

Rechnerstrukturen, Teil 1

Vorlesung 4 SWS WS 14/15

Prof. Dr Jian-Jia Chen

Dr. Lars Hildebrand

Fakultät für Informatik – Technische Universität Dortmund

lars.hildebrand@tu-dortmund.de

<http://ls1-www.cs.tu-dortmund.de>

Übersicht

1. Organisatorisches ✓
2. Einleitung ✓
- 3. Repräsentation von Daten**
4. Boolesche Funktionen und Schaltnetze
5. Rechnerarithmetik
6. Optimierung von Schaltnetzen
7. Programmierbare Bausteine
8. Synchrone Schaltwerke

3. Repräsentation von Daten

3. Repräsentation von Daten

1. Repräsentation von natürlichen Zahlen ✓
2. **Repräsentation von Texten**
3. Repräsentation ganzer Zahlen
4. Repräsentation rationaler Zahlen
5. Repräsentation anderer Daten

3.2 Repräsentation von Texten

Beobachtung

Alles, was wir wissen, können wir aufschreiben.

Folgerung

Repräsentation von Texten ist zentral.

Naiver Ansatz

- Codierung $A \rightarrow 0, B \rightarrow 1, \dots$
- und dann Binärcodierung

Probleme

- Welche Zeichen sollen codiert werden?
- Wie kann man Daten mit anderen austauschen?

Lösung: Definition von Standards

3.2 Repräsentation von Texten

Texte: Zeichenfolgen aus Buchstaben und Satzzeichen

- Darstellung mittels Bitfolgen
- Codierung jedes Buchstabens / Zeichens durch Bitfolge

ASCII = American Standard Code for Information Interchange

- 7 Bit (= max. 128 Zeichen), Tabelle mit Nummerierung aller Zeichen
- z.B. „a“ hat Nummer 97, „A“ hat Nummer 65, „?“ hat Nummer 63
- Klein- und Großbuchstaben nach Alphabet durchnummeriert
- übliche Erweiterung auf PCs: 8bit, weitere Sonderzeichen, z.B. Umlaute
- Erweiterung in Europa: Latin-1, (nach Norm ISO 8859-1)

Unicode, z.B. von Java verwendet

- 16 Bit (= max. 65536 Zeichen)
- siehe <http://www.unicode.org>
- als Obermenge weltweit geläufiger Zeichensätze

3.2 Repräsentation von Texten

ASCII-Tabelle (7 Bit)

000001010011100101110111
0000...	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL
0001...	BS	HT	LF	VT	FF	CR	SO	SI
0010...	DLE	DC1	XON	DC3	XOF	NAK	SYN	ETB
0011...	CAN	EM	SUB	ESC	FS	GS	RS	US
0100...		!	"	#	\$	%	&	'
0101...	()	*	+	,	-	.	/
0110...	0	1	2	3	4	5	6	7
0111...	8	9	:	;	<	=	>	?
1000...	@	A	B	C	D	E	F	G
1001...	H	I	J	K	L	M	N	O
1010...	P	Q	R	S	T	U	V	W
1011...	X	Y	Z	[\]	^	_
1100...	'	a	b	c	d	e	f	g
1101...	h	i	j	k	l	m	n	o
1110...	p	q	r	s	t	u	v	w
1111...	x	y	z	{		}	~	DEL

3.2 Repräsentation von Texten

ISO 8859-1 (8 Bit)

0020	0	@	P	`	p		°	À	Ð	à	ð
0021	1	A	Q	a	q	i	±	Á	Ñ	á	ñ
0022	2	B	R	b	r	¢	²	Â	Ò	â	ò
0023	3	C	S	c	s	£	³	Ã	Ó	ã	ó
0024	4	D	T	d	t	¤	´	Ä	Ô	ä	ô
0025	5	E	U	e	u	¥	µ	Å	Õ	å	õ
0026	6	F	V	f	v		¶	Æ	Ö	æ	ö
0027	7	G	W	g	w	§	·	Ç	×	ç	÷
0028	8	H	X	h	x	¨	¸	È	Ø	è	ø
0029	9	I	Y	i	y	©	¹	É	Ù	é	ù
002A	:	J	Z	j	z	ª	º	Ê	Ú	ê	ú
002B	;	K	[k	{	«	»	Ë	Û	ë	û
002C	<	L	\	l		¬	¼	Ì	Ü	ì	ü
002D	=	M]	m	}	-	½	Í	Ý	í	ý
002E	>	N	^	n	~	®	¾	Î	Þ	î	þ
002F	?	O	_	o		¯	¿	Ï	ß	ï	ÿ

3.2 Repräsentation von Texten

ISO 8859-1 (8 Bit)

- International Organization for Standardization (gegründet 1947)
- ISO Latin 1
 - 8 Bit Code
 - enthält viele Sonderzeichen für westeuropäische Sprachen (z. B. Umlaute (¨ a, ¨ o, ¨ u, . . .))
 - enthält nicht alle gewünschten Zeichen (z. B. fehlen japanische und andere asiatische Schriftzeichen, . . .)

3.2 Repräsentation von Texten

Unicode

- aktueller Standard Unicode 7.0 (Stand: 5. Oktober 2014)
- verwaltet vom Unicode-Konsortium (<http://www.unicode.org>)
- unterstützt verschiedene Codierungsformate (Unicode Transformation Format):
 - UTF-8,
 - UTF-16,
 - UTF-32
- längere Formate erweitern kürzere Formate
- vereinbart auch weitere Informationen
 - Schreibrichtung,
 - Kombination von Zeichen (Codepoints))
 - ...

3.2 Repräsentation von Texten

Unicode

00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F
20	21	22	23	24	25	26	27	28	29	2A	2B	2C	2D	2E	2F
30	31	32	33	34	35	36	37	38	39	3A	3B	3C	3D	3E	3F
40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F
50	51	52	53	54	55	56	57	58	59	5A	5B	5C	5D	5E	5F
60	61	62	63	64	65	66	67	68	69	6A	6B	6C	6D	6E	6F
70	71	72	73	74	75	76	77	78	79	7A	7B	7C	7D	7E	7F
80	81	82	83	84	85	86	87	88	89	8A	8B	8C	8D	8E	8F
90	91	92	93	94	95	96	97	98	99	9A	9B	9C	9D	9E	9F
A0	A1	A2	A3	A4	A5	A6	A7	A8	A9	AA	AB	AC	AD	AE	AF
B0	B1	B2	B3	B4	B5	B6	B7	B8	B9	BA	BB	BC	BD	BE	BF
C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	CA	CB	CC	CD	CE	CF
D0	D1	D2	D3	D4	D5	D6	D7	D8	D9	DA	DB	DC	DD	DE	DF
E0	E1	E2	E3	E4	E5	E6	E7	E8	E9	EA	EB	EC	ED	EE	EF
F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	FA	FB	FC	FD	FE	FF

- Lateinische Schriften und Symbole
- Lautschriften
- Andere europäische Schriften
- Nahost- und Südwestasiatische Schriften
- Afrikanische Schriften
- Südasiatische Schriften
- Südostasiatische Schriften
- Ostasiatische Schriften
- CJK-Ideogramme
- Kanadische Silben
- Symbole
- Diakritika
- UTF-16-Surrogates und privater Nutzungsbereich
- Verschiedene Zeichen
- Nicht belegte Codebereiche

3.2 Repräsentation von Texten

Unicode – Beispiel: Kanadische Silben

$\Delta \sigma^b \cap \text{C}$
 Inuktitut Syllabarium
 $\text{f b o} \triangleright \text{L} \text{f b} < \Delta \text{C}$

Anlaute	Silben				Auslaute
	ai	i	u	a	
∅	▽	△	▷	◁	
p	∇	∧	>	<	◁◁
t	∪	∩	∩	∩	◁◁
k	∩	∩	∩	∩	◁◁
g	∩	∩	∩	∩	◁◁
m	∩	∩	∩	∩	◁◁
n	∩	∩	∩	∩	◁◁
s	∩	∩	∩	∩	◁◁
l	∩	∩	∩	∩	◁◁
j	∩	∩	∩	∩	◁◁
v	∩	∩	∩	∩	◁◁
r	∩	∩	∩	∩	◁◁
q	∩	∩	∩	∩	◁◁
ng	∩	∩	∩	∩	◁◁
nng	∩	∩	∩	∩	◁◁
∩		∩	∩	∩	◁◁

3.2 Repräsentation von Texten

Zur Bedeutung der Repräsentation von Texten

Programme

- Ein Programm wird zunächst meist als Text (**Quelltext**) erzeugt und wie normaler Text repräsentiert.
- Übersetzungsprogramme (**Compiler**) erzeugen daraus Programmcode in Maschinensprache.
- In jeder Form müssen alle Anteile eines Programms durch **Bitfolgen** codiert werden.

```
public class HelloWorld {  
    /**  
     * @param args  
     */  
    public static void main(String[] args) {  
        // TODO Auto-generated method stub  
        System.out.println("Hallo Welt!");  
    }  
}
```

```
// Bytecode stream: 03 3b 84 00 01 1a 05 68 3b a7 ff f9  
// Disassembly:  
iconst_0      // 03  
istore_0      // 3b  
iinc 0, 1     // 84 00 01  
iload_0       // 1a  
iconst_2      // 05  
imul         // 68  
istore_0      // 3b  
goto -7       // a7 ff f9
```

```
:1000000075812F12019912025212025890004D125E  
:10001000027B90005B120285750200744D12022D66  
:100020001200691203271200E304F9D8FED9FC7408  
:100030000E12022DE59012037A20B304050280020D  
:1000400015028502A030B2D37502FF80CE4449501C  
:100050003820503128686578293A00414455776541
```

3. Repräsentation von Daten

3. Repräsentation von Daten

1. Repräsentation von natürlichen Zahlen ✓
2. Repräsentation von Texten ✓
- 3. Repräsentation ganzer Zahlen**
4. Repräsentation rationaler Zahlen
5. Repräsentation anderer Daten

3.3 Repräsentation ganzer Zahlen

Wunsch

ganze Zahlen $z \in \mathbb{Z}$ repräsentieren zu können

Vereinbarung

feste Repräsentationslänge ℓ

Wie viele verschiedene Zahlen kann man so höchstens repräsentieren?

- ℓ Positionen, jeweils 2 Möglichkeiten pro Position
- $\rightarrow 2^\ell$ verschiedene Bitmuster der Länge ℓ

Fragen

- Wie könne wir Zahlenbereich verwenden?
- Wie können wir positive und negative Zahlen unterscheiden?

3.3 Repräsentation ganzer Zahlen

Vorzeichenbetragsmethode

Vereinbarungen

- erstes Bit Vorzeichen
- restliche Bits Betragszahl in Binärdarstellung
- Vorzeichenbit $s = 1 \Leftrightarrow$ Vorzeichen negativ, da $\text{Zahl} = (-1)^s \cdot \text{Betragszahl}$ gilt

Es gilt

- kleinste darstellbare Zahl $- (2^{\ell-1} - 1)$
- größte darstellbare Zahl $2^{\ell-1} - 1$

Eigenschaften

- + symmetrisch
- + Vorzeichenwechsel einfach
- – 0 nicht eindeutig
- – Vergleich von Zahlen schwierig

3.3 Repräsentation ganzer Zahlen

Darstellung mit Bias (Exzessdarstellung)

Vereinbarungen

- feste Verschiebung b (**Bias**).
- z wird als $z + b$ als Binärzahl dargestellt
- übliche Werte für Bias $b = 2^{\ell-1}$ oder $b = 2^{\ell-1}-1$
- Beispiel $\ell=5 \rightarrow b=16$ oder $b=15$

Es gilt

- kleinste darstellbare Zahl $-b$
- größte darstellbare Zahl $2^{\ell} - 1 - b$

Eigenschaften

- + 0 eindeutig & alle Codes ausgenutzt
- + Vergleich von Zahlen einfach
- + bei üblichem Bias erstes Bit vorzeichenbitanalog
- – nicht symmetrisch
- – Vorzeichenwechsel schwierig

3.3 Repräsentation ganzer Zahlen

Einerkomplementdarstellung

Vereinbarungen

- nicht-negative Zahlen: **Binärdarstellung**
- negative Zahlen **Komplement der Binärdarstellung**

Es gilt

- kleinste darstellbare Zahl $-(2^{\ell-1} - 1)$
- größte darstellbare Zahl $2^{\ell-1} - 1$

Eigenschaften

- + symmetrisch
- + erstes Bit wie Vorzeichenbit
- – 0 nicht eindeutig
- – Verschwendung eines Codes
- – Vergleich von Zahlen schwierig

3.3 Repräsentation ganzer Zahlen

Zweierkomplementdarstellung

Vereinbarungen

- nicht-negative Zahlen: **Binärdarstellung**
- negative Zahlen **Komplement der Binärdarstellung + 1**

Es gilt

- kleinste darstellbare Zahl $-(2^{\ell-1})$
- größte darstellbare Zahl $2^{\ell-1} - 1$

Eigenschaften

- 0 eindeutig repräsentiert & alle Codes ausgenutzt
- + erstes Bit wie Vorzeichenbit
- - Vergleich von Zahlen schwierig

Zweierkomplementdarstellung ist der Standard

3.3 Repräsentation ganzer Zahlen

Geschlossene Form zur Berechnung des Zweierkomplements

Theorem

Ein Bitvektor $a = (a_{n-1}, \dots, a_0)$ repräsentiert bei einer Kodierung im Zweierkomplement die Zahl

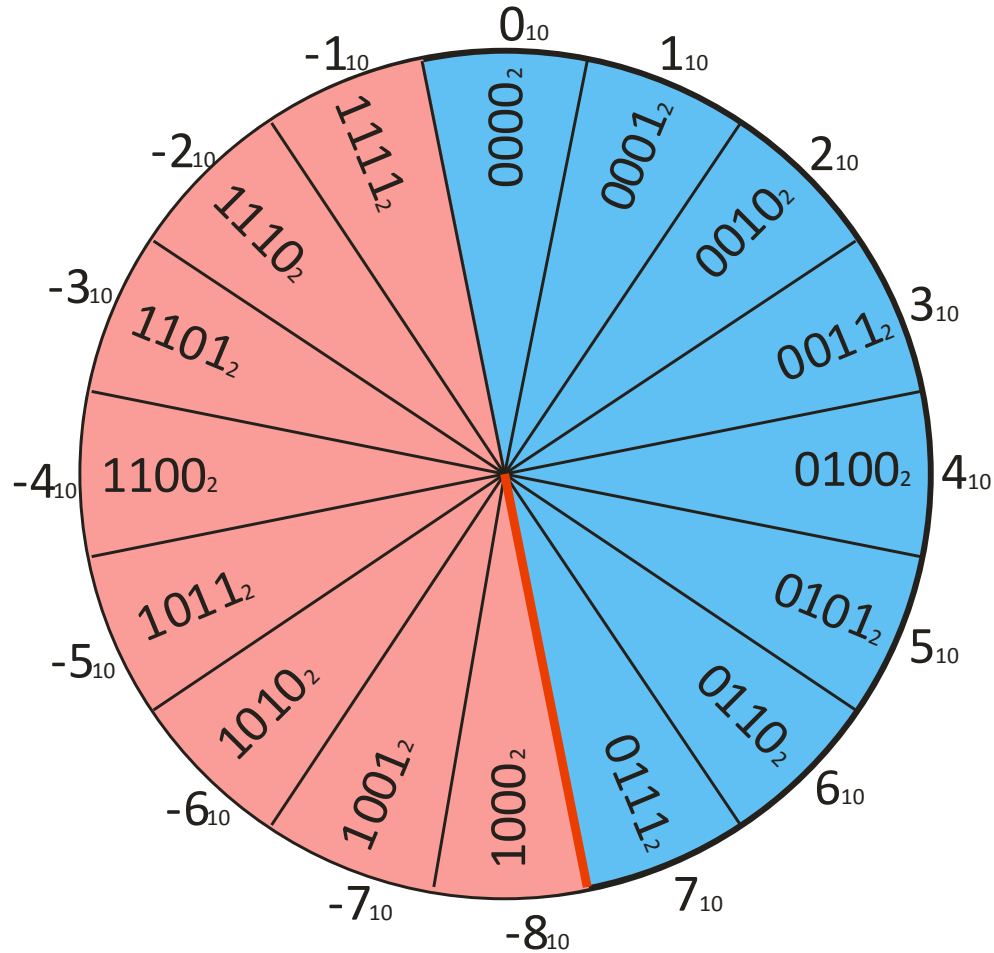
$$\text{int}(a) = -a_{n-1} \cdot 2^{n-1} + \sum_{i=0}^{n-2} a_i \cdot 2^i$$

Beispiel

$$\text{int}(1101_{2K}) = -1 \cdot 2^3 + (1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0) = -8_{10} + 5_{10} = -3_{10}$$

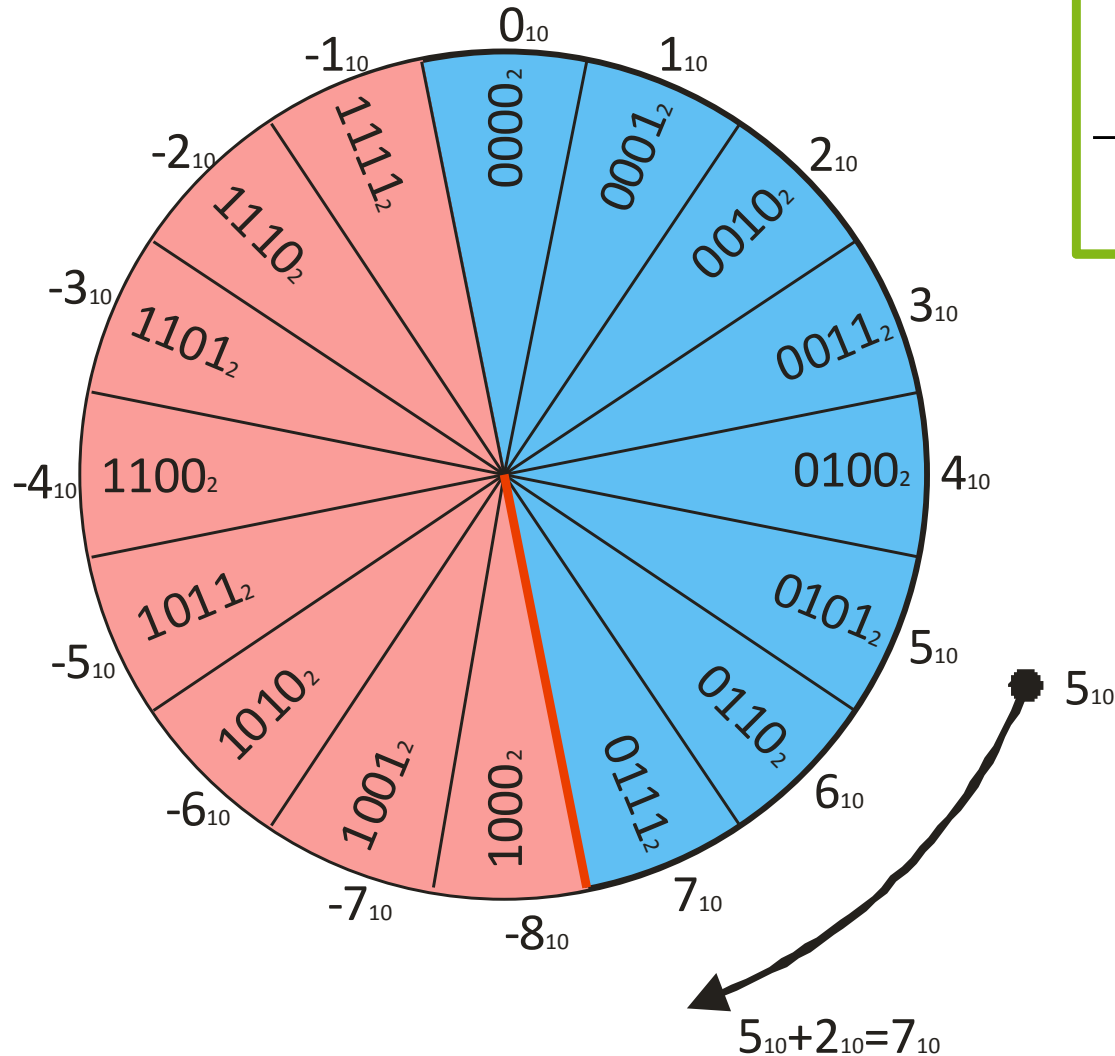
3.3 Repräsentation ganzer Zahlen

Für die Darstellung im Zweierkomplement ergibt sich folgendes **Zahlenrad**



3.3 Repräsentation ganzer Zahlen

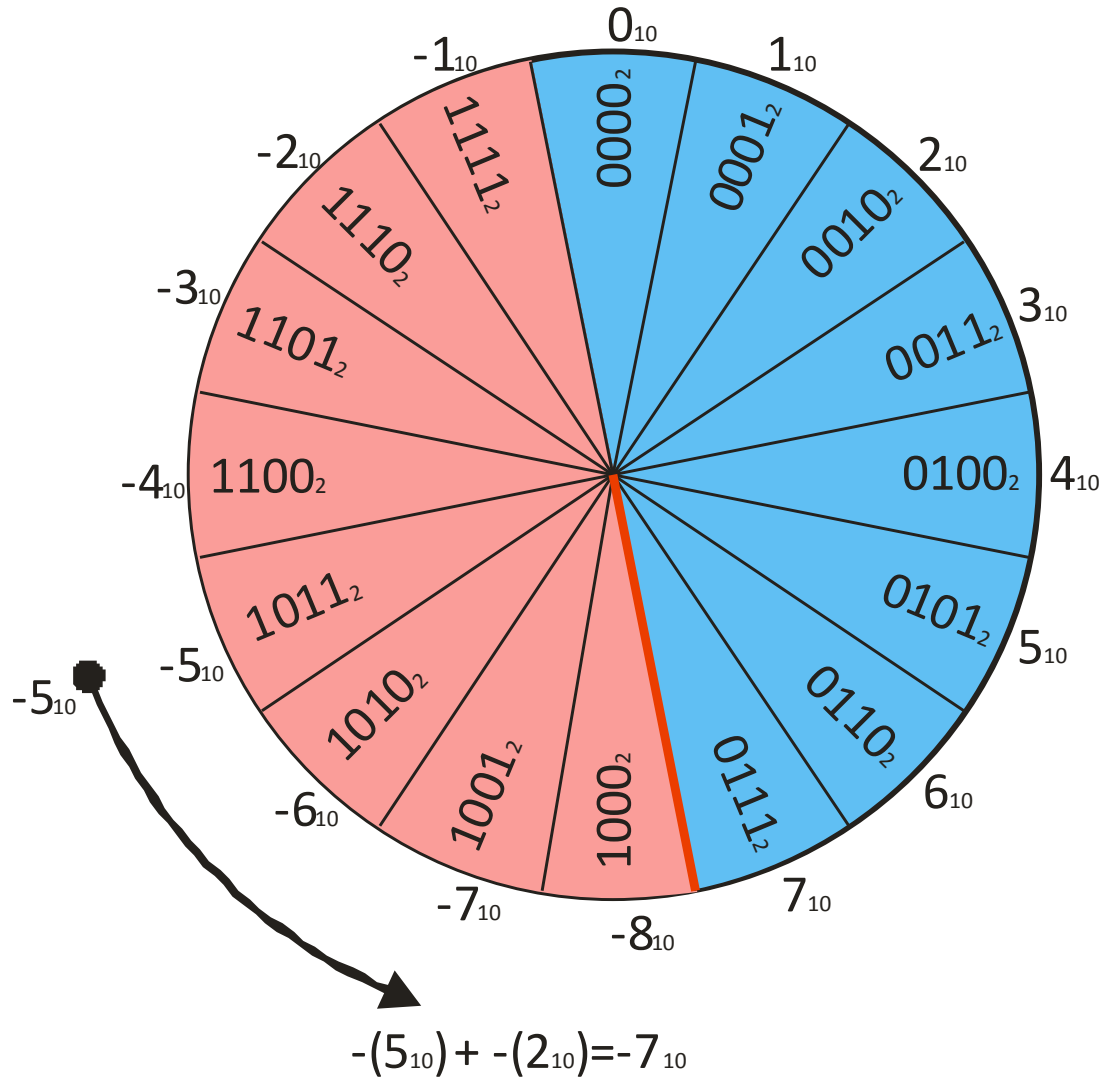
Keine Fallunterscheidung bei Addition und Subtraktion



5	0	1	0	1
+ 2	0	0	1	0
7	0	1	1	1

3.3 Repräsentation ganzer Zahlen

Keine Fallunterscheidung bei Addition und Subtraktion

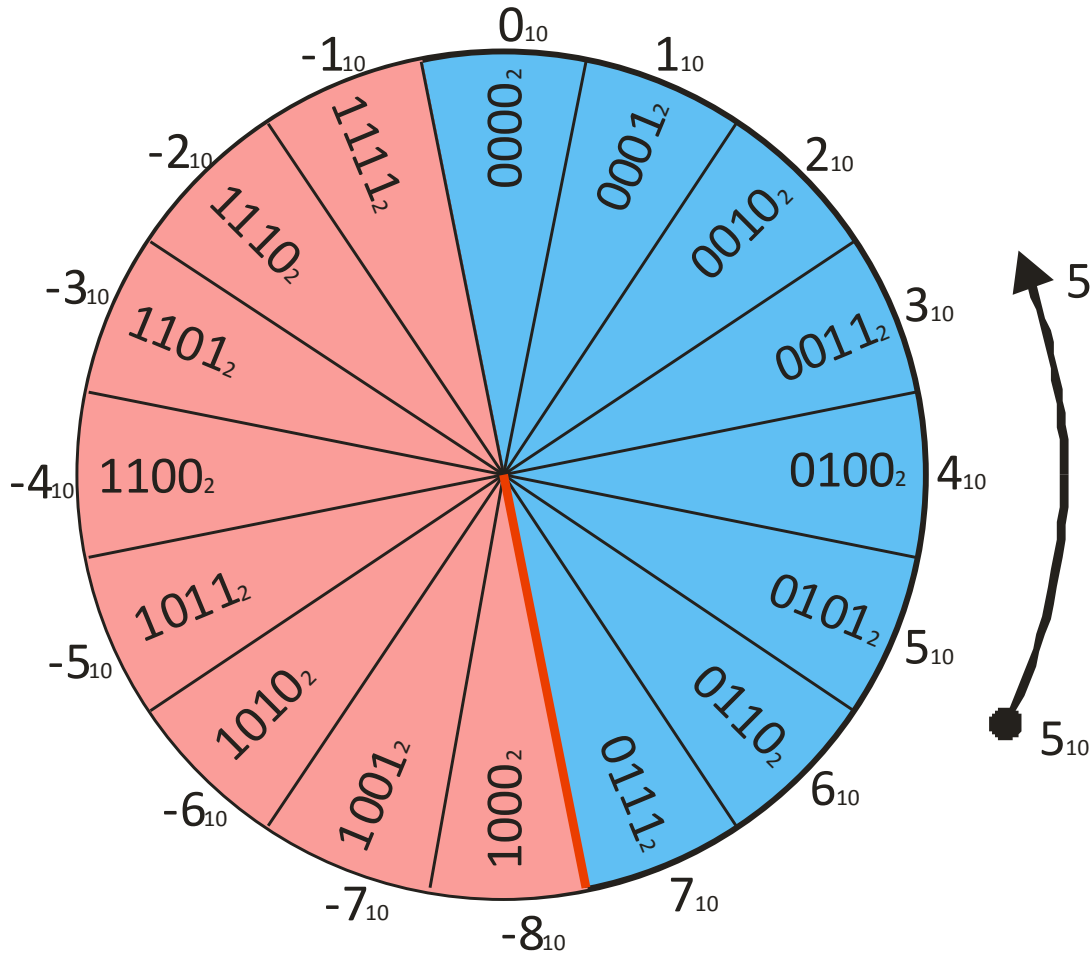


-5		1	0	1	1
+ -2		1	1	1	0
-7	1	1	0	0	1

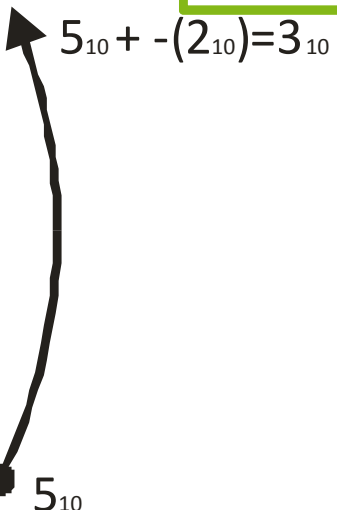
Übertragbit ignorieren

3.3 Repräsentation ganzer Zahlen

Keine Fallunterscheidung bei Addition und Subtraktion



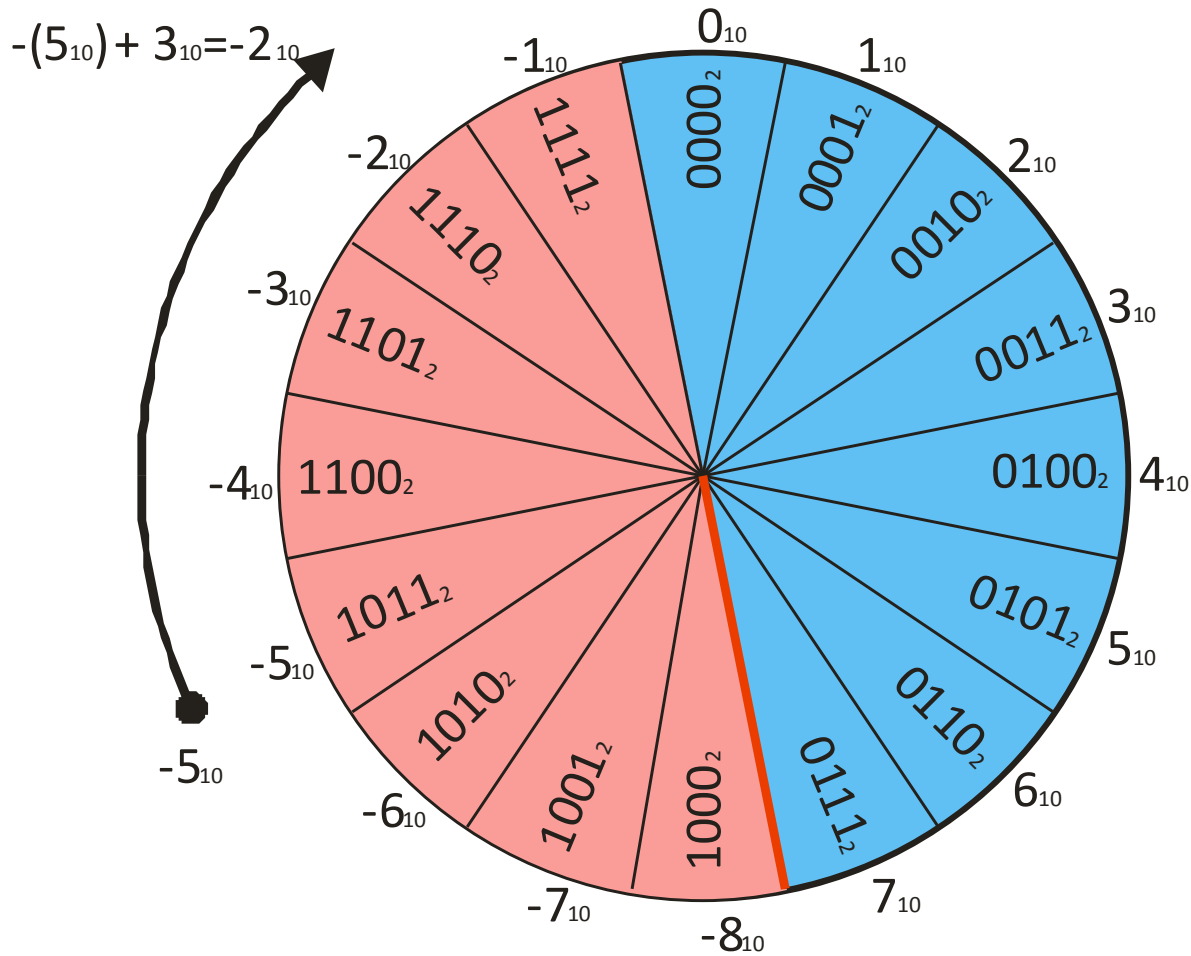
5		0	1	0	1
+ -2		1	1	1	0
3	1	0	0	1	1



Übertragbit ignorieren

3.3 Repräsentation ganzer Zahlen

Keine Fallunterscheidung bei Addition und Subtraktion

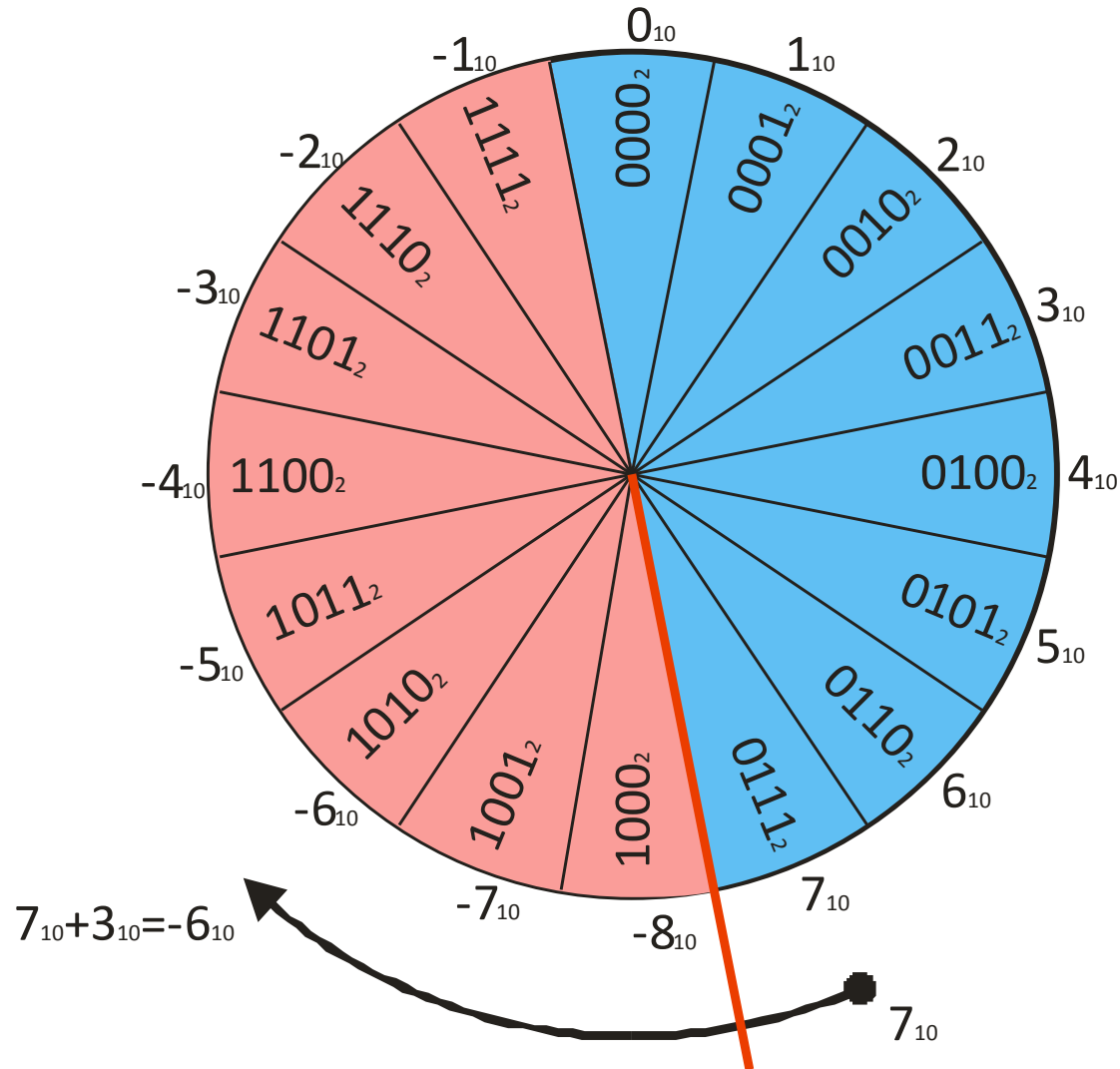


-5		1	0	1	1
+ 3		0	0	1	1
-2	0	1	1	1	0

Übertragbit ignorieren

3.3 Repräsentation ganzer Zahlen

Aber: **Vorsicht bei Bereichsüberschreitung**



7		0	1	1	1
+ 3		0	0	1	1
<hr/>					
-6	0	1	0	1	0

Übertragbit ignorieren

3.3 Repräsentation ganzer Zahlen

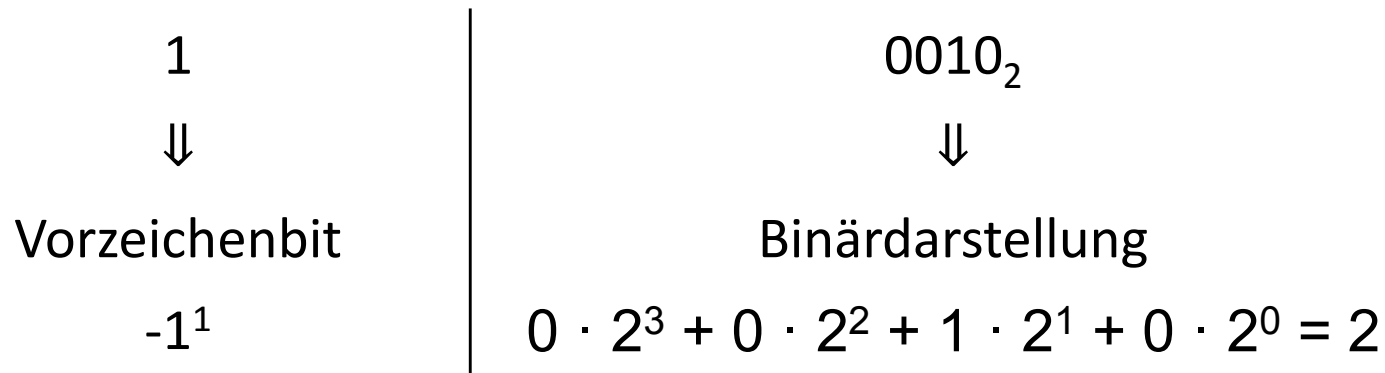
Alles verstanden?

Wir haben

- $\ell = 5$
- Bias $b = 2^{\ell-1} = 2^4 = 16$
- $z = 10010$

Welche Zahl wird denn nun repräsentiert?

Vorzeichenbetragsdarstellung



→ $10010_2 = -1 \cdot 2 = -2$

3.3 Repräsentation ganzer Zahlen

Alles verstanden?

Wir haben

- $\ell = 5$
- Bias $b = 2^{\ell-1} = 2^4 = 16$
- $z = 10010$

Welche Zahl wird denn nun repräsentiert?

Darstellung mit Bias $b = 16$ (Exzessdarstellung)

$$\begin{aligned}10010_2 &= 1 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 - b \\ &= 16 + 2 - b \\ &= 18 - 16 \\ &= 2\end{aligned}$$

$$\rightarrow 10010_2 = 2$$

3.3 Repräsentation ganzer Zahlen

Alles verstanden?

Wir haben

- $\ell = 5$
- Bias $b = 2^{\ell-1} = 2^4 = 16$
- $z = 10010$

Welche Zahl wird denn nun repräsentiert?

Einerkomplementdarstellung

- Führendes Bit 1 \rightarrow negative Zahl
- Komplement von $10010_2 = 01101_2$
- $01101_2 = 0 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0$
 $= 8 + 4 + 1 = 13$

$\rightarrow 10010_2 = -13$

3.3 Repräsentation ganzer Zahlen

Alles verstanden?

Wir haben

- $\ell = 5$
- Bias $b = 2^{\ell-1} = 2^4 = 16$
- $z = 10010$

Welche Zahl wird denn nun repräsentiert?

Zweierkomplementdarstellung

- Führendes Bit 1 \rightarrow negative Zahl
- Zweierkomplement: $10010_2 - 1_2 = 10001_2$
- Komplement von $10001_2 = 01110_2$
- $01110_2 = 0 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 = 14$

$\rightarrow 10010_2 = -14$

3.3 Repräsentation ganzer Zahlen

Alles verstanden?

Wir haben

- $\ell = 5$
- Bias $b = 2^{\ell-1} = 2^4 = 16$
- $z = 10010$

Welche Zahl wird denn nun repräsentiert?

- -2 Vorzeichenbetragsdarstellung
- 2 Darstellung mit Bias $b = 16$ (Exzessdarstellung)
- -13 Einerkomplementdarstellung
- -14 Zweierkomplementdarstellung

Und richtig sind ohne weitere Informationen über den Datentyp alle 4!

3.3 Repräsentation ganzer Zahlen

Beispielsammlung Darstellungen

Wir haben feste Länge $\ell = 5$

z	VZ-Betrag	Bias b=16	Bias b=15	1er-K.	2er-K.
-16	-	00000	-	-	10000
-15	11111	00001	00000	10000	10001
...
-1	10001	01111	01110	11110	11111
0	00000 10000	10000	01111	00000 11111	0000
1	00001	10001	10000	00001	00001
...
15	01111	11111	11110	01111	01111
16	-	-	11111	-	-

3. Repräsentation von Daten

3. Repräsentation von Daten

1. Repräsentation von natürlichen Zahlen ✓
2. Repräsentation von Texten ✓
3. Repräsentation ganzer Zahlen ✓
4. **Repräsentation rationaler Zahlen**
5. Repräsentation anderer Daten

3.4 Repräsentation rationaler Zahlen

Wunsch

rationale Zahlen $q \in \mathbb{Q}$ repräsentieren können

Vereinbarung

- feste Repräsentationslänge ℓ Bits
- feste Position des Kommas
- **Festkommadarstellung**

Beispiele für $\ell = 8$ bei 3 Nachkommastellen

- 10,34 \rightarrow 00 010,340
- 93 847,123 \rightarrow 93 847,123
- 0 \rightarrow 00 000,000
- 123 456,78 nicht darstellbar
(zu viele Stellen vor dem Komma)
- 12,345678 nicht darstellbar
(aber runden denkbar)

3.4 Repräsentation rationaler Zahlen

Beobachtung

- Basis 10 im Bereich der digitalen Rechner nur bedingt brauchbar
- Idee ist aber übertragbar auf Basis 2

Beispiel: 93 847,123

10^4	10^3	10^2	10^1	10^0	10^{-1}	10^{-2}	10^{-3}
10 000	1 000	100	10	1	0,1	0,01	0,001
9	3	8	4	7	1	2	3

Übertragung auf Basis 2

2^4	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}
16	8	4	2	1	$1/2 = 0,5$	$1/4 = 0,25$	$1/8 = 0,125$
1	0	1	1	0	1	0	1

$$= 16 + 4 + 2 + 0,5 + 0,125 = 22,625$$

3.4 Repräsentation rationaler Zahlen

Allgemein gilt

- bei v **Vorkommastellen**
- und m **Nachkommastellen**

$$a = \sum_{i=-m}^{v-1} z_i \cdot 2^i$$

3.4 Repräsentation rationaler Zahlen

Beobachtung

- nicht alle Zahlen sind darstellbar
 - z. B.: 0,2
 - z. B.: π
- ist das schlimm?
 - im Dezimalsystem ist $1/3$ nicht darstellbar
 - π hat unendlich viele Stellen, ist generell nicht darstellbar
- Festkommadarstellung hat **Vorteile**
 - bildet Rechenoperationen auf Ganzzahloperationen ab
 - keine Fließkommahardware nötig
 - schnelle Ausführung
 - Einsatz im Bereich der digitalen Signalverarbeitung
- **Nachteile**
 - feste Position des Kommas
 - unflexibel

3.4 Repräsentation rationaler Zahlen

Andere Idee

2014

3.4 Repräsentation rationaler Zahlen

Gleitkommazahlen

- Vorzeichenbit $s \in \{0,1\}$
- Mantisse $m \in \mathbb{Q}$
- Exponent $e \in \mathbb{Z}$ (bestimmt die Position des Kommas)

$$z = (-1)^s \cdot m \cdot 10^e$$

Normalisierte Gleitkommazahl

- $1 \leq m < 10$
- Beispiel: $2,014 \cdot 10^3$

3.4 Repräsentation rationaler Zahlen

Übertragung auf Basis $b=2$

- Vorzeichenbit $s \in \{0,1\}$
- Mantisse $m \in \mathbb{Q}$
- Exponent $e \in \mathbb{Z}$ (bestimmt die Position des Kommas)

$$z = (-1)^s \cdot m \cdot 2^e$$

Normalisierte Gleitkommazahl

- $1 \leq m < 2$

Beobachtung

- erste Ziffer der Mantisse ist **immer 1**
- 0 ist so nicht darstellbar

3.4 Repräsentation rationaler Zahlen

IEEE-754 1985

- Vorzeichenbit $s \in \{0,1\}$
- Mantisse $m \in \mathbb{Q}$
- Exponent $e \in \mathbb{Z}$
- $1 \leq m < 2$

$$z = (-1)^s \cdot m \cdot 2^e$$

Festlegung

- führende 1 der Mantisse wird nicht mit abgespeichert: **Implizite Eins**
- Mantisse in Binärcodierung als
Festkommazahlen mit Ziffern **ausschließlich hinter dem Komma**
- Exponent in Exzessdarstellung mit Bias $b = 2^{\ell_e - 1} - 1$

3.4 Repräsentation rationaler Zahlen

IEEE-754 1985

Definierte Varianten

Typ	Größe (1+r+p)	Exponent (r)	Mantisse (p)	Werte des Exponenten (e)	Biaswert (B)
single	32 bit	8 bit	23 bit	$-126 \leq e \leq 127$	127
double	64 bit	11 bit	52 bit	$-1022 \leq e \leq 1023$	1023

Besonderheiten beim Exponenten

Verkleinerung des zulässigen Bereichs um 1 auf beiden Seiten auf

- $e_{\min} = -b + 1 = -(2^{\ell e - 1} - 1) + 1,$
- $e_{\max} = 2^{\ell e} - 1 - b - 1 = 2^{\ell e - 1} - 1$

dient der Codierung **besonderer Zahlen**

3.4 Repräsentation rationaler Zahlen

IEEE-754 1985

Besondere Zahlen

Vorzeichen	Exponent	Mantisse	Zahl
0	$e_{\max} + 1$	0	$+\infty$
1	$e_{\max} + 1$	0	$-\infty$
s	$e_{\max} + 1$	$\neq 0$	NaN
0	$e_{\min} - 1$	0	0
1	$e_{\min} - 1$	0	-0
s	$e_{\min} - 1$	$\neq 0$	$(-1)^s \cdot \left(\sum_{i=1}^{l_m} m_i \cdot 2^{-i} \right) \cdot 2^{e_{\min}}$

NaN: Not a Number

letzte Zeile: denormalisierte Darstellung

3.4 Repräsentation rationaler Zahlen

IEEE-754 1985

Beispiel 1 (Typ: single)

- $\ell = 32, \ell_s = 1, \ell_e = 8, \ell_m = 23$
- $b = 2^7 - 1 = 127 \rightarrow e_{\min} = -b + 1 = -126, e_{\max} = 2^8 - b - 2 = 127$

Wir wollen -3 darstellen:

- **negativ**, also **Vorzeichenbit** $s=1$
- **Darstellung als Summe von Zweierpotenzen**
 $3 = 2 + 1 = 2^1 + 2^0 = (2^0 + 2^{-1}) \cdot 2^1$
- **Mantisse** $(11000\dots)_2$, **implizite Eins** entfällt, also $100\ 0000\ 0000\ \dots$
- **Exponent** **Exzessdarstellung** $1 + b = 128$ darstellen
- $128 = (1000\ 0000)_2$

1	1000 000	100 0000 0000 0000 0000 0000
⇓	⇓	⇓
Vorzeichenbit	Exponent	Mantisse

3.4 Repräsentation rationaler Zahlen

IEEE-754 1985

Beispiel 2 (Typ: single)

- $\ell = 32, \ell_s = 1, \ell_e = 8, \ell_m = 23$
- $b = 2^7 - 1 = 127 \rightarrow e_{\min} = -b + 1 = -126, e_{\max} = 2^8 - b - 2 = 127$

Wir wollen 0,0546875 darstellen:

- **positiv**, also **Vorzeichenbit** $s=0$
- **Darstellung als Summe von Zweierpotenzen**
 $0,0546875 = 1/32 + 1/64 + 1/128 = 2^{-5} + 2^{-6} + 2^{-7} = (2^0 + 2^{-1} + 2^{-2}) \cdot 2^{-5}$
- **Mantisse** $(11100\dots)_2$, **implizite Eins** entfällt, also 110 0000 0000 \dots
- **Exponent -5 Exzessdarstellung** $-5 + b = 122$ darstellen
- $128 = (0111\ 1010)_2$

0	0111 1010	110 0000 0000 0000 0000 0000
⇓	⇓	⇓
Vorzeichenbit	Exponent	Mantisse

3.4 Repräsentation rationaler Zahlen

IEEE-754 1985

Beispiel 3 (Typ: single)

- $\ell = 32, \ell_s = 1, \ell_e = 8, \ell_m = 23$
- $b = 2^7 - 1 = 127 \rightarrow e_{\min} = -b + 1 = -126, e_{\max} = 2^8 - b - 2 = 127$

Gegeben sei 0 1011 0001 010 1001 0000 0000 0000 0000:

0	1011 0001	010 1001 0000 0000 0000 0000
↓	↓	↓
Vorzeichenbit	Exponent	Mantisse

- **Vorzeichenbit** $s=0$, also **positiv**
- **Exponent** $(1011\ 0001)_2 = 177$
- **Exzessdarstellung** $177 - b = 177 - 127 = 50$
- **Mantisse** $(010\ 1001\ \dots)_2$, zuzüglich **impliziter Eins** $(1,010\ 1001\ \dots)_2$
- $(2^0 + 2^{-2} + 2^{-4} + 2^{-7}) \cdot 2^{50} = 2^{50} + 2^{48} + 2^{46} + 2^{43} = 1\ 486\ 539\ 720\ 753\ 152$

3.4 Repräsentation rationaler Zahlen

IEEE-754 1985

Denormalisierte Darstellung

- gilt, wenn Exponent = $e_{\min} - 1$
- und Mantisse $\neq 0$
- erlaub es, noch kleinere Zahlen darzustellen

$$(-1)^s \cdot \left(\sum_{i=1}^{l_m} m_i \cdot 2^{-i} \right) \cdot 2^{e_{\min}}$$

Beispiel

- `If x \neq y Then z := 1 / (x - y)`
- `x = 0 0000 0001 (1) 000 0000 0000 0000 0000 0001` $2^{-126} + 2^{-149}$
- `y = 0 0000 0001 (1) 000 0000 0000 0000 0000 0000` 2^{-126}

klar $x \neq y$, aber ohne denormalisierte Darstellung ist $x - y = 0$ gerundet

- denormalisiert $x - y = 2^{-149}$
- $x - y = 2^{-149} = 0 0000 0000 000 0000 0000 0000 0000 0001$

- auf jeden Fall besserer Programmierstil

`If x - y \neq 0 Then z := 1 / (x - y)`

3. Repräsentation von Daten

3. Repräsentation von Daten

1. Repräsentation von natürlichen Zahlen ✓
2. Repräsentation von Texten ✓
3. Repräsentation ganzer Zahlen ✓
4. Repräsentation rationaler Zahlen ✓
5. **Repräsentation anderer Daten**

3.5 Repräsentation anderer Daten

Was sind andere Daten?

- Bilder, Audiodaten, Videodaten
- Objekte, Felder
- für uns interessant: primitive Daten
Datentypen, die direkt von der Hardware unterstützt werden

Programme

- Bitmuster fester Länge (z. B.: 32 Bit, 64 Bit)
 - Befehl
 - Operanden

Problem: Was repräsentiert ein Byte im Speicher?

- Typbits höchstens für große Bereiche (Programm, Daten)
- Bei Daten sorgt der Compiler für richtige Interpretation
- Typisierung wichtig!

3.5 Repräsentation anderer Daten

Repräsentation von Datenfolgen

Speicher oft in **Worten** organisiert

- Wort ja nach Rechner 2 Bytes, 4 Bytes, . . .

Heterogene Daten

- werden hintereinander in den Speicher geschrieben
- dabei manchmal Wortgrenzen beachten
- dann leere Zellen (Bytes) möglich

Homogene Daten

- Arrays
- Problem Wie erkennt man das Ende der Folge?
 - feste Anzahl vereinbaren
 - Länge am Anfang speichern
 - spezielles Endezeichen verwenden

3. Repräsentation von Daten

3. Repräsentation von Daten

1. Repräsentation von natürlichen Zahlen ✓
2. Repräsentation von Texten ✓
3. Repräsentation ganzer Zahlen ✓
4. Repräsentation rationaler Zahlen ✓
5. Repräsentation anderer Daten ✓