Exkurs:

Fixkomma- und Gleitkommazahlen

Zahlen – mit und ohne Komma

- einfachste Variante: Ganze Zahlen
- Fixkommazahlen
- Gleitkommazahlen
- IEEE 754

Ganze Zahlen (1/5)

- z. B. 8 Bit
- ohne Vorzeichen
  - 00000000 (0) bis 11111111 (255)
  - klassische Dualzahlen

Ganze Zahlen (2/5)

- mit Vorzeichen
  - erster Ansatz: höchstes Bit ist Vorzeichen (0=positiv, 1=negativ)
  - 00000000 (0) bis 01111111 (127)
    10000000 (-0) bis 11111111 (-127)
  - diverse Probleme
    - Null doppelt (0; -0)
    - Addition von zwei Zahlen:

```
00000100 (4)
10001001 (-9)
--------
10001101 (-13) ???
```

Ganze Zahlen (3/5)

- mit Vorzeichen
  - zweiter Ansatz: **Einerkomplement**
  - negative Zahl -x entsteht aus x durch Kippen aller Bits
  - 00000000 (0) bis 01111111 (127)
    11111111 (-0) bis 10000000 (-127)
  - weiter Probleme: 0 immer noch doppelt, Addition:

```
00000100 ( 4)                      10110000 (-79)
11110110 (-9)        aber:         01100000 ( 96)
--------                        -----------
11111010 (-5)   ok              (1)00010000 ( 16)
```

Ergebnis sollte 17 sein!

  - Bei Addition Übertrag zum Ergebnis addieren!

Ganze Zahlen (4/5)

- mit Vorzeichen
  - dritter Ansatz: **Zweierkomplement**
  - ähnlich Einerkomplement; -x entsteht aus x durch Kippen aller Bits und anschließende **Addition von 1**
  - 00000000 (0) bis 01111111 (127)
    11111111 (-1) bis 10000000 (-128)

    (repräsentierte Zahlen sind 1 kleiner als im 1-Kompl.)
  - Vorteil: keine Doppelte 0; 256 verschiedene Zahlen

Ganze Zahlen (5/5)

Addition im Zweierkomplement:

–4 + 3 = –1:	–4 + 4 = 0:
11111100	11111100
00000011	00000100
--------	---------
11111111	100000000
	Übertrag ignorieren

Fixkommazahlen (1/5)

- Auch hier können wir das Zweierkomplement nutzen:
  - z. B. 8 Bit, 4 Bit vor dem Komma, 4 Bit dahinter: xxxx,xxxx
  - alle Werte mit Faktor 1/16 „skaliert“
  - 00000001 → 1/16 (=1/16 * 1)
    00010000 → 1 (=1/16 * 16)
    10000000 → –8 (=1/16 * (–128))

Fixkommazahlen (2/5)

- Addition mit Zweierkomplement

1 + (–8):

```
00010000
10000000
--------
10010000 → -7 (=1/16 * (-112))
```


Fixkommazahlen (3/5)

- „Abgeschlossenheit" unter Operationen
  - Addition
  - Subtraktion
  - Multiplikation
  - Division
- in der Mathematik: M abgeschlossen unter $\oplus$, falls $m_1 \oplus m_2$ in M für alle m_1, m_2 in M
- hier: Ergebnisse von $m_1 \oplus m_2$ sollten (in akzeptabler Näherung) darstellbar sein

Fixkommazahlen (4/5)

- Addition: relativ unproblematisch

```
010,111       110,111
001,101       001,101
-------       -------
100,100      1000,100
```

- Multiplikation: passt nur in wenigen Fällen

```
010,111 * 010,111 = 1000,010001
```

(vorne und hinten zu viele Stellen)

Fixkommazahlen (5/5)

- Fixkommazahlen meist nur dann nützlich, wenn klar ist, dass alle bearbeiteten Werte aus dem erfassten Bereich stammen
- Bei größerer Bandbreite an Zahlen: Gleitkommazahlen verwenden

Motivation für Gleitkommazahlen

mathematische / naturwiss. Notation

- Gravitationskonstante: $G = 6{,}6742 \cdot 10^{-11}$
- Lichtgeschw. $c = 2{,}99792458 \cdot 10^{8}$ m/s

Idee: „signifikante Ziffern“ und Größenordnung speichern

Gleitkommazahlen (1/7)

- Wdh.: Gleitkommazahlen (64 Bit, IEEE 754)
  - Ein Bit *v* für das Vorzeichen (gesetztes Bit bedeutet negatives Vorzeichen)
  - Elf Bit für den Exponenten. Zum Exponent *E* addiert man den **Exzess** $q = 2^{11-1} - 1$:
    $e = E + q = E + 1023$ (*e* ist vorzeichenlos, *E* nicht)
  - Die restlichen 52 Bit für die Nachkommastellen des Betrags des gebrochenen Anteils *f*

v	Exponent *e*	Mantisse *f*
1	11	52

Gleitkommazahlen (2/7)

- $a = m \cdot 2^e$, $b = n \cdot 2^f$; sei $e > f$, $f = e - g$
- Addition:
  $a+b = m \cdot 2^e + n \cdot 2^f = m \cdot 2^e + n \cdot 2^{e-g}$
  $= m \cdot 2^e + n \cdot 2^e / 2^g = (m + n/2^g) \cdot 2^e = \ldots$
  (Division durch 2^g: Rechts-Shift um g Stellen)
  - Beispiel: $1{,}0101 \cdot 2^4 + 1{,}111 \cdot 2^2 =$
    $1{,}0101 \cdot 2^4 + 0{,}01111 \cdot 2^4 =$
    $1{,}100011 \cdot 2^4$
- Subtraktion: wie Addition

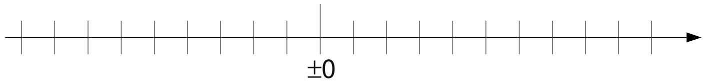
Gleitkommazahlen (3/7)

- $a = m \cdot 2^e$, $b = n \cdot 2^f$; sei $e > f$, $f = e - g$
- Multiplikation:
  $a \cdot b = m \cdot 2^e \cdot n \cdot 2^f = m{\cdot}n \cdot 2^{e+f} = \ldots$

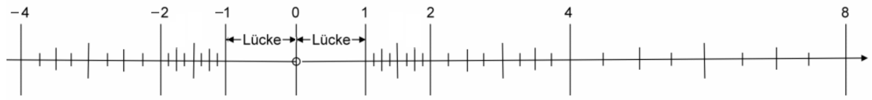
  Falls Darstellung von m·n nicht 1,... ist:
  $\rightarrow$ normalisieren
- Division: wie Multiplikation; $2^e / 2^f = 2^{e-f}$
- Immer beachten: Darstellung speichert nicht den Exponenten, sondern Exp. + Exzess

Gleitkommazahlen (4/7)

- Werteverteilung
  - Fixkommazahlen: gleichmäßige Abstände



  - Gleitkommazahlen: sehr ungleichmäßig:



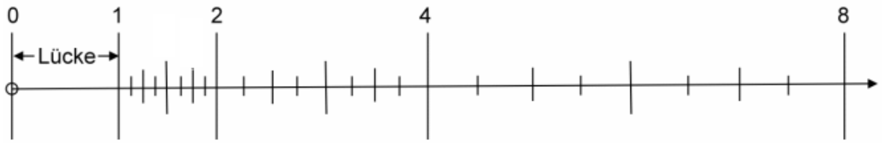
3 Bit für die Mantisse; 2 Bit für Exponent
Bild: http://www.cirsovius.de/CPM/Projekte/Artikel/Mathe/IEEE_Reals/IEEE1.html, bearbeitet

Gleitkommazahlen (5/7)

Beispiel:

- 3 Bit Mantisse (1,000...1,111)
- 2 Bit Exponent (0..3)

Exp.:	2^0	2^1	2^2	2^3
1,000	1,0000	2,0000	4,0000	8,0000
1,001	1,1250	2,2500	4,5000	9,0000
1,010	1,2500	2,5000	5,0000	10,000
1,011	1,3750	2,7500	5,5000	11,000
1,100	1,5000	3,0000	6,0000	12,000
1,101	1,6250	3,2500	6,5000	13,000
1,110	1,7500	3,5000	7,0000	14,000
1,111	1,8750	3,7500	7,5000	15,000



Gleitkommazahlen (6/7)

- „Lücke“ bei 0 beheben: **subnormale** (denormalisierte) Zahlen einfügen – erlaube führende 0

v	Exponent *e*	Mantisse *f*
0	00000000000	000010111100....

- interpretiert als: **0,**000010111100... · 2^{MIN}
- Abweichung von sonstiger Norm; erlaubt sehr kleine Zahlen um 0 herum

Gleitkommazahlen (7/7)

weitere Ausnahmen

- ±∞: maximaler Exponent, Mantisse 0
- NaN: maximaler Exponent, Mantisse > 0 (not a number: z. B. Ergebnis von 0/0)
  - versch. Mantissen → versch. „Nicht-Zahlen“
- ±0: minimaler Exponent, Mantisse 0 (wird wirklich zwischen +0 und -0 unterschieden)
- „Rechenregeln“: 1 / (+0) = +∞, 1 / (–0) = –∞