Grundlagen der Informatik

- Computerinterne Informationsdarstellung Forts. -

Prof. Dr. Klaus Volbert

Hochschule für angewandte Wissenschaften
Fakultät Informatik und Mathematik

Wintersemester 2010/11
Regensburg, 19. Oktober 2010

Umrechnung Zahl → 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:

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

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

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

Umrechnung Dezimalzahl → 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:
  - $(11{,}5)_{10} \rightarrow (?,?)_2$
  - $(27{,}1)_{10} \rightarrow (?,?)_2$

Umrechnung Dezimal → Dual ohne Verlust

HOCHSCHULE REGENSBURG UNIVERSITY OF APPLIED SCIENCES

- Satz: Ein gekürzter Bruch hat genau dann eine nicht periodische B-adische Darstellung, wenn alle Primfaktoren seines Nenners auch Primfaktoren von B sind
- Bemerkung: Für eine nicht periodische Darstellung im Dezimalsystem muss der gekürzte Nenner also ein Produkt der Zahlen 2 und 5 sein
- Beispiele:
  - Dualsystem hat Basis 2, Primfaktor ist 2
  - $(0{,}1)_{10} = \frac{1}{10}$, d.h. Nenner hat Primfaktoren 2 und 5

  $\Rightarrow$ Periodische Darstellung, da 5 nicht Primfaktor von 2

  - $(0{,}75)_{10} = \frac{3}{4}$, d.h. Nenner hat Primfaktor 2

  $\Rightarrow$ Keine Periodische Darstellung, da alle Primfaktoren des gekürzten Nenners auch Primfaktoren von 2 sind

Rechenregeln im Dualsystem

- Addition

$$\begin{array}{ll} (1101100)_2 & = (108)_{10} \\ (1011011)_2 & = (91)_{10} \\ 1\ 1\ 1\ 1 & \\ \hline (11000111)_2 & = (199)_{10} \end{array}$$

- Subtraktion und Negative Zahlen
  - Darstellung durch Betrag und Vorzeichen
  - Umsetzung im Rechner erfordert gesonderte Vorzeichenbehandlung
  - Weiterer Nachteil: Rechenwerk mit Addierer und Subtrahierer
  - Alternative: Zurückführung der Subtraktion auf die Addition
- Komplementbildung im Dualsystem
  - Einer-Komplement (B-1-Komplement mit Basis B=2)
  - Zweier-Komplement (B-Komplement mit Basis B=2)
- Anmerkung: Komplementbildung lässt sich verallgemeinern

Einer-Komplement

- Vorzeichenbit
  - Falls 1. Bit „von links“ (hohes Bit) gleich 1 ist: negative Zahl
- Komplementbildung einer negativen Zahl
  - Jedes weitere Bit wird invertiert (umgedreht)
- Beispiel (5 Bit)
  - $(9)_{10} = (01001)_2$, d.h. $-9 = 10110 \quad (= 11111 - 01001 = 2^5 - 1 - 9)$
    (Wert invertiert, Vorzeichenbit gesetzt)
- Addition
  - Zusatz: Kommt es zu einem Überlauf, muss hierfür 1 addiert werden
- Beispiel (5 Bit)
  - $(14)_{10} - (9)_{10} =$

$$\begin{array}{rl} & 01110 \\ + & 10110 \\ & \overset{1\ 1\ 1\ 1}{} \\ \hline & 00100 \\ \text{Überlauf:} & \quad 1 \\ \hline & 00101 = (5)_{10} \end{array}$$

Was passiert, wenn man
zwei negative Zahlen addiert?

Zweier-Komplement

- Darstellung
  - Bilde Einer-Komplement und addiere 1 (nur bei negativen Zahlen)
- Beispiel (5 Bit)
  - $(9)_{10} = (01001)_2$, d.h. $-9 = 10111 \quad (= 11111 - 01001 + 1 = 2^5 - 9)$
    (Wert invertiert, Vorzeichenbit gesetzt, 1 Addiert)
- Anmerkungen zur Addition
  - Keine zusätzliche Addition bei Überlauf erforderlich
  - Technisch einfacher zu realisieren
- Beispiel (5 Bit)
  - $(14)_{10} - (9)_{10} =$

$$\begin{array}{r} 01110 \\ +\ 10111 \\ \hline {\scriptstyle 1\,1\,1\,1}\ 00100 \\ \hline 00101 = (5)_{10} \end{array}$$

Exzess-Darstellung

- Der Zahlbereich wird durch Addition einer Konstanten (Exzess-Offset) verschoben
- Beispiel:

Art	Darstellung							
	000	001	010	011	100	101	110	111
Exzess-0	0	1	2	3	4	5	6	7
Exzess-1	-1	0	1	2	3	4	5	6
Exzess-2	-2	-1	0	1	2	3	4	5
Exzess-4	-4	-3	-2	-1	0	1	2	3

- Der Offset muss bei Addition und Subtraktion entsprechend mitgeführt werden

Übersicht zur Darstellung von Zahlen

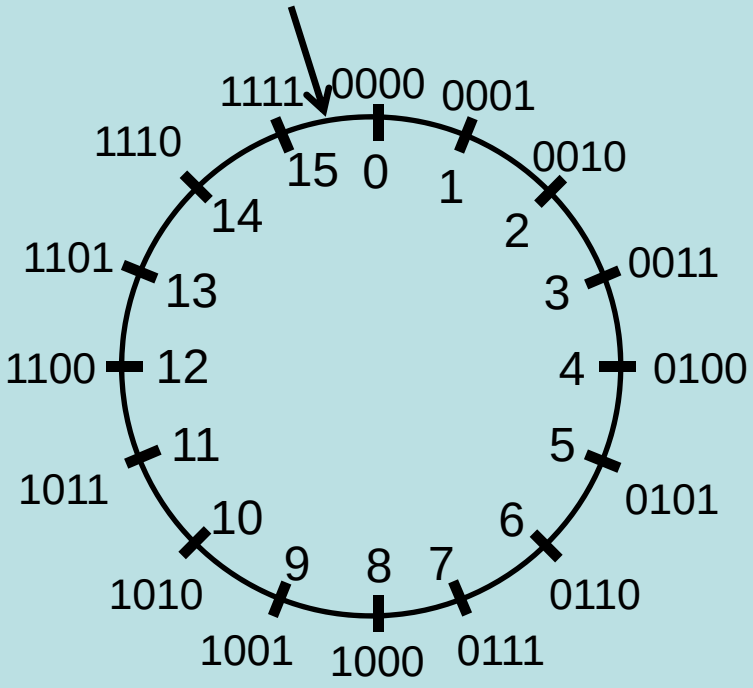
HOCHSCHULE REGENSBURG UNIVERSITY OF APPLIED SCIENCES

Dezimalzahl	Betrag und Vorzeichen	Einer-Komplement	Zweier-Komplement	Exzess-8 Darstellung
-8	-----	------	1000	0000
-7	1111	1000	1001	0001
-6	1110	1001	1010	0010
-5	1101	1010	1011	0011
-4	1100	1011	1100	0100
-3	1011	1100	1101	0101
-2	1010	1101	1110	0110
-1	1001	1110	1111	0111
0	1000,0000	1111, 0000	0000	1000
1	0001	0001	0001	1001
2	0010	0010	0010	1010
3	0011	0011	0011	1011
4	0100	0100	0100	1100
5	0101	0101	0101	1101
6	0110	0110	0110	1110
7	0111	0111	0111	1111

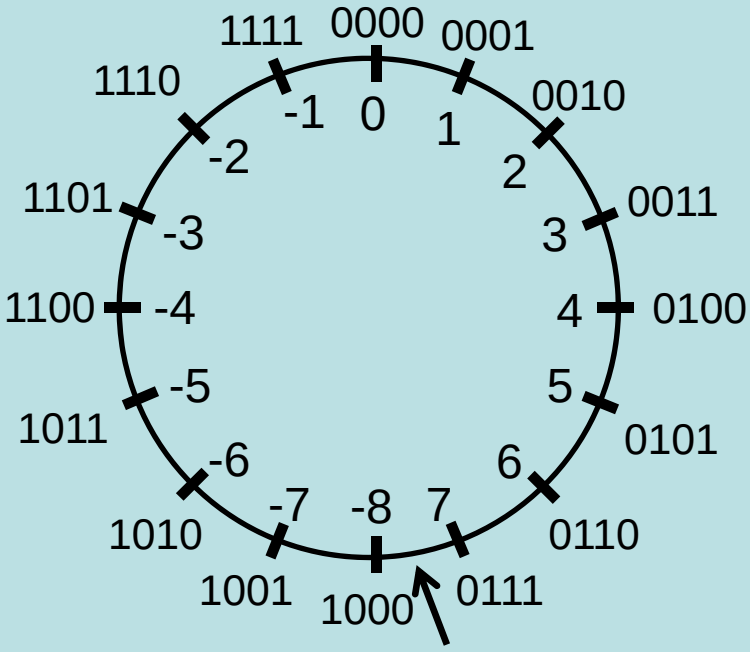
Darstellung durch Zahlenringe

HOCHSCHULE REGENSBURG UNIVERSITY OF APPLIED SCIENCES

4-Bit, vorzeichenlos



4-Bit, Zweier-Komplement



Anmerkungen Multiplikation und Division

- Durchführung mittels wiederholter Addition
- Vereinfachte Durchführung bei Multiplikation mit 2 oder Division durch 2:
  - Division (Verschiebung nach rechts):

$$(200)_{10}/(8)_{10} = (200)_{10}/(2^3)_{10} = (11001\cancel{000})_2 = (25)_{10}$$

  - Multiplikation (Verschiebung nach links):

$$(14)_{10} \cdot (8)_{10} = (14)_{10} \cdot (2^3)_{10} = (01110\ \underline{000})_2 = (112)_{10}$$

Nachteile der Festkommadarstellung

- Feste Anzahl von Bits erlaubt nur einen bestimmten Wertebereich
  - Beispiel:
    16 Bit erlauben $2^{16} = 65536$ Zahlen mit Wertebereich von beispielsweise $-32768 = -2^{15}, \ldots, 32767 = 2^{15} - 1$
- Die Stelle des Kommas ist allgemein festgelegt
- Aus diesen Gründen wird heute überwiegend die Gleitkommadarstellung verwendet

Gleitkommazahlen (Reelle Zahlen)

HOCHSCHULE REGENSBURG UNIVERSITY OF APPLIED SCIENCES

- Idee:
  - Zu jedem Wert wird auch die Stelle des Kommas gespeichert
- Jede reelle Zahl $x \in$ IR kann wie folgt dargestellt werden:

$$x = (-1)^v \cdot m \cdot B^e$$

- v Vorzeichen (0: positiv, 1: negativ)
- m Mantisse
- B Basis des Zahlensystems ($B \in$ IN, $B \geq 2$)
- e Exponent

- Bemerkung
  - Aufgrund von Ungenauigkeiten sollte man Gleitkommazahlen nie auf Gleichheit prüfen

Arithmetik bei Gleitkommazahlen

- Seien folgende Zahlen gegeben

$$x_1 = (-1)^{v_1} \cdot m_1 \cdot B^{e_1} \qquad x_2 = (-1)^{v_2} \cdot m_2 \cdot B^{e_2}$$

- Dann gelten:

$$x_1 + x_2 = ((-1)^{v_1} \cdot m_1 + (-1)^{v_2} \cdot m_2 \cdot B^{e_2 - e_1}) \cdot B^{e_1}$$
$$x_1 - x_2 = ((-1)^{v_1} \cdot m_1 - (-1)^{v_2} \cdot m_2 \cdot B^{e_2 - e_1}) \cdot B^{e_1}$$

$$x_1 \cdot x_2 = ((-1)^{v_1 + v_2} \cdot m_1 \cdot m_2) \cdot B^{e_1 + e_2}$$
$$\frac{x_1}{x_2} = ((-1)^{v_1 - v_2} \cdot \frac{m_1}{m_2}) \cdot B^{e_1 - e_2}$$

Darstellung nach IEEE 754 (1982)

- Formel zur Darstellung: $x = (-1)^v \cdot (2^{e-BIAS}) \cdot (1, m_{n-1} \ldots m_0)$
- Normalisierung:
  - Veränderung des Exponenten, so dass der gedachte Dezimal-punkt immer rechts von der 1. Ziffer $\neq 0$ liegt (Ausnahme: 0)
  - 1. Ziffer der Mantisse wird nicht gespeichert

Bits (Anzahl Bits)	einfach (float)	doppelt (double)
Vorzeichen (v)	1	1
Exponent (e)	8	11
Mantisse (m)	23	52
Gesamt	***32 (4 Byte)***	***64 (8 Byte)***
BIAS	127	1023
Bereich für e-BIAS	[-126,127]	[-1022,1023]

Anmerkung: min./max. Exponent für Spezialfälle: NaN, ∞, 0

- Beispiel (einfache Genauigkeit):

$$(17{,}625)_{10} = 16 + 1 + \frac{1}{2} + \frac{1}{8} = (10001{,}101)_2$$

Normalisierung:

$$= (1{,}0001101)_2 \cdot (2^4)_{10}$$

$$= (\underset{1}{0}\underset{8}{10000011}\underset{23}{00011010000000000000000})_{IEEE754}$$

Da für den Exponenten e: $e = 4 + BIAS = (131)_{10} = (10000011)_2$

Einfache Genauigkeit inkl. Sonderfälle

HOCHSCHULE REGENSBURG UNIVERSITY OF APPLIED SCIENCES

- Einfache Genauigkeit
  - 32 Bit: Vorzeichen (1 Bit), Exponent (8 Bit), Mantisse (23 Bit)

Exponent (e)	Zahlenwert	Bezeichnung
0	$(-1)^v \cdot 2^{-126} \cdot 0$, Mantisse	0, wenn Mantisse gleich 0, sonst denormalisierte Zahl
1	$(-1)^v \cdot 2^{-126} \cdot 1$, Mantisse	Normalisierte Zahl
…	$(-1)^v \cdot 2^{e-127} \cdot 1$, Mantisse	
254	$(-1)^v \cdot 2^{127} \cdot 1$, Mantisse	
255	Mantisse = 0: $(-1)^v \cdot \infty$, Überlauf	Unendlich
255	Mantisse ≠ 0: NaN (not a number)	Keine Zahl (NaN)

- Anmerkung:
  - Gleitkommazahlen nach IEEE 754 werden heute von vielen Prozessoren durch spezielle Gleitpunktrechenwerke unterstützt

Weitere Zahlen- und Zeichendarstellungen

HOCHSCHULE REGENSBURG UNIVERSITY OF APPLIED SCIENCES

- Varianten zur Zeichendarstellung
  - ASCII-Code (American Standard for Coded Information Interchange)
    - 7 Bit für 128 Zeichen, Erweiterung im 8. Bit möglich
  - Unicode
    - Internationaler Standard für Schriftzeichen und Textelemente aller Art
    - Vorgesehen sind max. 1.114.112 Zeichen (ca. 100.000 werden aktuell genutzt, das sind ca. 10 %)
- Varianten zur Zahlendarstellung
  - BCD-Code (Binary Coded Decimals)
    - Speicherung von Dezimalzahlen (ineffektiv bzgl. Speicher)
    - Für jede Dezimalziffer werden 4 (oder sogar 8) Bit verwendet und eine Ziffer wird nicht umgerechnet, sondern durch Ihren Dualwert angegeben
  - Graycode (z.B. A/D-Wandler)
    - Zwei aufeinanderfolgende Zahlen unterscheiden sich nur um ein Bit
  - Barcode
    - Strichcode auf vielen Artikeln (intern: ASCII- oder BCD-Code)

Übersicht ASCII-Code

HOCHSCHULE REGENSBURG UNIVERSITY OF APPLIED SCIENCES

000	NUL	033	!	066	B	099	c	132	ä	165	Ñ	198	ã	231	þ
001	Start Of Header	034	"	067	C	100	d	133	à	166	ª	199	Ã	232	Þ
002	Start Of Text	035	#	068	D	101	e	134	å	167	º	200	╚	233	Ú
003	End Of Text	036	$	069	E	102	f	135	ç	168	¿	201	╔	234	Û
004	End Of Transmission	037	%	070	F	103	g	136	ê	169	®	202	╩	235	Ù
005	Enquiry	038	&	071	G	104	h	137	ë	170	¬	203	╦	236	ý
006	Acknowledge	039	'	072	H	105	i	138	è	171	½	204	╠	237	Ý
007	Bell	040	(	073	I	106	j	139	ï	172	¼	205	═	238	¯
008	Backspace	041	)	074	J	107	k	140	î	173	¡	206	╬	239	´
009	Horizontal Tab	042	*	075	K	108	l	141	ì	174	«	207	¤	240	-
010	Line Feed	043	+	076	L	109	m	142	Ä	175	»	208	ð	241	±
011	Vertical Tab	044	,	077	M	110	n	143	Å	176	░	209	Ð	242	‗
012	Form Feed	045	-	078	N	111	o	144	É	177	▒	210	Ê	243	¾
013	Carriage Return	046	.	079	O	112	p	145	æ	178	▓	211	Ë	244	¶
014	Shift Out	047	/	080	P	113	q	146	Æ	179	│	212	È	245	§
015	Shift In	048	0	081	Q	114	r	147	ô	180	┤	213	ı	246	÷
016	Delete	049	1	082	R	115	s	148	ö	181	Á	214	Í	247	¸
017	-- frei --	050	2	083	S	116	t	149	ò	182	Â	215	Î	248	°
018	-- frei --	051	3	084	T	117	u	150	û	183	À	216	Ï	249	¨
019	-- frei --	052	4	085	U	118	v	151	ù	184	©	217	┘	250	·
020	-- frei --	053	5	086	V	119	w	152	ÿ	185	╣	218	┌	251	¹
021	Negative Acknowledge	054	6	087	W	120	x	153	Ö	186	║	219	█	252	³
022	Synchronous Idle	055	7	088	X	121	y	154	Ü	187	╗	220	▄	253	²
023	End Of Transmission Block	056	8	089	Y	122	z	155	ø	188	╝	221	¦	254	■
024	Cancel	057	9	090	Z	123	{	156	£	189	¢	222	Ì	255	
025	End Of Medium	058	:	091	[	124	|	157	Ø	190	¥	223	▀		
026	Substitude	059	;	092	\	125	}	158	×	191	┐	224	Ó		
027	Escape	060	<	093	]	126	~	159	ƒ	192	└	225	ß		
028	File Seperator	061	=	094	^	127	⌂	160	á	193	┴	226	Ô		
029	Group Seperator	062	>	095	_	128	Ç	161	í	194	┬	227	Ò		
030	Record Seperator	063	?	096	`	129	ü	162	ó	195	├	228	õ		
031	Unit Seperator	064	@	097	a	130	é	163	ú	196	─	229	Õ		
032		065	A	098	b	131	â	164	ñ	197	┼	230	µ		

Typische Wertebereiche (32-Bit-Architektur)

HOCHSCHULE REGENSBURG UNIVERSITY OF APPLIED SCIENCES

Datentyp	Bits	Wertebereich
`char, signed char`	8	$-128 \dots 127$
`unsigned char`	8	$0 \dots 255$
`short, signed short`	16	$-32768 \dots 32767$
`unsigned short`	16	$0 \dots 65535$
`int, signed int`	32	$-2.147.483.648 \dots 2.147.483.647$
`unsigned, unsigned int`	32	$0 \dots 4.294.967.295$
`long, signed long`	32	$-2.147.483.648 \dots 2.147.483.647$
`unsigned long`	32	$0 \dots 4.294.967.295$
`float`	32	$1{,}2 \cdot 10^{-38} \dots 3{,}4 \cdot 10^{38}$
`double`	64	$2{,}2 \cdot 10^{-308} \dots 1{,}8 \cdot 10^{308}$
`long double`	96	$3{,}4 \cdot 10^{-4932} \dots 1{,}1 \cdot 10^{4932}$