Systemorientierte Informatik 1

Prof. Manfred Schimmler
Lehrstuhl für Technische Informatik
Institut für Informatik
Christian Albrechts Universität zu Kiel
Tel.: 8804480
E-Mail: masch@informatik.uni-kiel.de

Prof. Dr. Manfred Schimmler

..... darf ich mich vorstellen? ......

1980-1981	Siemens AG Berlin
1981-1985	Assistent, CAU Kiel
1985-1987	Research Fellow University Aarhus
1987-1989	Research Fellow ANU Canberra
1989	Gründung ISATEC GmbH
1989-1996	Geschäftsführer, ISATEC GmbH, Kiel
1996-1997	Fachhochschulprofessor, FH Stralsund
1997-2004	Professor, TU Braunschweig
2004-2014	Professor, CAU Kiel
2009:	Gründung SciEngines GmbH
2011:	Gründung Isavision GmbH
2014:	Gründung Innovative Prognostic Services GmbH

Material zur Vorlesung

1. Skript: Das Skript ändert sich ständig. Daher sind noch einige Fehler drin. Diese werden im Laufe des Semesters gefunden und korrigiert. Für jeden inhaltlichen Fehler, der mir (erstmalig) von Ihnen gezeigt wird, spendiere ich einen Schokoriegel.
2. Bögen mit Übungsaufgaben stehen normalerweise Mittwoch ab 20 Uhr im Web. Diese Aufgaben sind bis zum 2. darauf folgenden Montag bis 8.00 im Schrein im Haus Hermann-Rodewald-Str.3 abzugeben. Sie müssen mit dem Namen und der Nummer der Übungsgruppe versehen sein. Mehrere Blätter müssen zusammengeheftet oder -geklammert sein.
3. Bücher und www-Links: Werden in der Vorlesung angegeben

http://www.techinf.informatik.uni-kiel.de/de/lehre/vorlesungen/inf-digisys-digitale-systeme-vorlesung

Übungsbetrieb

1. Jede Woche muss ein Bogen mit Übungsaufgaben selbst bearbeitet werden (nicht abgeschrieben). Die Lösungen werden mit bis zu 100 Punkten bewertet.
2. Die Übungsserien sind in festen Zweiergruppen abzugeben. Beide Übungspartner sollen dabei an jeder Aufgabe gleiche Anteile haben.
3. Zweimal im Semester wird man nach der Übung gebeten, seine Übungsaufgaben zu erklären und ggf. neue zu rechnen. Dieses sogenannte Kolloquium wird jeweils mit maximal 300 Punkten bewertet.
4. Wenn man in den Übungen eine Aufgabe an der Tafel vorrechnet, bekommt man 20 Punkte extra. Für gute Mitarbeit gibt es pro Übung 10 Punkte.

Klausurvoraussetzungen

Am Ende des Semesters wird eine Klausur geschrieben. Voraussetzungen für die Teilnahme sind:

1. Mindestens 900 Punkte aus dem Übungsbetrieb.
2. Maximal 2-maliges Fehlen ohne ärztliches Attest in den Übungen.
3. Anmeldung zur Klausur über die StudiDB, voraussichtlich Ende Dezember/Anfang Januar.

Klausurhilfsmittel

- Erlaubte Hilfsmittel bei der Klausur sind : Stifte, eine handgeschriebene Formelsammlung der Größe Din A4, beidseitig beschrieben.

- Nicht erlaubt sind: Taschenrechner, Skripte, Bücher, Handys, PDAs, Rechner, Tabellen, gedruckte Formelsammlungen, Lupen, Zettel des Nachbarn, …

Gesamtbewertung

- Die Übungspunkte gehen folgendermaßen in die Klausur ein: Hat man U% der 1800 Übungspunkte erreicht und K% der normal erreichbaren Klausurpunkte, erhält man 0,2*min(U%,K%) Klausurpunkte zusätzlich.
- Erreicht man also in der Klausur allein 80% der normal möglichen Klausurpunkte, kann man zusammen mit den 80% Übungspunkten (oder mehr) auf insgesamt 96% der Klausurpunkte kommen und so die Gesamtnote deutlich verbessern.

Accountvergabe

- Jeder Student bekommt einen Account mit dem er sich in der StudiDB der Informatik für Lehrveranstaltungen anmelden kann.
- Die Accounts können von jedem (internetfähigen) Rechner, oder auch auf den Rechnern im Foyer des Unihochhauses auf folgender Website freigeschaltet werden:
  https://sodom.informatik.uni-kiel.de:8486/initveranstaltung/anmeldung
- Achten Sie bei der Anmeldung darauf, die Daten genau so wie auf dem Leporello einzugeben
- Bis der Account nutzbar ist, vergeht höchstens ein Werktag.

Accountnutzung

- Auf die StudiDB kann jederzeit und von überall zugegriffen werden: https://sodom.informatik.uni-kiel.de:8484/studierende/Login.html
- Melden Sie sich bitte schnellstmöglich bei allen Lehrveranstaltungen an, die Sie besuchen.
- Das heißt in diesem Fall konkret für das Modul G1.2 (080122) „Übungen zu: Systemorientierte Informatik I – Digitale Systeme“
  Sie müssen sich dann für eine Übungsgruppe entscheiden und dieser beitreten – achten Sie dabei auf die Anzahl freier Plätze!

Termin für die StudiDB-Eintragung

- Wichtiger Hinweis: Bitte unbedingt bis heute, 27.10.14, in die Übungsgruppen eintragen! Sollte es dann noch zu viele offene Plätze in den Übungsgruppen geben, müssen wir dann aus Kostengründen beginnen, Übungsgruppen zusammenzulegen. Dabei bleiben Übungsgruppen mit vielen Teilnehmern eher bestehen, als solche mit wenigen. Wenn Sie sich also jetzt in Ihre Wunschübungsgruppe eintragen, erhöhen Sie die Wahrscheinlichkeit, dass diese auch stattfindet.

Themenübersicht

1. Zahlendarstellung in Computern
2. Grundlagen digitaler Schaltungen
3. Der MOS-Transistor
4. CMOS Technologie und CMOS-Gatter
5. Schaltnetze
6. Computer Arithmetik
7. Flip-Flops
8. Schaltwerke
9. Arithmetisch Logische Einheit
10. Speicher
11. Der DLX-Prozessor
12. Assembler.

1. Zahlendarstellung in Computern

Literatur:

Waldschmidt, K.: Schaltungen der Datenverarbeitung, Teubner, 1980, ISBN 3-519-06108-2

Klar, R.: Digitale Rechenautomaten, de Gruyter, 1976, ISBN 3-110-04194-4

Leonhard, E.: Grundlagen der Digitaltechnik, Hanser Verlag, 1976, ISBN 3-446-12158-7

Polyadische Darstellung von Zahlen

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

$$= b_{N-1}B^{N-1} + b_{N-2}B^{N-2} + \ldots + b_1B^1 + b_0B^0$$

heißt B-adische Darstellung von n

$b_i \in \{0,1,\ldots,B-1\}$ heißen Ziffern

Polyadische Darstellung von Zahlen

Kurzschreibweisen für B-adische Darstellung:

$$(b_{N-1}b_{N-2}...b_1b_0)_B$$

oder, wenn klar ist, um welche Basis es sich handelt:

$$b_{N-1}b_{N-2}...b_1b_0$$

Binär 2-adisch	Ternär 3-adisch	Oktal 8-adisch	Dezimal 10-adisch	Hexadezimal 16-adisch
0	0	0	0	0
1	1	1	1	1
10	2	2	2	2
11	10	3	3	3
100	11	4	4	4
101	12	5	5	5
110	20	6	6	6
111	21	7	7	7
1000	22	10	8	8
1001	100	11	9	9
1010	101	12	10	A
1011	102	13	11	B
1100	110	14	12	C
1101	111	15	13	D
1110	112	16	14	E
1111	120	17	15	F
10000	121	20	16	10

Satz:

Die N-stellige B-adische Darstellung ermöglicht es, jede ganze Zahl aus $\{0,1,...,B^N-1\}$ auf genau eine Weise darzustellen.

Beweis:
Jede Zahl kann dargestellt werden

- auf mindestens eine Weise (vollständige Induktion nach N)
- auf höchstens eine Weise (Abzählen der Kombination der Ziffern)

Tipp: Seien Sie misstrauisch, wenn Ihnen die Basis des jeweiligen Zahlensystems nicht explizit genannt wird:

Satz:
Man kann die Studierenden in diesem Hörsaal bereits am Ende dieser Vorlesung in 10 Kategorien einteilen: Solche, die das Binärsystem verstanden haben und solche, für die das nicht gilt.

Hornerschema

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

$$= b_{N-1}B^{N-1} + b_{N-2}B^{N-2} + \ldots + b_1 B^1 + b_0 B^0$$

$$= ((..(b_{N-1}B + b_{N-2})*B + b_{N-3})*B \ldots + b_1)*B + b_0$$

Umwandlung von Zahlen zwischen polyadischen Systemen
1. Verfahren der wiederholten Division mit Rest:

$$
\begin{array}{llcll}
n : & B & = & q_1 & \text{Rest} \quad b_0 \\
q_1 : & B & = & q_2 & \text{Rest} \quad b_1 \\
q_2 : & B & = & q_3 & \text{Rest} \quad b_2 \\
q_3 : & B & = & q_4 & \text{Rest} \quad b_3 \\
q_4 : & B & = & q_5 & \text{Rest} \quad b_4
\end{array}
$$

. . . .

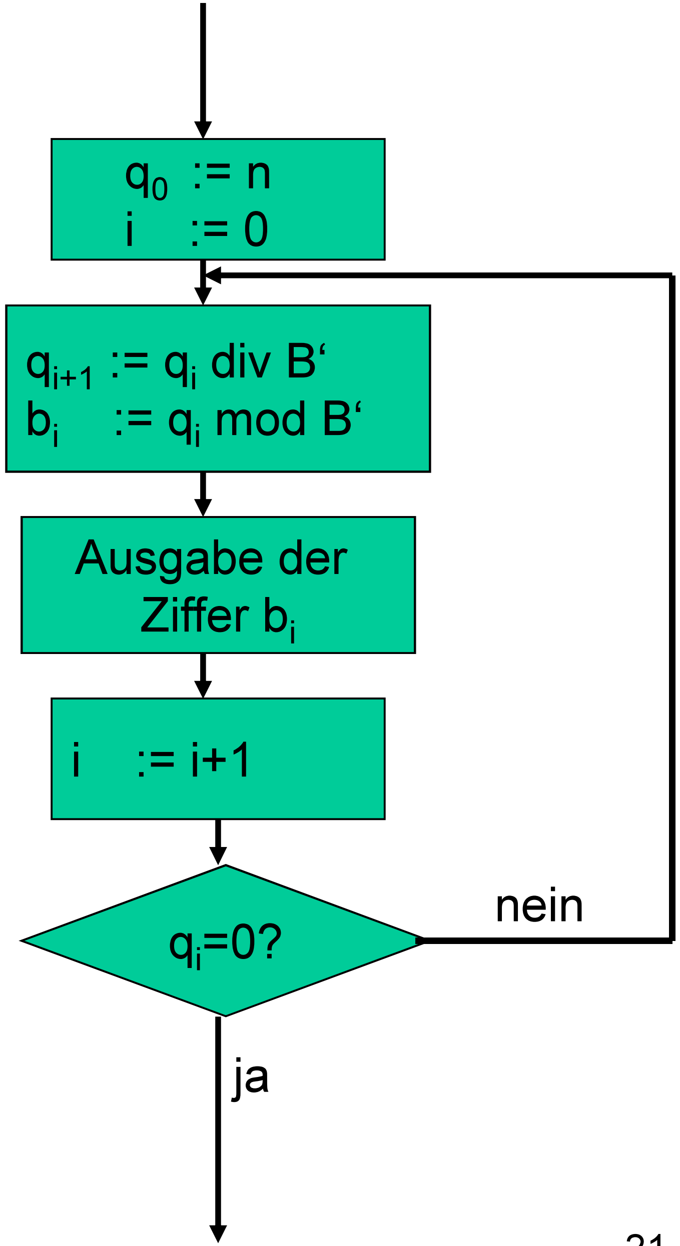
$$
\begin{array}{llcll}
q_{N-2} : & B & = & q_{N-1} & \text{Rest} \quad b_{N-2} \\
q_{N-1} : & B & = & 0 & \text{Rest} \quad b_{N-1}
\end{array}
$$

Umwandlung von Zahlen zwischen polyadischen Systemen

Rechnen im Quellsystem:
$(x)_B \rightarrow (y)_{B'}$

1. Stelle die Basis B‘ des Zielsystems im Quellsystem dar.
2. $q_0=n$
3. Wiederhole für aufsteigendes i:
   $q_{i+1} = q_i$ div B‘; $r_i = q_i$ mod B‘
   bis $q_{i+1}=0$.
4. Die r_i sind die B‘-adische Darstellung von y

Umwandlung von Zahlen aus dem B-adischen ins B‘-adische Zahlensystem durch Rechnen im Quellsystem (B-adisches Zahlensystem)



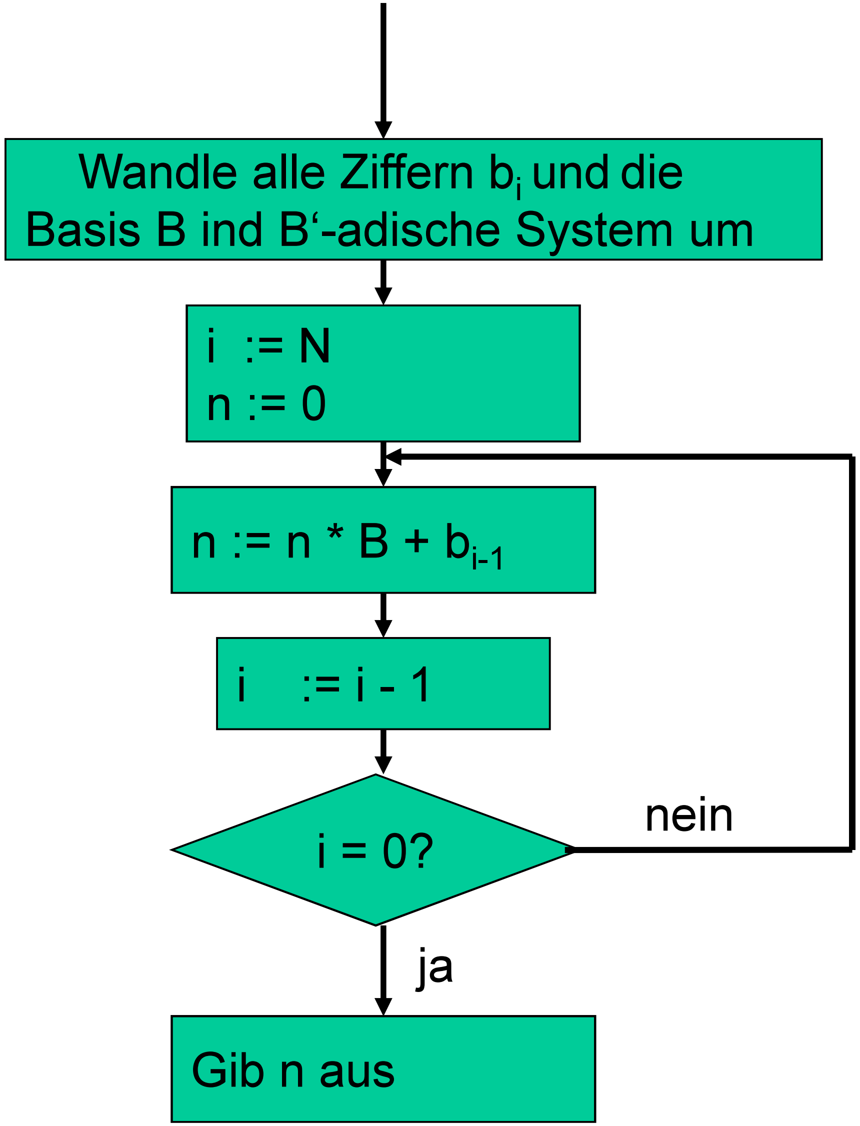
Umwandlung von Zahlen zwischen polyadischen Systemen
2. Abarbeitung des Hornerschemas von links nach rechts:

$$((..(b_{N-1}B + b_{N-2})*B + b_{N-3})*B ... + b_1)*B + b_0$$

Rechnen im Zielsystem:

1. Umwandlung aller b_i ins B‘-adische System
2. Umwandlung von B ins B‘-adische System
3. Ausrechnen im B‘-adischen System

Umwandlung von Zahlen aus dem B-adischen ins B‘-adische Zahlensystem durch Rechnen im Zielsystem (B‘-adisches Zahlensystem)



Umwandlung von Zahlen zwischen polyadischen Systemen, deren Basen Zweierpotenzen sind

1. Umwandlung aller Ziffern ins Binärsystem
2. Wandle die Quellzahl ziffernweise in eine Binärzahl um
3. Fasse geeignete Bits zusammen (LSB-first) für jeweils eine Ziffer im Zielsystem.
4. Erzeuge so die Ziffern im Zielsystem.

(LSB least significant bit, also LSB-first heißt: man beginnt mit dem geringwertigsten Bit)

Definition:

Sei n eine natürliche Zahl, dargestellt als N-stellige B-adische Zahl. Das B-Komplement von n ist die N-stellige B-adische Zahl gebildet aus den letzten N Ziffern von B^N-n. Das B-Komplement wird interpretiert als -n

Umwandlung einer Zahl ins Negative (B-Komplement):

$$(b_{N-1}b_{N-2}...b_1b_0)_B$$

Jede Ziffer b_i wird ersetzt durch die Ziffer $(B-1-b_i)$.
Auf die so entstehende Zahl wird 1 addiert.

4-stellige Zweierkomplement-zahlen

Dezimal 10-adisch	Binär 2-adisch
-8	1000
-7	1001
-6	1010
-5	1011
-4	1100
-3	1101
-2	1110
-1	1111
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111

Darstellbarer Bereich N-stelliger B-adischer Zahlen im B-Komplement für gerades B

$$\{-(B/2)B^{N-1},....,+(B/2)B^{N-1}-1\}$$

Genau die Zahlen, die mit einer Ziffer $\geq B/2$ beginnen, sind negativ.

Satz:

Genau dann ist bei Addition zweier N-stelliger 2-adischer Zahlen das Ergebnis wieder im (mit N Stellen) darstellbaren Bereich, wenn bei der Summe nach der Addition die Vorzeichenstelle (Stelle N-1) mit der Sicherungsstelle (Stelle N) übereinstimmt.

Definition:

Ein Restklassensystem ist ein Zahlensystem, das durch eine Menge von Moduli bestimmt ist. Diese Moduli sind natürliche Zahlen, die paarweise teilerfremd sind. Sei $\{p_1,p_2,...,p_k\}$ die Menge der Moduli und P das Produkt aller p_i. Dann sind im zugehörigen Restklassensystem alle Zahlen von 0 bis P-1 eindeutig durch die Reste bei der Division durch die p_i charakterisisert.

Arithmetik in Restklassensystemen

$(x_1,x_2,...,x_k)+(y_1,y_2,...,y_k) =$
$((x_1+y_1)\mathrm{mod}\ p_1,(x_2+y_2)\mathrm{mod}\ p_2,...,(x_k+y_k)\mathrm{mod}\ p_k)$

$(x_1,x_2,...,x_k)-(y_1,y_2,...,y_k) =$
$((x_1-y_1)\mathrm{mod}\ p_1,(x_2-y_2)\mathrm{mod}\ p_2,...,(x_k-y_k)\mathrm{mod}\ p_k)$

$(x_1,x_2,...,x_k)*(y_1,y_2,...,y_k) =$
$((x_1*y_1)\mathrm{mod}\ p_1,(x_2*y_2)\mathrm{mod}\ p_2,...,(x_k*y_k)\mathrm{mod}\ p_k)$

Vorteil: Arithmetik ohne Überträge
Nachteil: Division und Rückkonvertierung schwierig

Definition:

Ein Zahlensystem, bei dem die Zahlendarstellung nicht eindeutig ist, heißt redundantes Zahlensystem.

Addition im redundanten Zahlensystem

1. Addiere jede Stelle zu einer temporären Ergebnisziffer $t_i \in \{-2,-1,0,1,2\}$

2. Wandle t_i in zwei temporäre Ziffern c_{i+1} und d_i nach folgender Vorschrift um:

   $c_0 = 0$
   wenn $t_i = 2$ dann $c_{i+1}=+1$ und $d_i=0$
   wenn $t_i = 0$ dann $c_{i+1}=0$ und $d_i=0$
   wenn $t_i = -2$ dann $c_{i+1}=-1$ und $d_i=0$
   wenn $t_i = +1$ und $t_{i-1}<+1$ dann $c_{i+1}=0$ und $d_i=+1$
   wenn $t_i = +1$ und $t_{i-1}\geq+1$ dann $c_{i+1}=+1$ und $d_i=-1$
   wenn $t_i = -1$ und $t_{i-1}>-1$ dann $c_{i+1}=0$ und $d_i=-1$
   wenn $t_i = -1$ und $t_{i-1}\leq-1$ dann $c_{i+1}=-1$ und $d_i=+1$

3. Addiere schließlich an jeder Stelle i c_i und d_i zu s_i.

Darstellung rationaler Zahlen im Festkommaformat

$$n = \sum_{i=-M}^{N-1} b_i * B^i$$

$$= b_{N-1}B^{N-1} + b_{N-2}B^{N-2} + \ldots + b_1B^1 + b_0B^0$$

$$+ b_{-1}B^{-1} + b_{-2}B^{-2} + \ldots + b_{-M+1}B^{-M+1} + b_{-M}B^{-M}$$

Darstellbarer Bereich (bei geradem B):

$$[-(B/2)*B^{N-1} .. +(B/2)*B^{N-1} - B^{-M}]$$

Hornerschema für Brüche

$$n = \sum_{i=-M}^{-1} b_i * B^i$$

$$= b_{-1}B^{-1} + b_{-2}B^{-2} + \ldots + b_{-M+1}B^{-M+1} + b_{-M}B^{-M}$$

$$= ((..(b_{-M}B^{-1} + b_{-M+1})*B^{-1} + b_{-M+2})*B^{-1} \ldots + b_{-1})*B^{-1}$$

Umwandlung von Zahlen zwischen polyadischen Systemen

1. Verfahren der wiederholten Multiplikation mit Abschneiden:

$$n_1 := n \cdot B \qquad b_{-1} := \lfloor n_1 \rfloor \qquad n'_1 := n_1 - \lfloor n_1 \rfloor$$

$$n_2 := n'_1 \cdot B \qquad b_{-2} := \lfloor n_2 \rfloor \qquad n'_2 := n_2 - \lfloor n_2 \rfloor$$

$$n_3 := n'_2 \cdot B \qquad b_{-3} := \lfloor n_3 \rfloor \qquad n'_3 := n_3 - \lfloor n_3 \rfloor$$

....

$$n_M := n'_{M-1} \cdot B \qquad b_{-M} := \lfloor n_M \rfloor \qquad n'_M := n_M - \lfloor n_M \rfloor$$

Gleitkommazahlen
(floating point numbers)

n = V * 0,Mantisse * 2^{Exponent}

Dabei ist V = +1 wenn das Vorzeichen + ist
und V = -1, wenn das Vorzeichen – ist.

Der Bereich darstellbarer Zahlen bei m
Mantissenbits und e Exponentenbits ist

$$\left[\ -\left(1-2^{-m}\right)\cdot 2^{2^{e-1}-1} \bullet\bullet\bullet +\left(1-2^{-m}\right)\cdot 2^{2^{e-1}-1}\ \right]$$

Gleitkommazahlen haben den Vorteil, dass sie einen viel größeren Zahlenbereich abdecken als gleichlange Festkommazahlen.

Ferner bieten Sie in der Nähe der 0 eine wesentlich höhere Genauigkeit.

Multiplikation von Gleitkommazahlen

$N_1 = V_1 * 0{,}M_1 * 2^{E1}$, $N_2 = V_2 * 0{,}M_2 * 2^{E2}$

$N_1 * N_2 = (V_1 * V_2) * 0{,}(M_1 * M_2) * 2^{E1+E2}$

Addition von Gleitkommazahlen

$N_1 = V_1 * 0{,}M_1 * 2^{E1}$, $N_2 = V_2 * 0{,}M_2 * 2^{E2}$

1. Exponentendifferenz berechnen (z.B. E1>E2). d=E1-E2
2. Verschieben der Mantisse M_2 um d Stellen nach rechts. $M'_2=M_2 >> d$
3. Addition der Mantissen M_1 und M'_2
4. Berechnung des Vorzeichens des Ergebnisses
5. Normalisierung

$N_1+N_2=(V)*0{,}(M_1+M'_2)*2^{E1}$

IEEE 754 Format 32-Bit (float, single)

1 Vorzeichenbit
8 Exponentenbits (MSB first)
23 Mantissenbits (MSB first)

Der Wert w eine solchen Zahl berechnet sich als:

$w = (-1)^V*(1,M)*2^{E-127}$,	falls E>0 und E<255
$w = (-1)^V*(0,M)*2^{-126}$,	falls E=0 und M≠0
$w = (-1)^V*0$,	falls E=0 und M=0
$w = (-1)^V*$Infinity (∞),	falls E=255 und M=0
w = NaN (Not a number),	falls E=255 und M≠0

Darstellbarer Bereich ca $[-10^{41}..+10^{41}]$

IEEE 754 Format 64-Bit (double)

1 Vorzeichenbit
11 Exponentenbits (MSB first)
52 Mantissenbits (MSB first)

Der Wert w eine solchen Zahl berechnet sich als:

$w = (-1)^V * (1,M) * 2^{E-1023}$,	falls $E>0$ und $E<2047$
$w = (-1)^V * (0,M) * 2^{-1022}$,	falls $E=0$ und $M \neq 0$
$w = (-1)^V * 0$,	falls $E=0$ und $M=0$
$w = (-1)^V * \text{Infinity} (\infty)$,	falls $E=2047$ und $M=0$
w = NaN (Not a number),	falls $E=2047$ und $M \neq 0$

Darstellbarer Bereich ca $[-10^{300}..+10^{300}]$

IEEE 754 Format 80-Bit (extended)

1 Vorzeichenbit
15 Exponentenbits (MSB first)
64 Mantissenbits (MSB first)

Der Wert w eine solchen Zahl berechnet sich als:

$w = (-1)^V * (0,M) * 2^{E-16383}$,	falls $E>0$ und $E<32767$
$w = (-1)^V *$ Infinity (∞),	falls $E=32767$ und $M=0$
w = NaN (Not a number),	falls $E=32767$ und $M \neq 0$

Darstellbarer Bereich ca $[-10^{5000}..+10^{5000}]$

Codierung der Dezimalziffern

Dezimal-ziffer	Binär	Aiken	3-Excess	2aus5
0	0000	0000	0011	11000
1	0001	0001	0100	00011
2	0010	0010	0101	00101
3	0011	0011	0110	00110
4	0100	0100	0111	01001
5	0101	1011	1000	01010
6	0110	1100	1001	01100
7	0111	1101	1010	10001
8	1000	1110	1011	10010
9	1001	1111	1100	10100
Gewichte	8421	2421	keine	74210

EBCDIC (Extenden Binary Coded Decimal Interchange Code)

		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
		0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
0	0000																
1	0001																
2	0010																
3	0011																
4	0100	blank										§	.	<	(	+	|
5	0101	&										!	$	•	)	;	
6	0110	-	/									^	,	%		>	?
7	0111											:	#	@	‘	*	“
8	1000		a	b	c	d	e	f	g	h	i						
9	1001		j	k	l	m	n	o	p	q	r						
A	1010			s	t	u	v	w	x	y	z						
B	1011																
C	1100		A	B	C	D	E	F	G	H	I						
D	1101		J	K	L	M	N	O	P	Q	R						
E	1110			S	T	U	V	W	X	Y	Z						
F	1111	0	1	2	3	4	5	6	7	8	9						

ASCII (American Standard Code for Information Interchange)

		000	001	010	011	100	101	110	111
0	0000	NUL	DLE	SP	0	@	P	‘	p
1	0001	SOH	DC1	!	1	A	Q	a	q
2	0010	STX	DC2	“	2	B	R	b	r
3	0011	ETX	DC3	#	3	C	S	c	s
4	0100	EOT	DC4	$	4	D	T	d	t
5	0101	ENQ	NAK	%	5	E	U	e	u
6	0110	ACK	SYN	&	6	F	V	f	v
7	0111	BEL	ETB	´	7	G	W	g	w
8	1000	BS	CAN	(	8	H	X	h	x
9	1001	SKIP	EM	)	9	I	Y	i	y
A	1010	LF	SUB	*	:	J	Z	j	z
B	1011	VT	ESC	+	;	K	[	k	{
C	1100	FF	FS	,	<	L	\	l	|
D	1101	CR	GS	-	=	M	]	m	}
E	1110	SO	HOME	.	>	N	^	n	~
F	1111	SI	NL	/	?	O	_	o	DEL