02 - Numerik

Technische Grundlagen der Informatik

Automation Systems Group E183-1

Institute of Computer Aided Automation
Vienna University of Technology

email: tgi@auto.tuwien.ac.at

Numerik

- Methoden zur Lösung mathematischer Problemstellungen auf Computern

- Hauptfelder
  - effektive und effiziente Berechnung
  - Fehlerabschätzung
  - Aufwandsabschätzung
  - Stabilitätsanalyse

- Anwendungsgebiete in
  - Ingenieur-, Natur-, Wirtschafts- und Sozialwissenschaften
  - Vor allem Simulation komplexer Vorgänge
  - Z.B. Wettervorhersage, Windkanalversuche, Finanzmathematik

Zahlendarstellung im Computer

- In Mathematik unendliche Zahlenmengen
  - $\mathbb{N}$, $\mathbb{Z}$, $\mathbb{Q}$, $\mathbb{R}$, $\mathbb{C}$

- Am Computer endliche Zahlenmengen
  - Stellenwertsystem zur Basis 2 mit n Stellen
  - 2^n unterschiedliche „Zahlen“

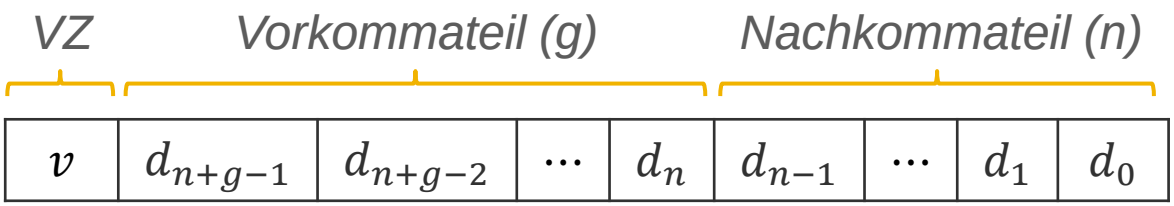
  - $\mathbb{N}$: 0 ... $2^n - 1$

  - $\mathbb{Z}$: negative Zahlen erfordern Kodierung
    - VZ + Betrag, Einer- / Zweierkomplement, Exzessdarstellung

  - $\mathbb{Q}$, $\mathbb{R}$: Nachkommastellen!

Festpunkt-Darstellung

- Gesamtlänge $N = 1 + g + n$ Bit
  - g Vorkommastellen
  - n Nachkommastellen
  - Vorzeichen v ($0 \Rightarrow$positiv, $1 \Rightarrow$negativ)



- entspricht Skalierung der ganzen Zahl Z um Faktor 2^{-n}

- Bitfolge $vd_{N-2}d_{N-3}\cdots d_1d_0$ interpretiert als
  - vorzeichenbehaftete Binärzahl mit n Nachkommastellen

$$vd_{N-2}d_{N-3}\ldots d_1d_0 \doteq (-1)^v \cdot 2^{-n} \sum_{j=0}^{N-2} d_j \cdot 2^j \doteq$$

Kodierung

$$\doteq (-1)^v \cdot d_{N-2}\ldots d_n . d_{n-1}\ldots d_1d_0$$ ← Festpunkt-Zahl

Festpunkt-Darstellung

- Für das Festpunkt-Zahlensystem mit $N = 16$ Bit Breite und $n = 3$ Nachkommastellen ist die Zahlenmenge durch

$$v d_{14} d_{13} \dots d_1 d_0 \doteq (-1)^v \cdot 2^{-3} \sum_{j=0}^{14} d_j \cdot 2^j$$

beschrieben.

- Bsp.:

VZ Vorkommateil (g) Nachkommateil (n)

$$1000\ 0000\ 0000\ 1011 \doteq -(1.011)_2 =$$
$$= (-1)^1 \cdot 2^{-3} \cdot (2^3 + 2^1 + 2^0) =$$
$$= -(1.375)_{10}$$

Festpunkt-Darstellung

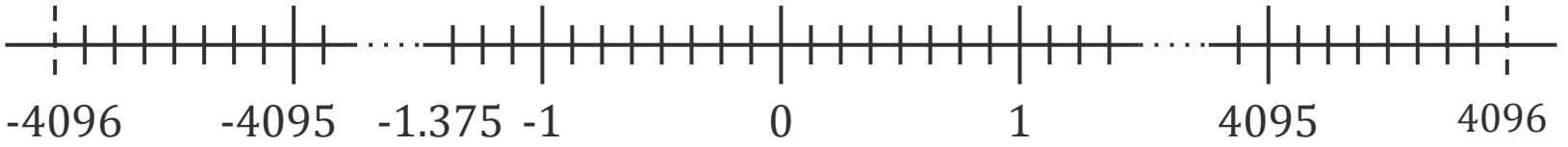
- Kleinste in diesem Zahlensystem ($N = 16, n = 3$) darstellbare Zahl

$$1111\ 1111\ 1111\ 1111 \doteq -(1111\ 1111\ 1111.111)_2 =$$
$$= (-1)^1 \cdot 2^{-3} \cdot (2^{14} + 2^{13} + \cdots + 2^1 + 2^0) =$$
$$= -(4095.875)_{10}$$

- Differenz zwischen zwei aufeinanderfolgenden Zahlen

$$0000\ 0000\ 0000\ 0001 \doteq (0.001)_2 =$$
$$= (-1)^0 \cdot 2^{-3} \cdot 2^0 =$$
$$= (0.125)_{10}$$

- Zahlenbereich $[-4095.875,\ +4095.875]$



Festpunkt-Darstellung

Beispiel

- Die Zahl $(-10.375)_{10}$ ist in das folgende (binäre) Festpunktformat umzurechnen.
- Format: $N = 12$ Bit Breite und $n = 3$ Nachkommastellen

  $\Rightarrow$ 1 Bit Vorzeichen, 8 Bit Vorkommateil, 3 Bit Nachkommateil

$$(-10.375)_{10} = (-1010.011)_2 =$$

$$= \underbrace{1}_{VZ}\underbrace{0000}_{g}\underbrace{1010011}_{n}$$

	$\cdot 2$
0.375	0
0.75	1
0.5	1

	$:2$
10	0
5	1
2	0
1	1

Festpunkt-Darstellung

Eigenschaften

- Jede Festpunktzahl ist rational ($\mathbb{Q}$)
  - d.h. irrationale Zahlen können nicht dargestellt werden
- Manche einfache rationale Zahlen können nicht genau dargestellt werden
  - z.B. $(1/3)_{10}$ dezimal dargestellt
  - bzw. $(1/10)_{10}$ binär dargestellt
- Wir müssen uns damit abfinden, dass reelle Zahlen im Rechner nur mit einer gewissen Genauigkeit dargestellt werden können
- Ergebnis einer Rechnung von 2 darstellbaren Zahlen muss nicht unbedingt darstellbar sein
  - d.h. es muss gerundet werden

Festpunkt-Darstellung

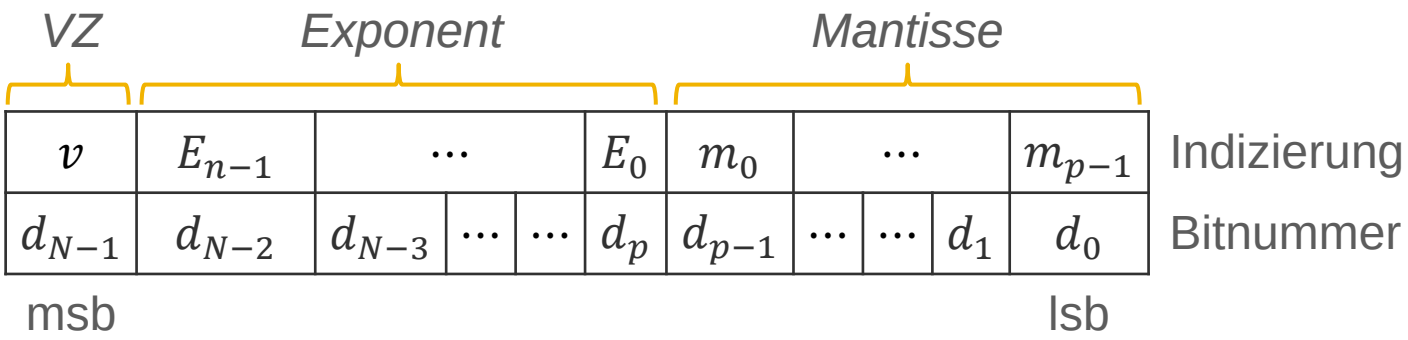
Gewünschte Eigenschaften für $\mathbb{R}$

- Große Anzahl an Nachkommastellen in der Umgebung von 0
  - sehr kleine Zahlen darstellbar
- Große Anzahl an Vorkommastellen
  - für Zahlen mit großem Absolutbetrag
- Anzahl der Nachkommastellen kann mit steigendem Absolutbetrag abnehmen, da auch ihre Bedeutung abnimmt
- $\Rightarrow$ Man benötigt ein Zahlensystem, bei dem die Anzahl der Nachkommastellen und damit die Position des Binärpunktes abhängig vom Absolutbetrag variieren (gleiten) kann: Gleitpunktzahlen

Gleitpunkt-Darstellung

- Exponentialschreibweise
  - $x = m \cdot b^e$ z.B. $0.0488 = 4.88 \cdot 10^{-2}$

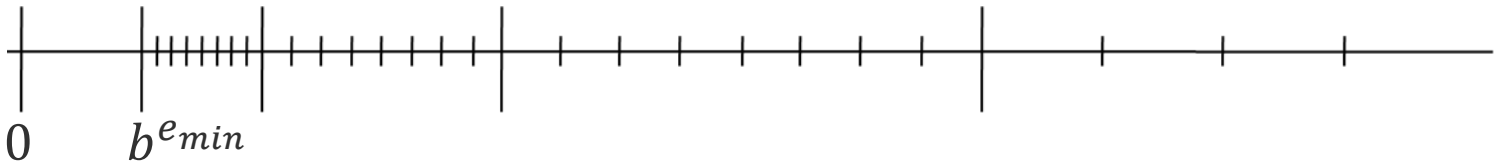
- Gesamtlänge $N = 1 + n + p$ Bit
  - Vorzeichenbit v ($0 \Rightarrow$ positiv, $1 \Rightarrow$ negativ)
  - n Stellen Exponent
  - p Stellen Mantisse



Gleitpunkt-Darstellung

Normalisierung

- Problem: Exponential-Darstellung ist mehrdeutig
  - z.B.: $0.00123 = 123 \cdot 10^{-5} = 12.3 \cdot 10^{-4} = \ ...$
- ⇒ Normalisierung notwendig
  - Normalisierungsbedingung: erste Stelle der Mantisse $(m_0) \neq 0$
    - genau 1 Stelle vor dem Komma
  - für obiges Beispiel folgt daher $0.00123 = 1.23 \cdot 10^{-3}$
- Normalisierte Zahlen auf der Zahlengerade



- Lücke um 0 unerwünscht
- 0 so nicht darstellbar: ⇒ Sonderdarstellung für 0, gilt als normalisiert

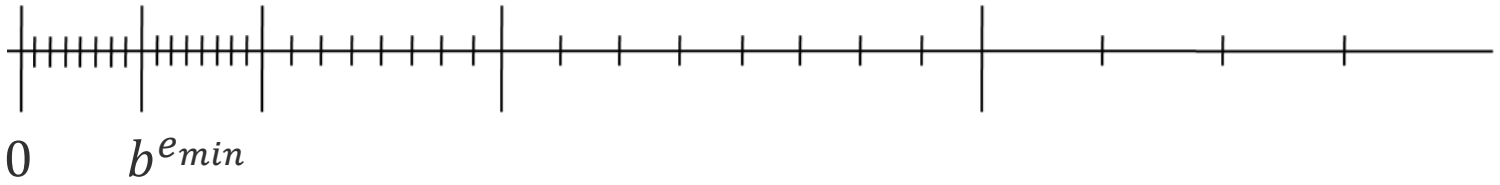
Struktur von Gleitpunkt-Zahlensystemen

Parameter eines Gleitpunkt-Zahlensystems

$$\mathbb{F}(b, p, e_{min}, e_{max}, denorm)$$

- b … Basis (base, radix) $(b \geq 2)$
- p … Mantissenlänge (precision) $(p \geq 2)$
- e_{min} … kleinster Exponent
- e_{max} … größter Exponent
- $denorm$ … Normalisierungsindikator
  - $true$ ⇒ enthält denormalisierte Zahlen
  - $false$ ⇒ enthält keine denormalisierten Zahlen

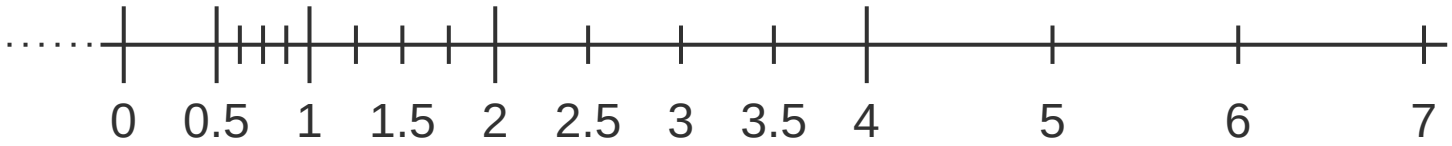
- Denormalisierte Zahlen auf der Zahlengerade



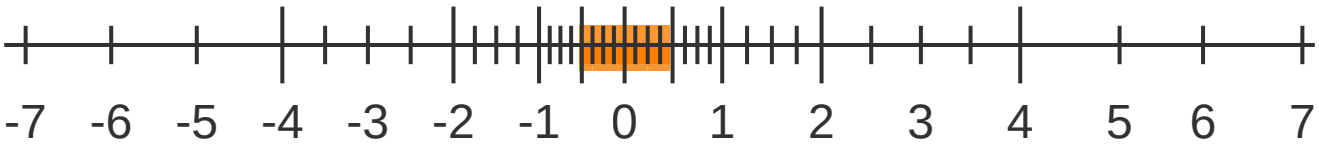
Gleitpunkt-Darstellung

Bsp.: Normalisierte und denormalisierte Gleitpunktzahlen

- Gleitpunkt-Zahlensystem mit $b = 2$, $p = 3$, $e_{min} = -1$, $e_{max} = 2$
- normalisierte Gleitpunktzahlen (positiver Teil dargestellt)



- mit denormalisierten Gleitpunktzahlen



  - Exponent für denormalisierte Zahlen: $b^{e_{min}}$
  - Durch Sonderwert $b^{e_{min}} - 1$ im Exponenten kodiert

Struktur von Gleitpunkt-Zahlensystemen

Anzahl der Gleitpunktzahlen

- Anzahl der *normalisierten* Zahlen im Gleitpunkt-Zahlensystem $\mathbb{F}(b, p, e_{min}, e_{max}, denorm)$:

Vorzeichen (+/-) *$m_0 \neq 0$* *Anzahl der möglichen Exponenten*

Die Zahl 0 $1 + 2 \cdot (b-1) \cdot b^{p-1} \cdot (e_{max} - e_{min} + 1)$

Anzahl der möglichen normalisierten Mantissen

- Anzahl der *denormalisierten* Zahlen: $2 \cdot 1 \cdot (b^{p-1} - 1)$

VZ *$m_0 = 0$* *Mantisse mit nur 0*

- IEC/IEEE Gleitpunkt-Zahlensystem $\mathbb{F}(2, 24, -126, 127, true)$:
  - $1 + 2^{24} \cdot 254 = 4261412865 \approx 4.26 \cdot 10^9$ normalisierte Zahlen
  - $2 \cdot (2^{23} - 1) = 16777214$ denormalisierte Zahlen

Struktur von Gleitpunkt-Zahlensystemen

Größte Gleitpunktzahl

- größte Gleitpunktzahl eines Gleitpunkt-Zahlensystems

$$x_{max} = M_{max} \cdot b^{e_{max}}$$

mit der Mantisse $M_{max} = (\delta.\delta\delta\cdots\delta\delta)_b$ $\qquad \delta = b - 1$

- Wert der Mantisse

p Stellen

$$\begin{array}{rccc} & \overbrace{\delta} & \ldots & \delta \\ + & & & 1 \\ \hline & 1 & 0 \ldots & 0 \end{array} \quad = b^p$$

- $M_{max} = (b^p - 1) \cdot \underbrace{b^{-(p-1)}}_{\text{Skalierung}} = (b^p - 1) \cdot b \cdot b^{-p} = b \cdot (1 - b^{-p})$

- $x_{max} = M_{max} \cdot b^{e_{max}} = b \cdot (1 - b^{-p}) \cdot b^{e_{max}}$

Struktur von Gleitpunkt-Zahlensystemen

Größte und kleinste Gleitpunktzahl

- kleinste positive normalisierte Gleitpunktzahl

$$x_{min} = M_{min} \cdot b^{e_{min}} = b^{e_{min}}$$

- IEC/IEEE Gleitpunkt-Zahlensystem $\mathbb{F}(2, 24, -126, 127, true)$:

$$x_{min} = 2^{-126} \approx 1.18 \cdot 10^{-38}$$

$$x_{max} = 2 \cdot (1 - 2^{-24}) \cdot 2^{127} \approx 3.40 \cdot 10^{38}$$

- Die kleinste positive denormalisierte Zahl eines Gleitpunkt-Zahlensystems im Falle von $denorm = true$:

$$\bar{x}_{min} = b^{e_{min}-p+1}$$

Struktur von Gleitpunkt-Zahlensystemen

Absolute Abstände der Gleitpunktzahlen

- Für eine normalisierte Gleitpunktzahl besteht die kleinste und die größte Mantisse aus den Ziffern
  - $m_0 = 1, m_1 = \cdots = m_{p-1} = 0$ *bzw.*
  - $m_0 = m_1 = \cdots = m_{p-1} = \delta = b - 1$

- Die Mantisse durchläuft somit Werte zwischen
  - $M_{min} = (1.00 \cdots 00)_b$ und
  - $M_{max} = (\delta.\delta\delta \cdots \delta\delta)_b$,

  mit einer konstanten Schrittweite von
  - $ulp = (0.00 \cdots 01)_b = b^{-p+1}$
  - Grundinkrement der Mantisse: ulp (**u**nit of **l**east **p**recision)

- Benachbarte Zahlen aus $\mathbb{F}$ haben im Intervall $[b^e, b^{e+1}]$ den konstanten Abstand

  $$\Delta x = 1\, ulp \cdot b^e = b^{e-p+1}$$

Struktur von Gleitpunkt-Zahlensystemen

Positive Zahlen aus dem Gleitpunkt-Zahlensystem $\mathbb{F}(2, 3, -1, 2, true)$

M	e	$(Wert)_2$	$(Wert)_{10}$	Intervall	Δx	denormalisiert
1.11	2	$(111)_2$	$(7)_{10}$	$[2^2,2^3)$	$(1.0)_2$	nein
1.10		$(110)_2$	$(6)_{10}$			
1.01		$(101)_2$	$(5)_{10}$			
1.00		$(100)_2$	$(4)_{10}$			
1.11	1	$(11.1)_2$	$(3.5)_{10}$	$[2^1,2^2)$	$(0.1)_2$	nein
1.10		$(11.0)_2$	$(3)_{10}$			
1.01		$(10.1)_2$	$(2.5)_{10}$			
1.00		$(10.0)_2$	$(2)_{10}$			
1.11	0	$(1.11)_2$	$(1.75)_{10}$	$[2^0,2^1)$	$(0.01)_2$	nein
1.10		$(1.10)_2$	$(1.5)_{10}$			
1.01		$(1.01)_2$	$(1.25)_{10}$			
1.00		$(1.00)_2$	$(1.0)_{10}$			
1.11	-1	$(0.111)_2$	$(0.875)_{10}$	$[2^{-1},2^0)$	$(0.001)_2$	nein
1.10		$(0.110)_2$	$(0.75)_{10}$			
1.01		$(0.101)_2$	$(0.625)_{10}$			
1.00		$(0.100)_2$	$(0.5)_{10}$			
0.11	-2	$(0.011)_2$	$(0.375)_{10}$	$(0,2^{-1})$	$(0.001)_2$	ja
0.10		$(0.010)_2$	$(0.250)_{10}$			
0.01		$(0.001)_2$	$(0.125)_{10}$			
1.00	-2	0	0	--	--	nein

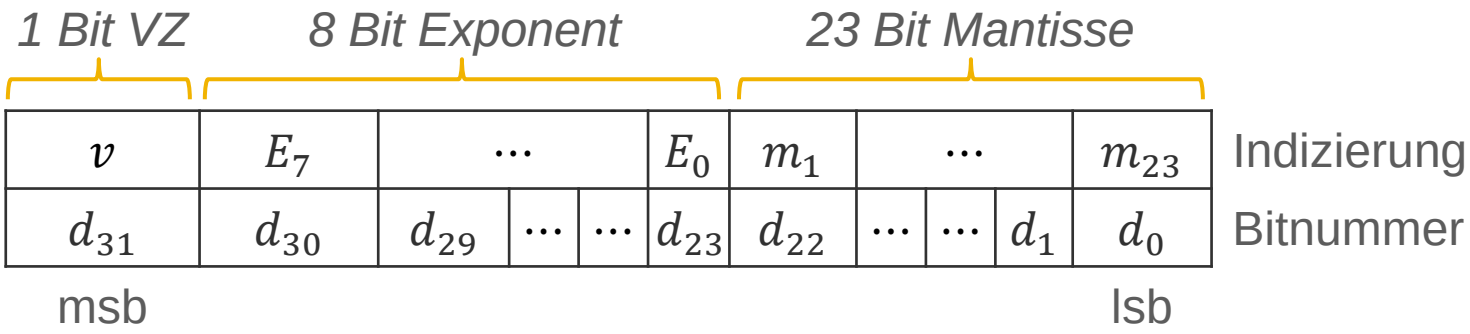
Gleitpunkt-Zahlensysteme nach IEEE 754

- $\mathbb{F}(2{,}24, -126, +127, true)$
- $\mathbb{F}(2{,}53, -1022, +1023, true)$
- Exponent ist in Exzessdarstellung

	Format			
Parameter	Single	Single Ext.	Double	Double Ext.
b	2	2	2	2
p	24	≥ 32	53	≥ 64
e_{min}	-126	≤ -1022	-1022	≤ -16382
e_{max}	$+127$	$\geq +1023$	$+1023$	$\geq +16383$
denorm	$true$	$true$	$true$	$true$
Exzess des Exponenten	$+127$	$unspez.$	$+1023$	$unspez.$
Bitbreite des Exponenten	8	≥ 11	11	≥ 15
Bitbreite des Formats	32	≥ 43	64	≥ 79

Gleitpunkt-Zahlensysteme nach IEEE 754

Codierung nach IEEE 754 Single Precision Format (1 von 3)



- Normalisierte Gleitpunktzahlen:
  - Normalisierungsbedingung: $m_0 \neq 0$ daher immer $m_0 = 1$

    ⇒ Vorkommastelle m_0 wird weggelassen: „Implizites erstes Bit“

- Denormalisierte Gleitpunktzahlen:
  - spezieller Exponentenwert $e_{min} - 1$ zeigt implizites erstes Bit $m_0 = 0$ an
  - als Exponentenwert gilt: e_{min}

Gleitpunkt-Zahlensysteme nach IEEE 754

Codierung nach IEEE 754 Single Precision Format (2 von 3)

Die Zahl Null

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

- $+0, -0$

Not a Number (NaN)

- Sonderwert für „Ergebnisse“ nicht möglicher Berechnungen:
  - $\frac{0}{0}$
  - $\sqrt{-1}$
- $e = e_{max} + 1 = 128$
- Darstellung: alle Exponentenbits sind 1, Mantisse > 0, z.B.:

0	1	1	1	1	1	1	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Gleitpunkt-Zahlensysteme nach IEEE 754

Codierung nach IEEE 754 Single Precision Format (3 von 3)

Unendlich

- Ergebnis ist betragsmäßig zu groß, um im betreffenden Format dargestellt werden zu können ⇒ „Überlauf“
  - es wird auf $+\infty$ bzw. $-\infty$ als Rückgabewert zurückgegriffen
    - $e = e_{max} + 1 = 128$

$+\infty$

0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

$-\infty$

1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

$$\sqrt{+\infty} = +\infty$$

$$\frac{1}{+\infty} = +0 \quad \text{bzw.} \quad \frac{1}{-\infty} = -0$$

$$4 - \infty = -\infty$$

$$\frac{1}{+0} = +\infty \quad \text{bzw.} \quad \frac{1}{-0} = -\infty$$

Gleitpunkt-Zahlensysteme nach IEEE 754

Die Grundformate einfacher und doppelter Genauigkeit

Format		Exponent allgemein	Exponent dezimal	NKSt. f der Mant.	Wert der Gleitpunktzahl
single	1	$e_{max}+1$	128	$f \neq 0$	NaN
	2	$e_{max}+1$	128	$f = 0$	$(-1)^v \cdot \infty$
	3	$e_{min} \leq e \leq e_{max}$	$-126 \leq e \leq 127$	$beliebig$	$(-1)^v \cdot 1.f \cdot 2^e$
	4	$e_{min}-1$	-127	$f \neq 0$	$(-1)^v \cdot 0.f \cdot 2^{-126}$
	5	$e_{min}-1$	-127	$f = 0$	$(-1)^v \cdot 0$
double	1	$e_{max}+1$	1024	$f \neq 0$	NaN
	2	$e_{max}+1$	1024	$f = 0$	$(-1)^v \cdot \infty$
	3	$e_{min} \leq e \leq e_{max}$	$-1022 \leq e \leq 1023$	$beliebig$	$(-1)^v \cdot 1.f \cdot 2^e$
	4	$e_{min}-1$	-1023	$f \neq 0$	$(-1)^v \cdot 0.f \cdot 2^{-1022}$
	5	$e_{min}-1$	-1023	$f = 0$	$(-1)^v \cdot 0$

v … VZ-Bit der Mantisse

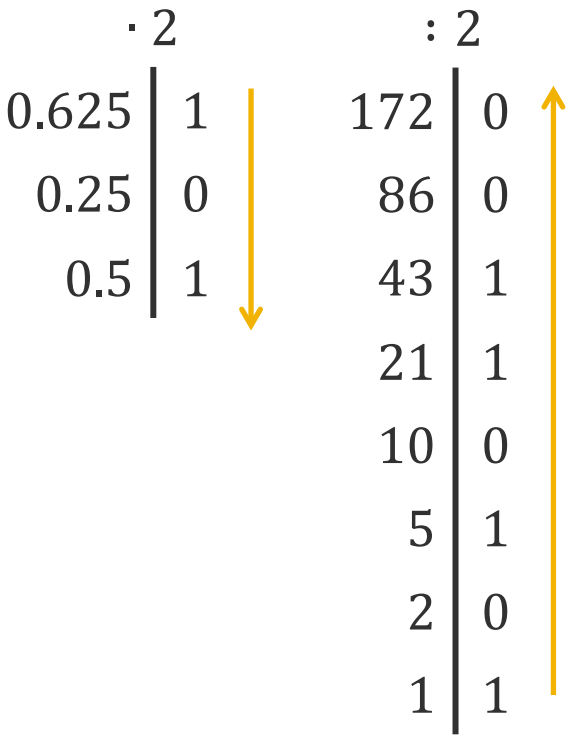
Gleitpunkt-Zahlensysteme nach IEEE 754

Bsp: Codierung einer Dezimalzahl in das IEEE754-Format (1 von 2)

- Wandeln Sie die Zahl $(-172.625)_{10}$ in das IEEE 754 Single Precision Format um!

- (1) Umwandeln ins Binärsystem

$$(-172.625)_{10} = (-10101100.101)_2$$



Gleitpunkt-Zahlensysteme nach IEEE 754

Bsp: Codierung einer Dezimalzahl in das IEEE754-Format (2 von 2)

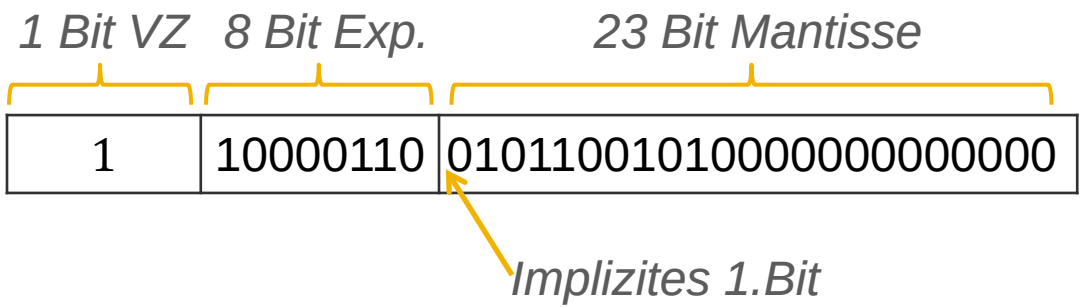
- (2) Normalisierung

$$(10101100.101)_2 \cdot 2^0 = (1.0101100101)_2 \cdot 2^7$$

- (3) Exponent berechnen

$$\begin{array}{rl} 0\ 1\ 1\ 1\ \ 1\ 1\ 1\ 1 = (127)_{10} & \textit{Exzess} \\ +\ {}_1 0\ {}_1 0\ {}_1 0\ {}_1 0\ \ {}_1 0\ {}_1 1\ {}_1 1\ {}_1 1 = (7)_{10} & \textit{Exponent d. normalisierten Darstellung} \\ \hline 1\ 0\ 0\ 0\ \ 0\ 1\ 1\ 0 = (134)_{10} & \textit{Exponent in Exzessdarstellung} \end{array}$$

- (4) Vorzeichenbit setzen
  - negative Zahl ⇒ 1

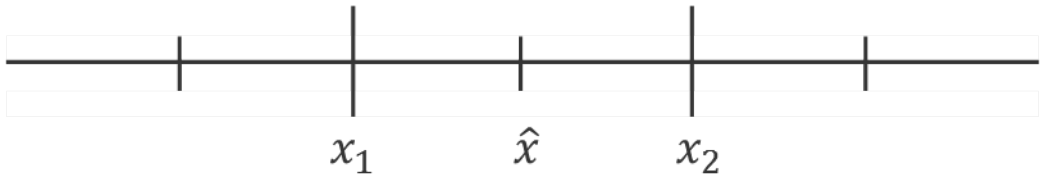


Arithmetik auf Gleitpunkt-Zahlensystemen

Runden (1 von 5)

- unendlich viele reelle Zahlen $\mathbb{R}$ $\Longleftrightarrow$
  endlich viele in einem Computer darstellbare Gleitpunktzahlen $\mathbb{F}$

- Mittels Rundungsfunktion $\square$ reelle Zahl auf Gleitpunktzahl abbilden
  - Abbildung $\square: \mathbb{R} \rightarrow \mathbb{F}$,
    die jeder reellen Zahl $x \in \mathbb{R}$
    eine bestimmte „benachbarte“ Zahl $\square x \in \mathbb{F}$ zuordnet



  - $x_1,\ x_2 \in \mathbb{F},\ \hat{x}$ ... Grenzwert
  - **Rundungsfunktion** $\square$ **bestimmt** als Ergebnis der Rundung $\square x$ **einen der beiden Werte** x_1 oder x_2

Arithmetik auf Gleitpunkt-Zahlensystemen

Runden (2 von 5)

- Berechnungen mit Zahlen aus $\mathbb{F}$: Ergebnis meist keine Zahl aus $\mathbb{F}$

  $\Rightarrow$ Runden des Ergebnisses auf eine Zahl aus $\mathbb{F}$ notwendig

- Zu jeder zweistelligen arithmetischen Operation $\circ\colon \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ definiert man die gerundete Operation $\boxed{\circ}\colon \mathbb{F} \times \mathbb{F} \to \mathbb{F}$

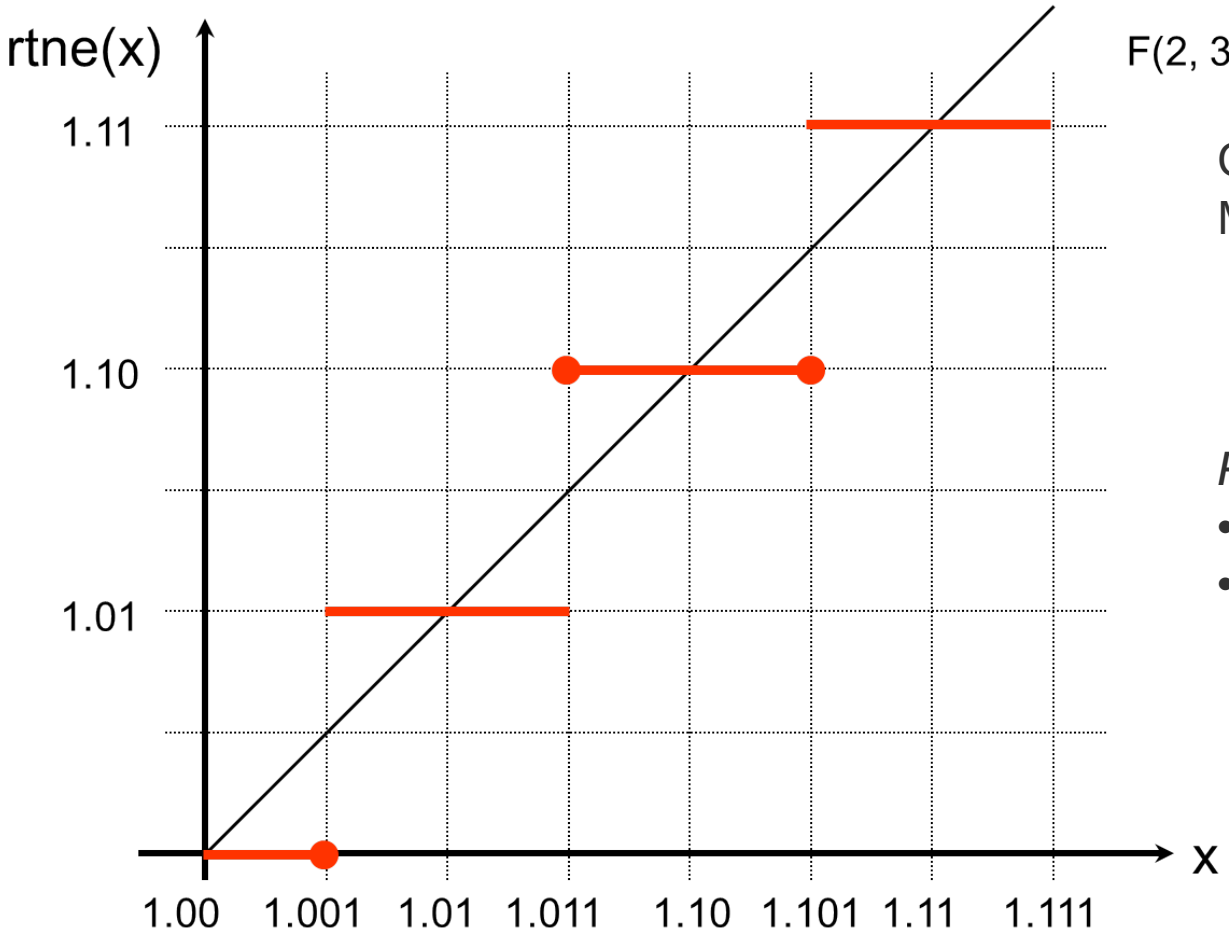
  $$x \boxed{\circ} y = \square(x \circ y)$$

- Eigenschaften einer Rundungsoperation $\square\colon \mathbb{R} \to \mathbb{F}$
  - $\square x = x$
    Eine Gleitpunktzahl wird auf sich selbst gerundet (*Projektivität*)
  - $x \leq y \Longrightarrow \square x \leq \square y$
    Die Relation $x \leq y$ bleibt auch nach der Rundung erhalten (*Monotonie*)

Arithmetik auf Gleitpunkt-Zahlensystemen

Runden (3 von 5)

- Optimale Rundung (round to nearest)



F(2, 3, -1, 2, true)

Grenzpunkt liegt genau in der Mitte:

$$\hat{x} = \frac{x_1 + x_2}{2}$$

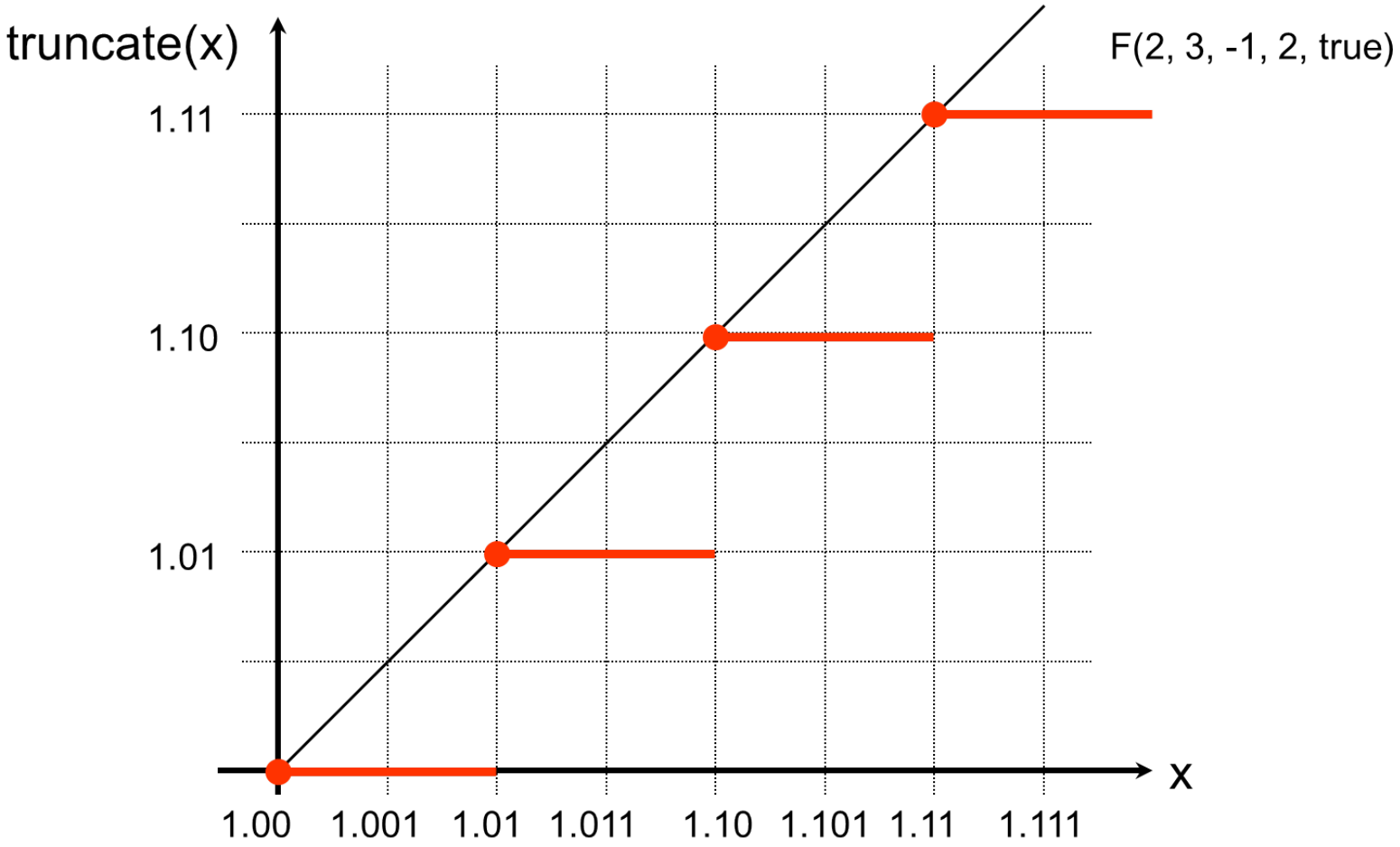
Falls $x = \hat{x}$, 2 Möglichkeiten:

- *Round away from zero*
- *Round to even*
  auf den Nachbarn runden, dessen letzte Mantissenstelle gerade ist

Arithmetik auf Gleitpunkt-Zahlensystemen

Runden (4 von 5)

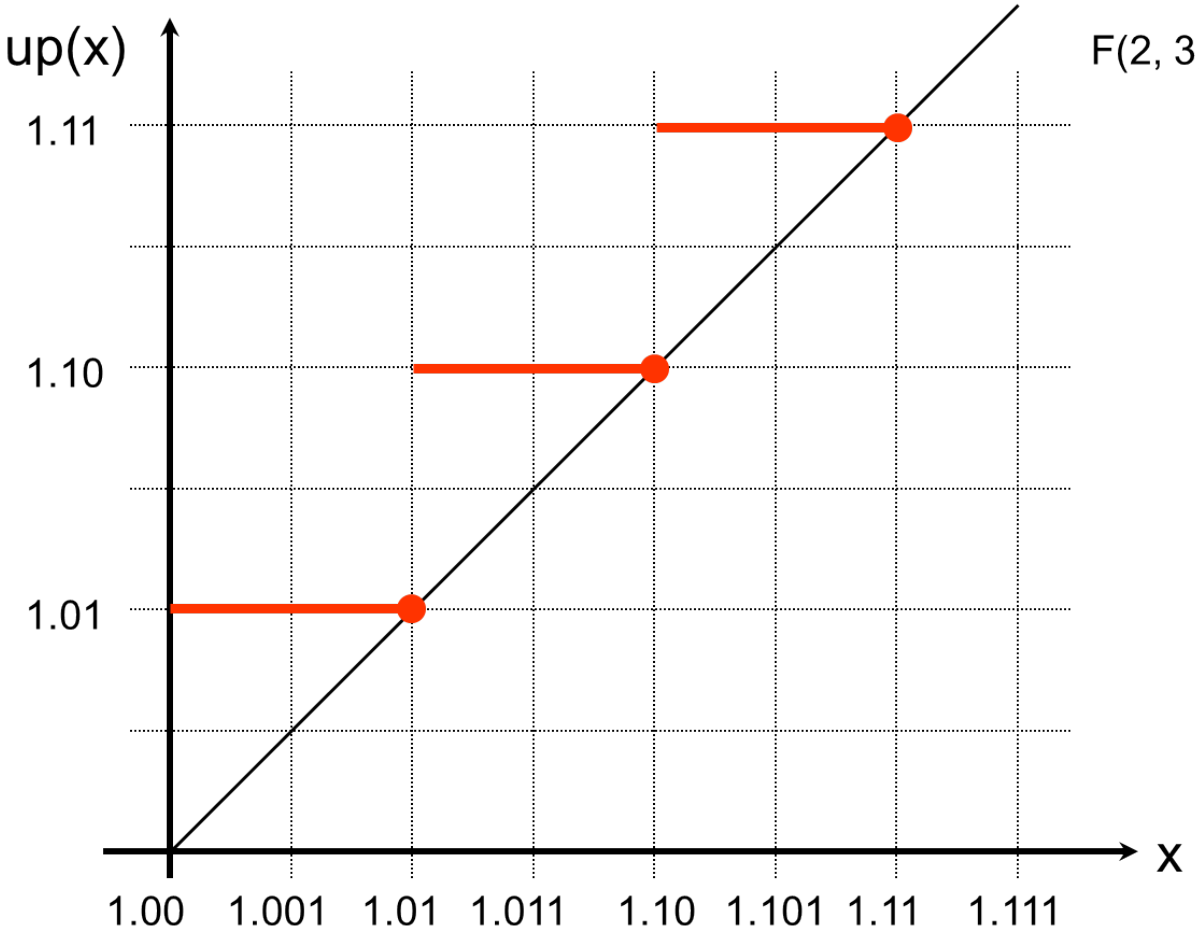
- Abschneiden (truncate)



Arithmetik auf Gleitpunkt-Zahlensystemen

Runden (5 von 5)

- Gerichtetes Runden (directed rounding)



Aufrunden: $\square x = \max(x_1, x_2)$

Abrunden: $\square x = \min(x_1, x_2)$

Arithmetik auf Gleitpunkt-Zahlensystemen

Runden – Beispiel 1

- Bsp.: $(-1.626)_{10}$ auf 2 (dezimale) Nachkommastellen
- Optimale Rundung (round to nearest)
  - $x_1 = -1.62$, $x_2 = -1.63$, $\hat{x} = -1.625$, $\square\, x = -1.63$
- Abschneiden (truncate)
  - $\square\, x = -1.62$
- Gerichtetes Runden (directed rounding)
  - Aufrunden $\square\, x = -1.62$
  - Abrunden $\square\, x = -1.63$

Arithmetik auf Gleitpunkt-Zahlensystemen

Runden – Beispiel 2

Bsp.: $(1.101)_2 = (1.625)_{10}$ auf 2 (binäre) Nachkommastellen

- Optimale Rundung (round to nearest)
  - $\hat{x} = (1.101)_2, x_1 = (1.10)_2,\ x_2 = (1.11)_2$
  - Zahl liegt genau am Grenzwert, daher weitere Rundungsregel notwendig
    - *Round away from zero:* $\square\, x = (1.11)_2 = (1.75)_{10}$
    - *Round to even:* $\square\, x = (1.10)_2 = (1.5)_{10}$
- Abschneiden (truncate)
  - $\square\, x = (1.10)_2 = (1.5)_{10}$
- Gerichtetes Runden (directed rounding)
  - Aufrunden
    - $(1.11)_2 = (1.75)_{10}$
  - Abrunden
    - $(1.10)_2 = (1.5)_{10}$

Arithmetik auf Gleitpunkt-Zahlensystemen

Rundungsfehler (1 von 2)

- absoluter Rundungsfehler $\varepsilon(x) = \varepsilon\square(x) = \square x - x$
- relativer Rundungsfehler $\rho(x) = \frac{\square x - x}{x} = \frac{\varepsilon(x)}{x}$

Bsp.: $(1.101)_2 = (1.625)_{10}$ auf 2 (binäre) Nachkommastellen

…

- *Round away from zero:* $\square\, x = (1.11)_2 = (1.75)_{10}$
  - $\varepsilon(x) = (1.75)_{10} - (1.625)_{10} = (0.125)_{10}$
- *Round to even:* $\square\, x = (1.10)_2 = (1.5)_{10}$
  - $\varepsilon(x) = (1.5)_{10} - (1.625)_{10} = -(0.125)_{10}$

Arithmetik auf Gleitpunkt-Zahlensystemen

Rundungsfehler - Beispiel (1von 2)

- Bsp. $x = a + b + c$

- Demonstration anhand von $\mathbb{F}(10,3,-9,10,false)$, $a = 1.05 \times 10^3$, $b = c = 4.55 \times 10^0$, optimale Rundung

- Auswertungsreihenfolge von links nach rechts:

$$
\begin{aligned}
(a \boxplus b) \boxplus c &= \square(a + b) \boxplus c = \\
&= \square(\square(a + b) + c) = \\
&= \square(\square(1.05 \times 10^3 + 4.55 \times 10^0) + 4.55 \times 10^0) = \\
&= \square(\square(1.05455 \times 10^3) + 4.55 \times 10^0) = \\
&= \square(1.05 \times 10^3 + 4.55 \times 10^0) = \\
&= \square(1.05455 \times 10^3) = \\
&= 1.05 \times 10^3
\end{aligned}
$$

Arithmetik auf Gleitpunkt-Zahlensystemen

Rundungsfehler - Beispiel (2 von 2)

- Auswertungsreihenfolge von rechts nach links

$$
\begin{aligned}
a \boxplus (b \boxplus c) &= a \boxplus (\square(b + c)) = \\
&= \square(a + (\square(b + c))) = \\
&= \square(1.05 \times 10^3 + (\square(4.55 \times 10^0 + 4.55 \times 10^0))) = \\
&= \square(1.05 \times 10^3 + (\square(9.10 \times 10^0))) = \\
&= \square(1.05 \times 10^3 + 9.10 \times 10^0) = \\
&= \square(1.0591 \times 10^3) = \\
&= 1.06 \times 10^3
\end{aligned}
$$

Arithmetik auf Gleitpunkt-Zahlensystemen

Pseudo-Arithmetik

- keine Gültigkeit der Assoziativität

$$a \boxplus (b \boxplus c) \neq (a \boxplus b) \boxplus c$$

$$a \boxtimes (b \boxtimes c) \neq (a \boxtimes b) \boxtimes c$$

- keine Gültigkeit der Distributivität

$$a \boxtimes (b \boxplus c) \neq (a \boxtimes b) \boxplus (a \boxtimes c)$$

- aber wegen

$$a \boxplus b = \square(a + b) = \square(b + a) = b \boxplus a$$

und

$$a \boxtimes b = \square(a \cdot b) = \square(b \cdot a) = b \boxtimes a$$

⇒ Kommutativität

Arithmetik auf Gleitpunkt-Zahlensystemen

Rundung und Vergleich

- Da verschiedene reelle Zahlen bei der Rundung nach $\mathbb{F}$ in dieselbe Gleitpunktzahl übergehen können, ist es im Allgemeinen keine gute Idee, Gleitpunktzahlen auf $= 0$ abzufragen.

- Um festzustellen, ob der exakte Wert eines arithmetischen Ausdrucks positiv ist, muss man verlangen, dass seine Auswertung in $\mathbb{F}$ weit genug von Null entfernt ist:

$$Ausdruck \geq \alpha > 0$$

Arithmetik auf Gleitpunkt-Zahlensystemen

Iterative Summation

- Berechnung einer Näherung der unendlichen Summe

$$\sum_{i \geq 1} \frac{1}{i^2}$$

- indem man die Summanden für $i = 1, 2, 3, \ldots$ aufaddiert
- Die Zwischensummen werden immer größer und die Summanden immer kleiner
- Man gelangt zu einem bestimmten N, ab dem

$$\sum_{i=1}^{N} \frac{1}{i^2} \boxplus \frac{1}{(N+1)^2} = \sum_{i=1}^{N} \frac{1}{i^2}$$

  d.h., dass die Summe ihren Wert nicht mehr ändert.

- beginnt man mit $i = N, N-1, N-2, \ldots$, so erhält man sogar einen genaueren Näherungswert $\left(\frac{\pi^2}{6}\right)$

Arithmetik auf Gleitpunkt-Zahlensystemen

Addition – Vorgehensweise

1. Exponenten angleichen
   - größeren Exponenten bestimmen
   - kleineren Exponenten an den größeren anpassen
   - entsprechende Mantisse verschieben
2. Mantissen addieren
3. Normalisieren
4. Runden

Arithmetik auf Gleitpunkt-Zahlensystemen

Gleitpunkt-Arithmetik (1 von 2)

Bsp.1: $2.15 \times 10^{12} - 1.25 \times 10^{-5}$

1. Exponenten angleichen: $1.25 \times 10^{-5} = 0.00000000000000000125 \times 10^{12}$
2. Mantissen addieren

$$\begin{array}{rll} & 2.15 & \times\ 10^{12} \\ - & 0.00000000000000000125 & \times\ 10^{12} \\ \hline & 2.14999999999999875 & \times\ 10^{12} \end{array}$$

3. Normalisieren entfällt
4. Runden (3 Stellen Mantisse): 2.15×10^{12}

- vor Berechnung abschneiden?

$$\begin{array}{rll} & 2.15 & \times\ 10^{12} \\ - & 0.00 & \times\ 10^{12} \\ \hline & 2.15 & \times\ 10^{12} \end{array}$$

Runden:
round to nearest mit
round to even

Arithmetik auf Gleitpunkt-Zahlensystemen

Gleitpunkt-Arithmetik (2 von 2)

- Bsp.2: $10.1 - 9.93$
  - $10.1 \times 10^0 = 1.01 \times 10^1$
  - $9.93 \times 10^0 = 0.993 \times 10^1$

$$\begin{array}{rrl} 1.01 & \times\ 10^1 & \\ -\ 0.993 & \times\ 10^1 & \\ \hline 0.017 & \times\ 10^1 & = 0.17 \times 10^0 \end{array}$$

- vor Berechnung abschneiden?

$$\begin{array}{rrl} 1.01 & \times\ 10^1 & \\ -\ 0.99 & \times\ 10^1 & \\ \hline 0.02 & \times\ 10^1 & = 0.20 \times 10^0 \end{array}$$

Zusätzliche Stellen notwendig!

Wie viele?

Runden:
round to nearest mit
round to even

Arithmetik auf Gleitpunkt-Zahlensystemen

Gleitpunkt-Arithmetik: Guard Digit

- 1 zusätzliche Stelle: *Guard Digit (g)*

- Bsp.: $1.01 \times 10^1 - 0.993 \times 10^1$
  - Ergebnis berechnen, dann runden (3 Stellen Mantisse):

$$\begin{array}{rl} 1.01 & \times\ 10^1 \\ -\,0.993 & \times\ 10^1 \\ \hline 0.017 & \times\ 10^1 \end{array} \quad = 0.17 \times 10^0$$

  - 3 Stellen Mantisse + Guard Digit,
    Restliches abschneiden, dann Ergebnis berechnen

$$\begin{array}{rl} 1.01 & \times\ 10^1 \\ -\,0.993 & \times\ 10^1 \\ \hline 0.017\ g & \times\ 10^1 \end{array} \quad = 0.17 \times 10^0$$

Runden:
round to nearest mit
round to even

Arithmetik auf Gleitpunkt-Zahlensystemen

Gleitpunkt-Arithmetik: Guard Digit

- Bsp.: $1.01 \times 10^2 - 3.76 \times 10^0$
  - Ergebnis berechnen, dann runden (3 Stellen Mantisse):

$$\begin{array}{rll} 1.01 & \times & 10^2 \\ -0.0376 & \times & 10^2 \\ \hline 0.9724 & \times & 10^2 \end{array} \approx 97.2 \times 10^0$$

  - 3 Stellen Mantisse + Guard Digit
    Restliches abschneiden, dann Ergebnis berechnen

$$\begin{array}{rll} 1.01 & \times & 10^2 \\ -0.037 & \times & 10^2 \\ \hline 0.973 & \times & 10^2 \\ g & & \end{array} \approx 97.3 \times 10^0$$

→ 1 zusätzliche Stelle reicht nicht...

Runden:
round to nearest mit
round to even

Arithmetik auf Gleitpunkt-Zahlensystemen

Gleitpunkt-Arithmetik: Round Digit

- noch 1 zusätzliche Stelle: *Round Digit*

- Bsp.: $1.01 \times 10^2 - 3.76 \times 10^0$
  - Ergebnis berechnen, dann runden (3 Stellen Mantisse):

$$\begin{array}{rl} 1.01 & \times\ 10^2 \\ -0.0376 & \times\ 10^2 \\ \hline 0.9724 & \times\ 10^2 \end{array} \approx 97.2 \times 10^0$$

  - 3 Stellen Mantisse + Guard Digit + Round Digit
    Restliches abschneiden, dann Ergebnis berechnen

$$\begin{array}{rl} 1.01 & \times\ 10^2 \\ -0.0376 & \times\ 10^2 \\ \hline 0.9724 & \times\ 10^2 \\ g\ r & \end{array} \approx 97.2 \times 10^0$$

Runden:
round to nearest mit
round to even

Arithmetik auf Gleitpunkt-Zahlensystemen

Gleitpunkt-Arithmetik: Round Digit

- Bsp.: $4.5674 \times 10^0 + 2.5003 \times 10^{-4}$
  - Ergebnis berechnen, dann runden (5 Stellen Mantisse):

$$\begin{array}{rll} 4.5674 & \times & 10^0 \\ +0.00025003 & \times & 10^0 \\ \hline 4.56765003 & \times & 10^0 \end{array} \approx 4.5677 \times 10^0$$

  - 5 Stellen Mantisse + Guard Digit + Round Digit
    Restliches abschneiden, Ergebnis berechnen

$$\begin{array}{rll} 4.5674 & \times & 10^0 \\ +0.000250 & \times & 10^0 \\ \hline 4.567650 & \times & 10^0 \\ \quad g\ r & & \end{array} \approx 4.5676 \times 10^0$$

→ 2 zusätzliche Stellen reichen nicht...

Runden:
round to nearest mit
round to even

Arithmetik auf Gleitpunkt-Zahlensystemen

Gleitpunkt-Arithmetik Sticky Bit

- 3. zusätzliche Stelle, damit man korrekt runden kann

  ⇒ 1 zusätzliches Bit: *Sticky Bit*

  verändert sich nicht mehr, sobald es einmal den Wert 1 angenommen hat!!

  - Kommen rechts vom Round Digit noch Stellen $\neq 0$?

- Sticky Bit eigentlich true/false
  - true … es gibt rechts vom Round Digit noch Stellen $\neq 0$
  - false … es gibt rechts vom Round Digit keine Stellen $\neq 0$
  - …wird mit 1(true) bzw. 0(false) kodiert

Arithmetik auf Gleitpunkt-Zahlensystemen

Addition/Subtraktion – Vorgehensweise

- (1) Angleichung der Exponenten
- (2) Mantissen addieren/subtrahieren
  - Vorzeichen gleich: Addition
  - Vorzeichen ungleich: Subtraktion
- (3) Ergebnis normalisieren
- (4) Runden (Guard Digit, Round Digit, Sticky Bit)

- Optimale Rundung

G	R	S	Ergebnis / Mantisse
0	x	x	unverändert
1	1	x	+= 1
1	0	0	**Weitere** Rundungsregel für Grenzfall nötig!
1	0	1	Vorzeichen gleich ⇒ Ergebnis += 1 Vorzeichen unterschiedlich ⇒ unverändert

Arithmetik auf Gleitpunkt-Zahlensystemen

Addition/Subtraktion – Vorgehensweise Runden

- Optimale Rundung / round to even

G	R	S	Ergebnis / Mantisse
1	0	0	wenn lsb = 0 ⇒ unverändert wenn lsb = 1 ⇒ += 1

- Optimale Rundung / round away from zero

G	R	S	Ergebnis / Mantisse
1	0	0	+= 1

Arithmetik-Beispiele nach (modifiziertem) IEEE 754

Beispiel $A + B,\ A - B$

Bsp. Addition und Subtraktion :

- $A = (5.58)_{10}$ und $B = (62.27)_{10}$
- Umrechnung ins Gleitpunktformat
  nach verkürztem und leicht verändertem IEEE 754-Standard:
  - 5 Bit Exponent,
  - 10 Bit Mantisse mit explizitem (!) ersten Bit,
  - Exzess = 15
  - Runden durch Abschneiden

- $A + B,\ A - B$ mit optimaler Rundung mit "round to even"

Arithmetik-Beispiele nach (modifiziertem) IEEE 754

Umrechnung von A ins Gleitpunktformat

Umrechnung ins Gleitpunktformat

- Zahl $A = (5.58)_{10}$
  - Umwandeln ins Binärsystem

    $$(5.58)_{10} = (101.1001010)_2$$

  - Normalisieren

    $$(101.1001010)_2 \times 2^0 = (1.011001010)_2 \times 2^2$$

  - Exponent berechnen

    $$\begin{array}{rl} 0\ 1\ 1\ 1\ 1 = (15)_{10} & \textit{Exzess} \\ +\ 0\ 0\ 0\ 1\ 0 = (2)_{10} & \textit{Exponent der normalisierten Darstellung} \\ \hline 1\ 0\ 0\ 0\ 1 = (17)_{10} & \textit{Exponent in Exzessdarstellung} \end{array}$$

  - Vorzeichenbit: 0

Arithmetik-Beispiele nach (modifiziertem) IEEE 754

Umrechnung von B ins Gleitpunktformat

- Zahl $B = (62.27)_{10}$
  - Umwandeln ins Binärsystem

    $$(62.27)_{10} = (111110.0100)_2$$

  - Normalisieren

    $$(111110.0100)_2 \times 2^0 = (1.111100100)_2 \times 2^5$$

  - Exponent berechnen

    $$\begin{array}{rl} 0\,1\,1\,1\,1 = (15)_{10} & \textit{Exzess} \\ +\,0\,0\,1\,0\,1 = (5)_{10} & \textit{Exponent der normalisierten Darstellung} \\ \hline 1\,0\,1\,0\,0 = (20)_{10} & \textit{Exponent in Exzessdarstellung} \end{array}$$

  - Vorzeichenbit: 0

	VZ	*Exponent*					*Mantisse*									
A	0	1	0	0	0	1	1	0	1	1	0	0	1	0	1	0
B	0	1	0	1	0	0	1	1	1	1	1	0	0	1	0	0

Arithmetik-Beispiele nach (modifiziertem) IEEE 754

Addition $A + B$ (1 von 3)

- (1) Angleichung der Exponenten
  - A < B ⇒ Exponent von A an Exponent von B anpassen
  - „Hinausgeschobene“ Bits füllen Guard/Round/Sticky auf

	VZ	*Exponent*					*Mantisse*									
A	0	1	0	0	0	1	1	0	1	1	0	0	1	0	1	0
B	0	1	0	1	0	0	1	1	1	1	1	0	0	1	0	0

Exp.+3 *um 3 Bit „nach hinten geschoben“*

	VZ	*Exponent*					*Mantisse*										*G*	*R*	*S*
A	0	1	0	1	0	0	0	0	0	1	0	1	1	0	0	1	0	1	0
B	0	1	0	1	0	0	1	1	1	1	1	0	0	1	0	0			

Arithmetik-Beispiele nach (modifiziertem) IEEE 754

Addition $A + B$ (2 von 3)

- (2) Mantissen addieren

	VZ	Exponent					Mantisse										G	R	S
A	0	1	0	1	0	0	0	0	0	1	0	1	1	0	0	1	0	1	0
+ B	0	1	0	1	0	0	1	1	1	1	1	0	0	1	0	0			
	0	1	0	1	0	0	10	0	0	0	1	1	1	1	0	1	0	1	0

- (3) Ergebnis normalisieren

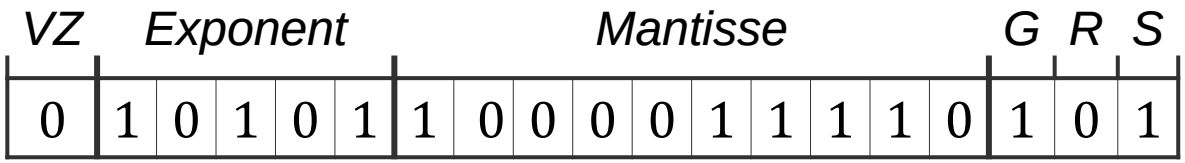
	VZ	Exponent					Mantisse										G	R	S
	0	1	0	1	0	0	10	0	0	0	1	1	1	1	0	1	0	1	0
	0	1	0	1	0	1	1	0	0	0	0	1	1	1	1	0	1	0	1

Exp.+1 *um 1 Bit „nach hinten geschoben“*

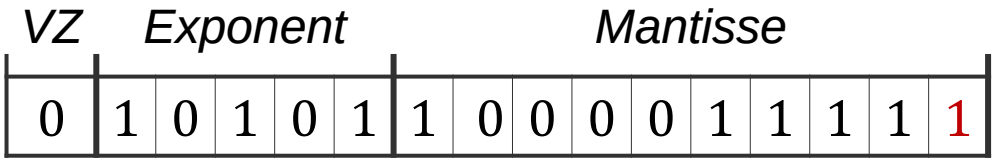
Arithmetik-Beispiele nach (modifiziertem) IEEE 754

Addition $A + B$ (3 von 3)

- (4) Runden



Runden



G	R	S	Ergebnis (Erg.)
0	x	x	unverändert
1	1	x	Ergebnis += 1
1	0	0	lsb = 0 ⇒ unverändert lsb = 1 ⇒ Erg. += 1
1	0	1	VZ gleich ⇒ Erg. += 1 VZ untersch. ⇒ unverändert

⇒ Ergebnis:

- $0\ 10101\ 1000011111$ bzw. $(1.000011111)_2 \times 2^6$

Arithmetik-Beispiele nach (modifiziertem) IEEE 754

Subtraktion $A - B$ (1 von 3)

- (1) Angleichung der Exponenten
  - A < B ⇒ Exponent von A an Exponent von B anpassen
  - „Hinausgeschobene“ Bits füllen Guard/Round/Sticky auf

	VZ	*Exponent*					*Mantisse*									
A	0	1	0	0	0	1	1	0	1	1	0	0	1	0	1	0
B	0	1	0	1	0	0	1	1	1	1	1	0	0	1	0	0

Exp.+3 *um 3 Bit „nach hinten geschoben“*

	VZ	*Exponent*					*Mantisse*										*G*	*R*	*S*
A	0	1	0	1	0	0	0	0	0	1	0	1	1	0	0	1	0	1	0
B	0	1	0	1	0	0	1	1	1	1	1	0	0	1	0	0			

Arithmetik-Beispiele nach (modifiziertem) IEEE 754

Subtraktion $A - B$ (2 von 3)

- (2) Mantissen subtrahieren
  - $B > A \Rightarrow$ wir wissen, dass Ergebnis negativ sein wird
  - Trick um uns Rechnen über 0 zu ersparen: berechnen $B - A$ und setzen Ergebnis negativ $\Rightarrow A - B = -(B - A)$

	VZ	*Exponent*					*Mantisse*										*G*	*R*	*S*
B	0	1	0	1	0	0	1	1	1	1	1	0	0	1	0	0			
– A	0	1	0	1	0	0	0	0	0	1	0	1	1	0	0	1	0	1	0
	1	1	0	1	0	0	1	1	1	0	0	0	1	0	1	0	1	1	0

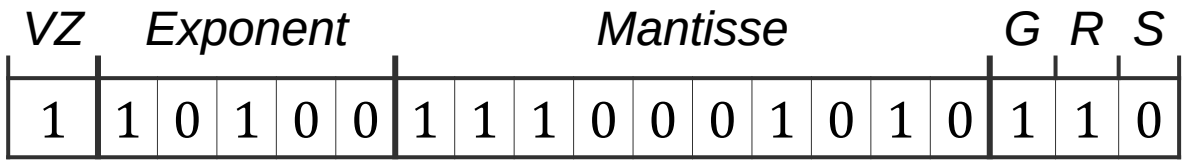
- (3) Ergebnis normalisieren
  - ist bereits normalisiert!

VZ	*Exponent*					*Mantisse*										*G*	*R*	*S*
1	1	0	1	0	0	1	1	1	0	0	0	1	0	1	0	1	1	0

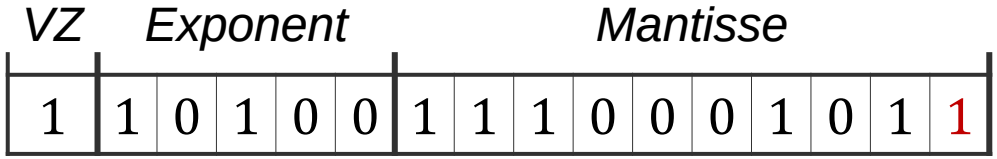
Arithmetik-Beispiele nach (modifiziertem) IEEE 754

Subtraktion $A - B$ (3 von 3)

- (4) Runden



Runden



G	R	S	Ergebnis (Erg.)
0	x	x	unverändert
1	1	x	Ergebnis += 1
1	0	0	lsb = 0 ⇒ unverändert lsb = 1 ⇒ Erg. += 1
1	0	1	VZ gleich ⇒ Erg. += 1 VZ untersch. ⇒ unverändert

⇒ Ergebnis:

- $1\ 10100\ 1110001011$ bzw. $(-1.110001011)_2 \times 2^5$

Arithmetik auf Gleitpunkt-Zahlensystemen

Multiplikation - Vorgehensweise

- (1) Multiplikation der Mantissen
- (2) Summe der Exponenten
- (3) Normalisieren
- (4) Runden

Arithmetik-Beispiele nach (modifiziertem) IEEE 754

Multiplikation $A \cdot B$ (1 von 4)

	VZ	*Exponent*					*Mantisse*									
A	0	1	0	0	0	1	1	0	1	1	0	0	1	0	1	0
B	0	1	0	1	0	0	1	1	1	1	1	0	0	1	0	0

- (1) Multiplikation der Mantissen

1	0	1	1	0	0	1	0	1	0	·	1	1	1	1	1	0	0	1	0	0
1	0	1	1	0	0	1	0	1	0											
	1	0	1	1	0	0	1	0	1	0										
		1	0	1	1	0	0	1	0	1	0									
			1	0	1	1	0	0	1	0	1	0								
				1	0	1	1	0	0	1	0	1	0							
							1	0	1	1	0	0	1	0	1	0				
									0	0	0	0	0	0	0	0	0	0		
10	1	0	1	1	0	1	1	0	0	1	1	1	1	0	1	0	0	0		

Arithmetik-Beispiele nach (modifiziertem) IEEE 754

Multiplikation $A \cdot B$ (2 von 4)

	VZ	Exponent					Mantisse									
A	0	1	0	0	0	1	1	0	1	1	0	0	1	0	1	0
B	0	1	0	1	0	0	1	1	1	1	1	0	0	1	0	0

- (2) Summe der Exponenten: $(Exp(A) + Exp(B))_e = (Exp(A))_e + (Exp(B))_e - e$

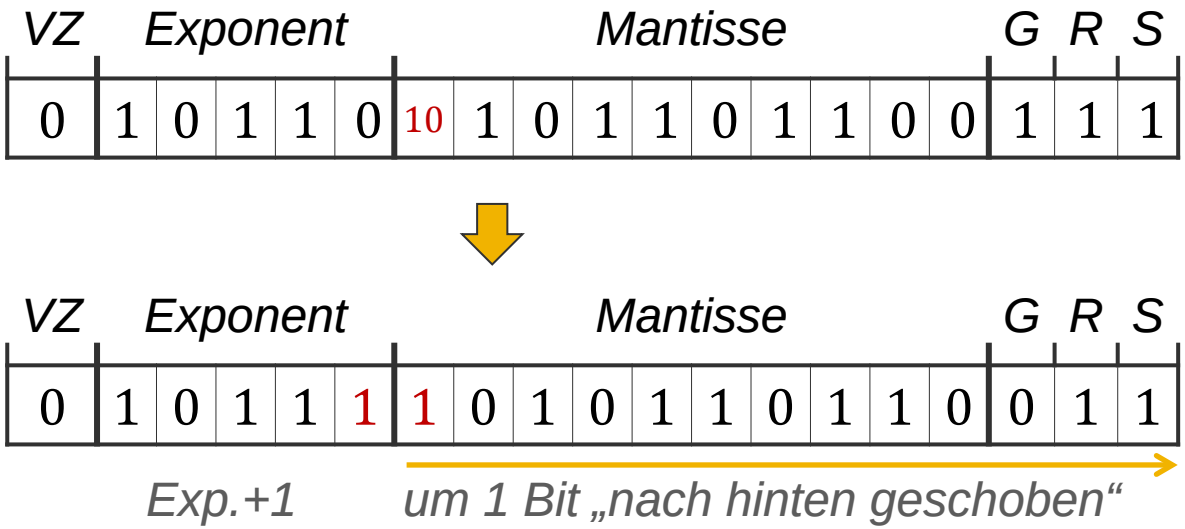
$$
\begin{array}{rrrrrrl}
 & 1 & 0 & 0 & 0 & 1 = (17)_{10} & Exp(A)_e \\
- & 0 & 1 & 1 & 1 & 1 = (15)_{10} & e \\
\hline
 & 0 & 0 & 0 & 1 & 0 = (2)_{10} & Exp(A)
\end{array}
$$

$$
\begin{array}{rrrrrrl}
 & 1 & 0 & 1 & 0 & 0 = (20)_{10} & Exp(B)_e \\
+ & 0 & 0 & 0 & 1 & 0 = (2)_{10} & Exp(A) \\
\hline
 & 1 & 0 & 1 & 1 & 0 = (22)_{10} & (Exp(A) + Exp(B))_e
\end{array}
$$

Arithmetik-Beispiele nach (modifiziertem) IEEE 754

Multiplikation $A \cdot B$ (3 von 4)

- (3) Normalisieren

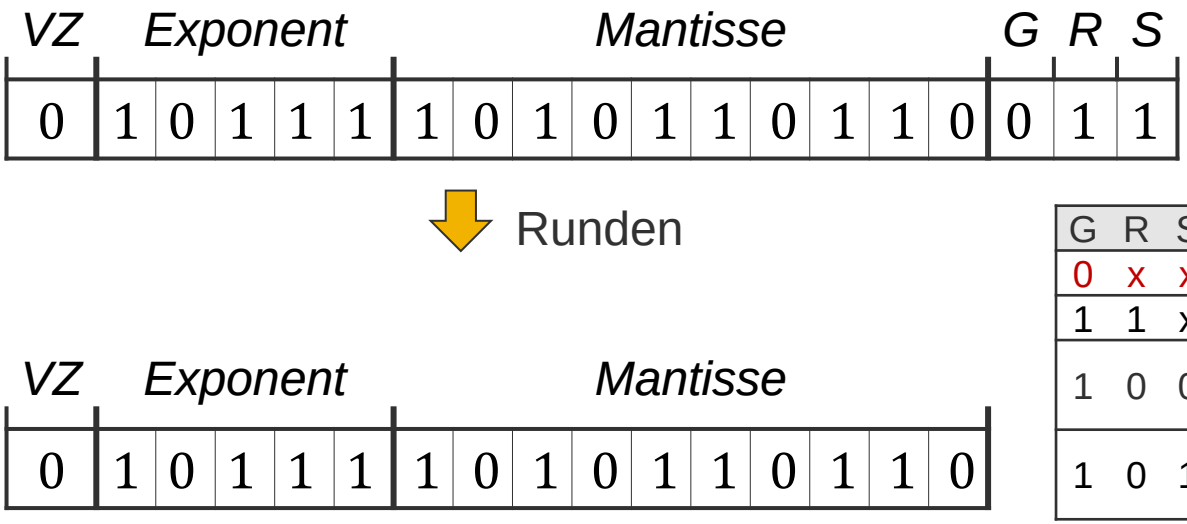


erste 13 Stellen der Multiplikation

Arithmetik-Beispiele nach (modifiziertem) IEEE 754

Multiplikation $A \cdot B$ (4 von 4)

- (4) Runden



G	R	S	Ergebnis (Erg.)
0	x	x	unverändert
1	1	x	Ergebnis += 1
1	0	0	lsb = 0 ⇒ unverändert lsb = 1 ⇒ Erg. += 1
1	0	1	VZ gleich ⇒ Erg. += 1 VZ untersch. ⇒ unverändert

⇒ Ergebnis:

- $0\ 10111\ 1010110110$ bzw. $(1.010110110)_2 \times 2^8$

Genauigkeitsbetrachtungen

Fehlerfortpflanzung

- Subtraktion zweier betragsmäßig annähernd gleich großer Zahlen:
  - Auslöschung
  - Die vorderen übereinstimmenden Mantissenstellen der beiden Operanden heben einander auf
  - Damit werden Ungenauigkeiten an hinteren (weniger wichtigen) Stellen relevanter

- Rechnen mit exakten Werten ($x \in \mathbb{R}$):
  - gutartige Auslöschung
  - Die nach der Auslöschung im Ergebnis verbleibenden hinteren Stellen sind unverfälscht
- Rechnen mit gerundeten Operanden($x \in \mathbb{F}$):
  - katastrophale Auslöschung

Genauigkeitsbetrachtungen

Fehlerfortpflanzung – Bsp.

- Bsp.: $x^2 - y^2$ und $(x - y) \cdot (x + y)$

  $x = 10.1$, $y = 9.99$, optimales Runden auf 3 Stellen genau

- exakt: $x^2 - y^2 = 102.01 - 99.8001 = 2.2099$

- nahezu gutartige Auslöschung

$$(x \boxminus y) \boxtimes (x \boxplus y) = \big(\square(0.11)\big) \boxtimes \big(\square(20.09)\big) =$$

$$= 0.11 \boxtimes 20.1 = \square(2.211) = 2.21$$

  - relativer Rundungsfehler $|\rho(2.21)| = \frac{2.21 - 2.2099}{2.2099} \approx 4 \times 10^{-5}$

- katastrophale Auslöschung

$$(x^{\boxed{2}} \boxminus y^{\boxed{2}}) = (\square(102.01)) \boxminus (\square(99.8001)) = 102 \boxminus 99.8 = 2.2$$

  - relativer Rundungsfehler $|\rho(2.2)| = \frac{2.2 - 2.2099}{2.2099} \approx 4 \times 10^{-3} \gg 4 \times 10^{-5}$

Genauigkeitsbetrachtungen

Fehlerfortpflanzung

- Aufgrund von Rundungsfehlern Abweichung zwischen
  - im Computer implementierten arithmetischen Operationen von
  - zu Grunde liegenden mathematisch exakten Operationen

⇒ Jedes Zwischenergebnis einer numerischen Berechnung kann vom exakten Ergebnis abweichen

- Zwischenergebnisse sind Operanden für nachfolgende Rechenschritte
  ⇒ nachfolgende Rechenschritte mit verfälschten Argumenten

⇒ Fehlerfortpflanzung