Lernzettel

Zahlensysteme, Repräsentationen digitaler Werte und grundlegende Arithmetik

Universität: Technische Universität Berlin
Kurs/Modul: Technische Grundlagen der Informatik (TechGI) - Digitale Systeme
Erstellungsdatum: September 6, 2025

Lernzettel: Zahlensysteme, Repräsentationen digitaler Werte und grundlegende Arithmetik

(1) Zahlensysteme und Repräsentationen.
Zahlensysteme beschreiben, wie man Werte als Ziffernketten in einer Basis darstellt. Die gängigsten Basen sind Binär (2), Dezimal (10), Oktal (8) und Hexadezimal (16). Allgemein gilt:

$$N = \sum_{k=0}^{n-1} d_k\, b^k, \qquad d_k \in \{0, \ldots, b-1\}.$$

Zahlensysteme (Beispiele):

- Binär (b=2): $\{0, 1\}$ als Ziffern. Beispiel: 173 dezimal = 10101101 $_2$ (8 Bit unsigned).
- Dezimal (b=10): $\{0, \ldots, 9\}$.
- Hexadezimal (b=16): $\{0, \ldots, 9, A, B, C, D, E, F\}$. Oft genutzt für Wortdarstellungen.

Vorzeichenbehaftete Darstellungen. Häufig verwendet man mehrere Darstellungen für negative Werte.

- Zweierkomplement (Two's complement): Für n Bits liegt der Wertebereich bei -2^{n-1} bis $2^{n-1} - 1$.
- Von-Neumann-Darstellung (Signed magnitude) oder One's complement sind weniger üblich in Schaltungen.

Zweierkomplement einer Zahl m : $\quad \mathrm{TC}(m) = \overline{m} + 1 \quad$ (im n-Bit-Wort, $\overline{m}$ = Bit-Inversion).

Fixed-Point-Darstellungen. Zahlen mit fester Nachkommastelle werden oft als Qm.n oder Qm.f bezeichnet. Allgemein gilt:

$$V = \sum_{k=0}^{m+n-1} b_k\, 2^k, \quad \text{Wert } v = \frac{V}{2^f}.$$

Dabei ist f die Anzahl der Nachkommastellen. Beispiele:

- In einem 8-Bit-Wort mit 4 Nachkommastellen (Q4.4): 3.25 $\Rightarrow V = 52 = 00110100_2$ und $v = 52/2^4 = 3.25$.

Gleitkommadarstellung (IEEE 754, fokussiert). Eine Gleitkommazahl hat Vorzeichen s, Exponenten e und Mantisse f:

$$(-1)^s\,(1.f)\,2^{\,e-b},$$

mit Bias b (z.B. 127 bei 8-Bit-Exponenten im Single-Precision-Format). Beispiel (Einzelpräzision, Signaleinheit 1/8/23):

$$5.75_{10} \approx 0\,|\,10000001\,|\,01110000000000000000000_2.$$

Endianness.

- Big-endian: höchstwertiges Byte zuerst.
- Little-endian: niedrigstwertiges Byte zuerst.

(2) Repräsentation digitaler Werte in der Praxis.

- Wortbreite w bestimmt maximale Werte, Rechenbereich und Overflow-Verhalten.
- Sign-Flag, Overflow-Flag, Carry-Flag in Schaltungen.
- Logische ($\gg$) vs. arithmetische Verschiebung ($\gg$ mit Vorzeichenerhaltung).

(3) Grundlegende Arithmetik.

- **Addition.** Binäre Addition mit Carry:

$$s = x \oplus y \oplus c, \qquad c_{\text{out}} = (x \wedge y) \vee (x \wedge c) \vee (y \wedge c).$$

- **Subtraktion.** Subtraktion via Addition:

$$a - b = a + (\neg b + 1) \quad \text{(in Zweierkomplement-Darstellung).}$$

- **Überlauf.** Bei Zweierkomplement-Addition tritt Überlauf auf, wenn der Vorzeichenbit des Ergebnisses vom Vorzeichen der Operanden abweicht. Formal:

$$\text{ovf} = c_{n-1} \oplus c_n \quad \text{(Carry in/Carry out des MSB).}$$

- **Multiplikation.** Binäre Multiplikation per Shift-and-Add:

$$P = \sum_{i=0}^{m-1} x_i \, (M \ll i),$$

  wobei x_i Bits des Multipliers und M der Multiplikand ist.

- **Division.** Grundprinzip: $N = D \cdot Q + R$ mit Quotient Q und Rest R.
- **Zusammenhang zu ALU.** Rechenschritte bilden Bausteine von Addierern, Multiplikatoren, Dividern und Logikschaltungen.

(4) Beispiele und Übungsaufgaben.

Beispiel 1 – Dezimal zu Binär (Unsigned 8 Bit).

$$173_{10} = 128 + 32 + 8 + 4 + 1 \;=\; 10101101_2.$$

Beispiel 2 – Zweierkomplement (8 Bit).
Wandeln Sie -42 in 8-Bit Zweierkomplement um.

$$42_{10} = 00101010_2, \quad \neg 42 = 11010101_2, \quad -42 = 11010110_2 \text{ (TC, 8 Bit)}.$$

Beispiel 3 – Fixed-Point (Q4.4, 8 Bit).
Darstellung von 3.25 in Q4.4:

$$v = 3.25 \quad \Rightarrow \quad V = 3.25 \times 2^4 = 52 \;\Rightarrow 52_{10} = 00110100_2 \;\Rightarrow v = \frac{00110100_2}{2^4} = 3.25.$$

Beispiel 4 – Gleitkommazahl (IEEE 754: Single).
Darstellung von 5.75 (Binärform):

$$s = 0, \quad e = 129 = 10000001_2, \quad f = 01110000000000000000000_2.$$

Gesamtbitfolge: $0\,10000001\,01110000000000000000000_2$.

Beispiel 5 – Endianness.
16-Bit-Zahl $0x1234$:

- Big-endian: Bytefolge "12""34".
- Little-endian: Bytefolge "34""12".

Übungsaufgaben (optional).

- Konvertieren Sie 255_{10} in 8-Bit-Binärdarstellung (unsigned).
- Geben Sie das 8-Bit-Zweierkomplement von -128 an.
- Wandeln Sie 6.75 in Q3.2 (8 Bit) um.
- Ermitteln Sie die IEEE-754-Darstellung von -2.5 (Single Precision).