)**	**X ∨ (X ∧ Y)**	**X (erwartet)**
X ∨ (X ∧ Y) = X	0 0 1 1	0 1 0 1	0 0 0 1	0 ∨ 0 = 0 0 ∨ 0 = 0 1 ∨ 0 = 1 1 ∨ 1 = 1	0 (= X) ✓ 0 (= X) ✓ 1 (= X) ✓ 1 (= X) ✓

Operatoren – Logik – Gesetze

- Nachweis für das Assoziativgesetz (3 Operanden X, Y und Z):

Nachweis	X	Y	Z	$(Y \wedge Z)$	$(X \wedge Y)$	$X \wedge (Y \wedge Z)$	$(X \wedge Y) \wedge Z$
$X \wedge (Y \wedge Z) =$ $(X \wedge Y) \wedge Z$	0	0	0	0	0	$0 \wedge 0 = 0$	$0 \wedge 0 = 0$ ✓
	0	0	1	0	0	$0 \wedge 0 = 0$	$0 \wedge 1 = 0$ ✓
	0	1	0	0	0	$0 \wedge 0 = 0$	$0 \wedge 0 = 0$ ✓
	0	1	1	1	0	$0 \wedge 1 = 0$	$0 \wedge 1 = 0$ ✓
	1	0	0	0	0	$1 \wedge 0 = 0$	$0 \wedge 0 = 0$ ✓
	1	0	1	0	0	$1 \wedge 0 = 0$	$0 \wedge 1 = 0$ ✓
	1	1	0	0	1	$1 \wedge 0 = 0$	$1 \wedge 0 = 0$ ✓
	1	1	1	1	1	$1 \wedge 1 = 1$	$1 \wedge 1 = 1$ ✓

(2. Hälfte s. nächste Seite)

Operatoren – Logik – Gesetze

Nachweis	X	Y	Z	(Y ∨ Z)	(X ∨ Y)	X ∨ (Y ∨ Z)	(X ∨ Y) ∨ Z
X ∨ (Y ∨ Z) = (X ∨ Y) ∨ Z	0	0	0	0	0	0 ∨ 0 = 0	0 ∨ 0 = 0 ✓
	0	0	1	1	0	0 ∨ 1 = 1	0 ∨ 1 = 1 ✓
	0	1	0	1	1	0 ∨ 1 = 1	1 ∨ 0 = 1 ✓
	0	1	1	1	1	0 ∨ 1 = 1	1 ∨ 1 = 1 ✓
	1	0	0	0	1	1 ∨ 0 = 1	1 ∨ 0 = 1 ✓
	1	0	1	1	1	1 ∨ 1 = 1	1 ∨ 1 = 1 ✓
	1	1	0	1	1	1 ∨ 1 = 1	1 ∨ 0 = 1 ✓
	1	1	1	1	1	1 ∨ 1 = 1	1 ∨ 1 = 1 ✓

- Je nach Anzahl der Variablen müssen entsprechend viele Fälle durchgetestet werden: bei 1 Variable X 2 Fälle, bei 2 Variablen X und Y 4 Fälle, bei 3 Variablen X, Y und Z 8 Fälle usw.
- Allgemein müssen bei N Variablen 2^N Fälle getestet werden, da *jede* der Variablen genau zwei Zustände 0 und 1 annehmen kann.

Operatoren – Logik – Gesetze

- Mittels Wertetabellen und Logik-Gesetzen lässt sich nachweisen, dass die logischen Operatoren NAND bzw. NOR alleine (!) bereits ausreichend wären, um *alle* anderen logischen Operatoren nachzubilden:
  - Zunächst können NAND und NOR als explizite Negation von AND und OR umgeschrieben werden (s. entsprechende Definitionen/Wertetabellen und logische Gatter):

Operator	NAND bzw. NOR in impliziter Normalschreibweise	NAND bzw. NOR als explizite Negation von AND bzw. OR
NAND	$X \barwedge Y$ (X NAND Y)	$\neg(X \wedge Y)$
NOR	$X \veebar Y$ (X NOR Y)	$\neg(X \vee Y)$

Operatoren – Logik – Gesetze

- Ferner lässt sich jede Negation von Z (d. h. ¬Z) ausschließlich durch NAND bzw. NOR ausdrücken:

Operator	Umschreibung
NAND	$\neg Z = Z \barwedge Z$
NOR	$\neg Z = Z \veebar Z$

mit Definitionen von NAND und NOR wie in Tabelle rechts

Fälle	X	Y	NAND	NOR
Z = X = Y	0	0	1	1
(hier nicht relevant)	0	1	1	0
	1	0	1	0
Z = X = Y	1	1	0	0

- Unter Ausnutzung obiger Zusammenhänge ergibt sich:

Operat.	AND/OR aus NAND/NOR und NOT	Z = (X ⊼ Y) bzw. Z = (X ⊽ Y)
(N)AND	$(X \wedge Y) = \neg(\neg(X \wedge Y)) = \neg(X \barwedge Y)$	$\neg(X \barwedge Y) = (X \barwedge Y) \barwedge (X \barwedge Y)$
(N)OR	$(X \vee Y) = \neg(\neg(X \vee Y)) = \neg(X \veebar Y)$	$\neg(X \veebar Y) = (X \veebar Y) \veebar (X \veebar Y)$

Operatoren – Logik – Gesetze

- Da man AND, OR und NOT alleine aus NAND (bzw. NOR) zusammenkonstruieren kann, lassen sich *alle* anderen möglichen logischen Funktionen erzeugen (s. Aufgaben unten).
- Das heißt, dass das gesamte Computeruniversum im Sinne *jeder* Funktion/Berechnung, die ein Computer *überhaupt* ausführen kann, alleine durch zwei Arten logischer Entitäten realisiert werden kann:
  - durch zwei logische Werte 0 und 1 (bzw. wahr/falsch);
  - durch den logischen Operator NAND (bzw. entsprechend NOR), der zwei logische Werte 0 und/oder 1 verknüpft.

Übungsaufgaben

- Bestimmen Sie durch Wertetabellen, ob die linken und rechten Seiten jeweils gleich sind:

(Un-)Gleichung:	Rechnung (Wertetabelle):
$X \rightarrow Y \stackrel{?}{=}$ $\neg X \vee Y$	
$X \rightarrow Y \stackrel{?}{=}$ $\neg(X \wedge \neg Y)$	

Übungsaufgaben

$\neg X \rightarrow \neg Y \overset{?}{=}$ $X \rightarrow Y$	
$\neg X \rightarrow \neg Y \overset{?}{=}$ $Y \rightarrow X$	
$X \leftrightarrow Y \overset{?}{=}$ $(X \rightarrow Y) \wedge (Y \rightarrow X)$	

Übungsaufgaben

$X \oplus Y \stackrel{?}{=}$ $\neg(X \leftrightarrow Y)$	
$X \oplus Y \stackrel{?}{=}$ $\neg(X \rightarrow Y) \wedge \neg(Y \rightarrow X)$	
$X \oplus Y \stackrel{?}{=}$ $(X \wedge \neg Y) \vee (\neg X \wedge Y)$	

Übungsaufgaben

- Weisen Sie ferner die Gültigkeit der beiden DeMorgan-Gesetze per Wertetabelle nach:

Gesetz 1:	Gesetz 2:
$\neg(X \land Y) = \neg X \lor \neg Y$	$\neg(X \lor Y) = \neg X \land \neg Y$

Zahlenkodierung

- Ganz- und Bruchzahlen werden im Rechner auf unterschiedliche Weise kodiert:
  - Ganzzahlen werden im so genannten *Zweier-komplement* (*arithmetische* Negation) kodiert;
  - Bruchzahlen werden in der (wissenschaftlichen) *Exponenten-Darstellung* kodiert (Taschenrechner):
    - Bruchzahlen sind alle reellen (Dual-)Zahlen mit einer bestimmten Anzahl Vor- und Nachkommastellen;
    - Ganzzahlen sind dann als 'Sonderfall' von Bruchzahlen realisierbar (d. h. alle Nachkomma-Stellen sind 0).

Zahlenkodierung – Ganzzahlen

- Kodierung von Ganzzahlen im Zweierkomplement:
  - Ganzzahlen werden im Rechner grundsätzlich mit einer festen Anzahl Stellen gespeichert: 8, 16, 32, 64 oder noch mehr Dualstellen.
  - Davon wird die höchste (linkeste) Stelle als Vorzeichen ± interpretiert (Beispiel für 8-/16-stellige ganze Dualzahlen, wobei X für eine 0 oder 1 steht):

Stellen	2^{15}	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^{9}	2^{8}	2^{7}	2^{6}	2^{5}	2^{4}	2^{3}	2^{2}	2^{1}	2^{0}
16	±	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
8									±	X	X	X	X	X	X	X

Positive Zahlen enthalten eine 0 an der Vorzeichenstelle ±, negative Zahlen eine 1.

Zahlenkodierung – Ganzzahlen

- Durch den Wegfall einer Dualstelle halbiert sich die *Größe* der darstellbaren Dualwerte, die Gesamtzahl darstellbarer Werte bleibt gleich:

Bereich # Stellen	negativer Wertebereich	positiver Wertebereich	Gesamtzahl dar-stellbarer Werte
8 Stellen	−128 bis −1	0 bis +127	256 (= 2^8)
16 Stellen	−32 768 bis −1	0 bis +32 767	65 536 (= 2^{16})
32 Stellen	−2 147 483 648 bis −1	0 bis +2 147 483 647	4 294 967 296 (= 2^{32})
64 Stellen	−9 223 372 036 854 775 808 bis −1	0 bis +9 223 372 036 854 775 807	18 446 744 073 709 551 616 (= 2^{64})

Zahlenkodierung – Ganzzahlen

- Kodierung positiver und negativer Dualzahlen im Zweier-Komplement (am Beispiel mit 8 Stellen):

Positive Zahlen:

Bereich	Dual	Dezimal
Alle Kombinationen für 7 Stellen von 0000000 = 0 bis 1111111 = 127	00000000	+0
	00000001	+1
	00000010	+2
	…	…
	01111101	+125
	01111110	+126
	01111111	+127

Negative Zahlen:

Bereich	Dual	Dezimal
Alle Kombinationen für 7 Stellen von 1111111 = −1 bis 0000000 = −128	11111111	−1
	11111110	−2
	11111101	−3
	…	…
	10000010	−126
	10000001	−127
	10000000	−128

Zahlenkodierung – Ganzzahlen

- Vorteile dieser Kodierung:
  - Positive Zahlen können durch *ein* Verfahren in negative und wieder zurück gewandelt werden:
    - Bildung des 2er-Komplements (*arithmetische* Negation):
      - Eine positive oder negative Dualzahl wird ziffernweise *einer-negiert* (*logische* Negation NOT für jede Dualstelle);
      - auf das Negationsergebnis wird insgesamt noch 1 aufaddiert.
    - Beispiele:

Operation	Dezimal	Dualzahl	Negation	Addition 1	Ergebnis
+ nach –	+42	00101010	11010101	11010110	–42
– nach +	–42	11010110	00101001	00101010	+42

Zahlenkodierung – Ganzzahlen

- Was würde passieren, wenn man statt des asymmetrischen Zweierkomplements eine symmetrische Darstellung positiver und negativer Ganzzahlen nutzen würde?

Darstellung	asymmetrisch (Zweierkompl.)	symmetrisch (?)
negativer Bereich	…	…
	$11111101_2 = -3_{10}$	$10000010_2 = -2_{10}$
	$11111110_2 = -2_{10}$	$10000001_2 = -1_{10}$
	$11111111_2 = -1_{10}$	$10000000_2 = -0_{10}$
positiver Bereich	$00000000_2 = +0_{10}$	$00000000_2 = +0_{10}$
	$00000001_2 = +1_{10}$	$00000001_2 = +1_{10}$
	$00000010_2 = +2_{10}$	$00000010_2 = +2_{10}$
	…	…

- doppelte Kodierung der 0 als positive und negative Null (sinnvoll?);
- nur 255 statt 256 darstellbare Werte bei 8-Bit-Kodierung.

Zahlenkodierung – Bruchzahlen

- Kodierung von Bruchzahlen (jeder Größenordnung) in Exponenten-Darstellung:
  - Auch Bruchzahlen werden mit einer festen Größe von 32, 64 oder mehr Stellen dargestellt/kodiert.
  - Die externe Darstellung erfolgt dabei in *normalisierter* Form (wissenschaftliche Darstellung wie auf Taschenrechner):

±	Vorkomma-Anteil	Nachkomma-Anteil	Exponent	Zahlenwert
+	3.	141592654	$\bullet 10^{0}$	+3.141592654
–	1.	048576	$\bullet 10^{6}$	–1048576
+	6.	25	$\bullet 10^{-2}$	+0.0625
–	9.	8765	$\bullet 10^{-5}$	–0.000098765

Zahlenkodierung – Bruchzahlen

- Die normalisierte Darstellung besteht aus drei Komponenten:
  - Vorzeichen (VZ) der gesamten Zahl (±);
  - Ziffernfolge (Mantisse):
    - *ein*stelliger Vorkommateil, d. h. genau eine Vorkomma-Ziffernstelle (im Dezimalsystem die Ziffern 0 bis 9, allgemein 0 bis Basis−1; 0 nur dann, wenn gesamte darzustellende Zahl = 0);
    - mehrstelliger Nachkommateil (je nach gewünschter Genauigkeit mehr oder weniger Nachkomma-Ziffern).
  - *Ein* Exponent 10^N (allgemein Basis^N) für die ganze Zahl, mit dem die Mantisse multipliziert werden muss, um den eigentlichen Zahlenwert zu erhalten.

Zahlenkodierung – Bruchzahlen

- Interne Kodierung einer normalisierten Zahl mit 32 Dualstellen (einfache Genauigkeit):

$VZ_{Mantisse}$	Exponent+127	Mantisse mit den Stellenwerten 2^0 bis 2^{-23}
±	X X X X X X X X	1 Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y
1	8	23

  - Die rote 1 mit dem Stellenwert 2^0 ist konstant und muss daher nicht mitgespeichert werden; alle anderen 23 Stellen Y mit den Stellenwerten 2^{-1} bis 2^{-23} sind variabel.
  - Der weiß hinterlegte Exponent der Mantisse zur Basis 2 ist eine 8-stellige Dualzahl, die den Exponentenwert zuzüglich des konstanten Wertes 127 enthält (Verschiebekonstante).

Zahlenkodierung – Bruchzahlen

Beispiel:

Kodierung der Zahl $-11.625_{10} = -1011.101_2 = -1.011101 \bullet 10^{11}{}_2$:

$VZ_{Mantisse}$	Exponent+127	Mantisse mit den Stellenwerten 2^0 bis 2^{-23}
–	1 0 0 0 0 0 1 0	1 0 1 1 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

- Der korrekte/korrigierte Exponent zur Basis 2 errechnet sich aus $10000010_2 = 130_{10} - 127_{10}$ (Verschiebekonstante) $= 3_{10}$.
- Die Mantisse $1.011101_2 = 1.453125_{10}$ muss mit dem korrigierten Exponenten $10^{11}{}_2 = 2^3{}_{10} = 8_{10}$ multipliziert werden: $1.453125_{10} \bullet 8_{10} = 11.625_{10}$.
- Abschließend ist noch das Vorzeichen VZ zu berücksichtigen: -11.625_{10}.

Zahlenkodierung – Bruchzahlen

- Für spezielle 'Zahlen'-Werte existieren eigene Kodierungen (hier nicht wiedergegeben):
  - Null: eigene Darstellung für die 0 (Mantisse aus lauter 0en, Exponent aus lauter 0en);
  - INF (±Unendlich): eigene Darstellungen für $-\infty$ und $+\infty$ (z. B. aus Berechnungen wie $1 \div 0$ oder $-1 \div 0$);
  - NaN ('Not a Number'): eigene Darstellung für (sinnlose) Ergebniswerte von Berechnungen wie $0 \div 0$, $0 \cdot \infty$, $\infty - \infty$, $\sqrt{-1}$ (Letzteres undefiniert nur in der Menge der reellen Zahlen) usw.

Zahlenkodierung – Bruchzahlen

- Für verschiedene Anwendungen werden unterschiedlich genaue Gleitpunktzahlen benötigt:

Genauigkeit (Anzahl Stellen)	Einfach ('Single')	Doppelt ('Double')	Erweitert ('Extended')
Gesamtstellen	32	64	80
Vorzeichen	1	1	1
Exponent	8	11	15
Mantisse	23 (+1)	52 (+1)	64 (+1)
Verschiebe-Konstante	+127 (2^7-1)	+1023 ($2^{10}-1$)	+16383 ($2^{14}-1$)
Dezimalstellen **Wertebereich**	ca. 7–8 ca. $\pm 10^{\pm 38}$	ca. 15–16 ca. $\pm 10^{\pm 308}$	ca. 19–20 ca. $\pm 10^{\pm 4932}$

Übungsaufgaben

- Bilden Sie das Zweier-Komplement folgender Zahlen im jeweils vorgegebenen Zahlensystem:

Zahl (8 Bits)	Rechnung
01101001_2	
2222_4	
96_{16}	

Zahl (8 Bits)	Rechnung
10010110_2	
0000_4	
FF_{16}	

Übungsaufgaben

- Stellen Sie folgende Zahlen im jeweiligen Zahlensystem in normalisierter Schreibweise dar:

Zahl	Normalisierung (ohne Zweierkomplement oder Verschiebekonstante im Exponenten)
16777216_{10}	
0.000256_{10}	
299792.458_{10}	
1010.0101_2	
32103210.3210_4	
$AB.BAABBA_{12}$	
$CAFE.AFFE_{16}$	