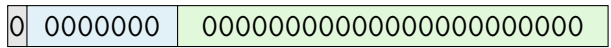
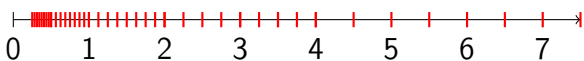
# Aufbau




$$SignBit \times Mantisse \times 2^{Exponent-Bias} \tag{1}$$

- Feste Anzahl an signifikanten Stellen
- Größerer Wertebereich als Fixkomma Zahlen
- Konsequenz: Höhere Genauigkeit bei kleinen Zahlen

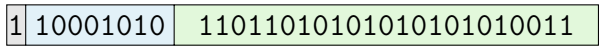


# Aufbau

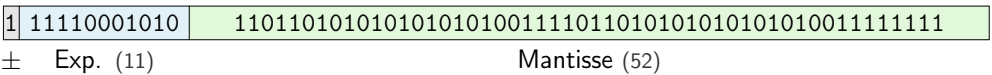
- Normalisiert: $1 \leq \textit{Mantisse} < 2$
- Führende Eins nicht abgespeichert
- Exponent(gespeichert) = Exponent(real) + Bias
  - Bias statt Zweierkomplement
  - Lexikografischer Vergleich statt Subtraktion und Vergleich mit 0
  - In der Theorie weniger Operationen

# Datentypen `float`/`double`

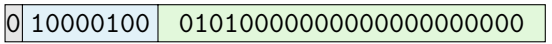
| | Größe | Dezimalziffern | Abs. Min. | Abs. Max. |
|---|---|---|---|---|
| `float` (Single Prec.) | 32 Bit | $\approx 7$ | $\approx 1.18 \cdot 10^{-38}$ | $\approx 3.4 \cdot 10^{38}$ |




| | | | | |
|---|---|---|---|---|
| `double` (Double Prec.) | 64 Bit | $\approx 15$ | $\approx 10^{-308}$ | $\approx 10^{308}$ |



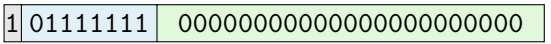
# Quiz: Zahlen zuordnen 1

Welchen Wert hat 

?

- ☐ $1.3125_{10} \times 2^{132}$
- ☐ $42.0_{10}$
- ☐ $0x42280000$

# Quiz: Zahlen zuordnen 2

Welchen Wert hat 

?

- ☐ $-1.0_{10}$
- ☐ $-127.0_{10}$
- ☐ 0

# Addition und Subtraktion

- Kleineren Wert auf selben Exponenten bringen wie großen Wert (denormalisieren)
- Mantissen addieren bzw. subtrahieren
- Mantisse entsprechend der Genauigkeit runden
- Ergebnis normalisieren

# Subtraktion – Beispiel

| + | $2^1$ | 1.0000 | – | + | $2^0$ | 1.5000 |
|---|---|---|---|---|---|---|

Gleicher Exponent

| + | $2^1$ | 1.0000 | – | + | $2^1$ | 0.7500 |
|---|---|---|---|---|---|---|

Mantrisse subtrahieren

$=$

| + | $2^1$ | 0.2500 |
|---|---|---|

Normalisieren

$=$

| + | $2^{-1}$ | 1.0000 |
|---|---|---|

# Multiplikation und Division

- Exponenten addieren bzw. subtrahieren
- Mantissen multiplizieren bzw. dividieren (Führende 1 beachten)
- Mantisse entsprechend der Genauigkeit runden
- Ergebnis normalisieren

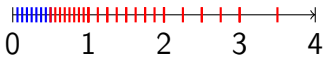
# Probleme bei Genauigkeit

- Rundung: Ergebnis muss wieder FP-Darstellung gespeichert werden
  - Verschiedene Rundungsmodi, Standard: *round to nearest, ties to even*
- Absorption: Addition/Sub. von sehr großer und sehr kleiner Zahl
  - Keine Veränderung der großen Zahl wg. Rundung
  - Beispiel: $1000000.00f + 0.01f = 1000000.00f$
- Auslöschung: Subtraktion großer ähnlicher Zahlen
  - Subtraktion verstärkt Rundungsfehler
  - Beispiel: $1000000.1f - 1000000.0f = 0.125f \neq .1f$
  - Grund: $1000000.1f$ tatsächlich dargestellt als 1000000.125

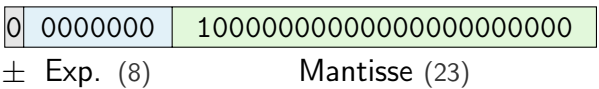
# Assoziativität und Distributivität

- Sowohl Addition und Multiplikation
- Nicht assoziativ
  - $(x + y) + z \neq x + (y + z)$
  - $(x \times y) \times z \neq x \times (y \times z)$
- Nicht distributiv
  - $x(y + z) \neq (xy) + (xz)$
- Achtung: -ffast-math (-Ofast) in GCC ignoriert diese zwecks Geschwindigkeit
- Weiterführend: What Every Computer Scientist Should Know About Floating-Point Arithmetic
  `https://www.itu.dk/~sestoft/bachelor/IEEE754_article.pdf`

# Denormale Zahlen / Subnormale Zahlen

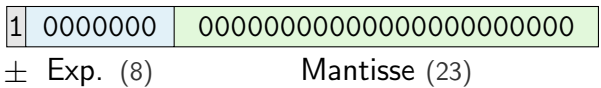



- Zahlen deren Exponent kleiner ist, als eine normalisierte Darstellung zulassen würde
- Beispiel, single precision, normalisiert: $1.0_2 \times 2^{-127}$
- Denormalisiert: $0.1_2 \times 2^{-126}$
- Exponent hat speziellen Wert: alle Bits 0



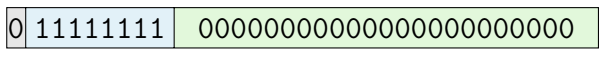
# Null mit Vorzeichen

- Null: Exponent und Mantisse alle Bits 0
- Sign-Bit kann gesetzt sein $\rightarrow +/-0$ möglich
- Üblicherweise: $x + 0 = x$
- Sonderfall: $x = -0 \rightarrow -0 + 0 = +0$
- $-0 \neq +0$



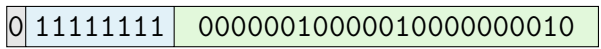
# Unendlich / Infinity / $\infty$

- Alle Bits in Exponent $= 1$
- Alle Bits in Mantisse $= 0$
- $\rightarrow$ je nach Sign-Bit: $+/-$ Unendlich
- z.B. Ergebnis bei $x/0$



# Not a Number / NaN

- Alle Bits in Exponent $= 1$
- Mantisse $\neq 0$
- $\rightarrow$ Not a Number
- z.B. Ergebnis bei $0/0$ und Unendlich - Unendlich
- $x \circ \text{NaN} = \text{NaN}$ 2
- für jeden NaN Wert: $\text{NaN}_1 == \text{NaN}_2 \rightarrow$ false



# Quiz: Sonderfälle 1

Welches Ergebnis hat NaN == NaN?

- ☐ true
- ☐ false
- ☐ Segmentation Fault

# Quiz: Sonderfälle 2

Welches Ergebnis hat NaN != Infinity?

- ☐ true
- ☐ false
- ☐ 1

# Quiz: Sonderfälle 3

Welches Ergebnis hat -Infinity < Infinity?

- ☐ true
- ☐ false
- ☐ Arithmetic Exception: Invalid Operation

# Quiz: Sonderfälle 4

Welches Ergebnis hat 10 != NaN?

- ☐ true
- ☐ false
- ☐ Infinity

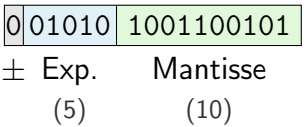
# Quiz: Sonderfälle 5

Welches Ergebnis hat 5.0 / 0.0?

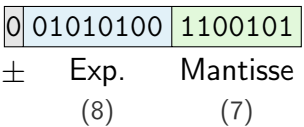
- [ ] -Infinity
- [ ] NaN
- [ ] Infinity
- [ ] Arithmetic exception: Division by Zero

# Weitere Floating Point Formate

- 16 Bit half precision / half




- Brain Floating Point / bfloat




- Extended Formate