# Aufgaben zu Zahlendarstellungen

## Aufgabe 1 Dezimalzahlen und Binärzahlen

KAP **04**

Berechnen Sie für folgende Dezimalzahlen die entsprechende Binärdarstellung!
Bei Zahlen mit Nachkommastellen ist eine maximale Ungenauigkeit von 5% erlaubt.

a) $25_{10} =$

b) $1,66_{10} \approx$

c) $2.8125 \approx$

## Aufgabe 2 Vorzeichenbehaftete Zahlen (1)

KAP **04**

Führen Sie die folgenden Rechnungen jeweils im Einer- und Zweierkomplement durch. Die Wortlänge betrage dabei 8 Bit. Geben Sie auch den dezimalen Wert des Resultats an.

a) $13 + 21$

b) $125 + 3$

c) $31 - 19$

d) $-87 - 55$

e) $-11 - 21$

## Aufgabe 3 Vorzeichenbehaftete Zahlen (2)

KAP **04**

Führen Sie die folgenden Rechnungen jeweils im Einer- und Zweierkomplement durch. Die Wortlänge betrage dabei 8 Bit. Geben Sie auch den dezimalen Wert des Resultats an.

a) $12 + 27$

b) $127 + 1$

c) $33 - 17$

d) $-88 - 77$

e) $-12 - 14$

## Aufgabe 4 Vorzeichenbehaftete Festkomma-Zahlen

KAP **04**

Führen Sie die Addition 1,5+3,375 für 8-Bit-Festkommazahlen mit drei Nachkommastellen im Zweierkomplement durch! Geben Sie auch den dezimalen Wert des Resultats an!

## Aufgabe 5 Bewertung der beiden Zahlendarstellungen — KAP 04

In modernen Schaltungen wird praktisch nur noch das Zweierkomplement verwendet. Worin liegt der Vorteil dieser Darstellung?

## Aufgabe 6 Absoluter Wert — KAP 04

a) Konstruieren Sie eine Schaltung, die den absoluten Wert einer vorzeichenbehafteten 4-Bit-Zahl in Zweierkomplement-Darstellung berechnet!
Sie dürfen davon ausgehen, dass Volladdierer (oder Halbaddierer) als Bauteile zur Verfügung stehen.

b) Zeigen Sie die Korrektheit Ihrer Schaltung mithilfe der Definitionen der Zweierkomplement-Darstellung $[\![\cdot]\!]$ und der Binär-Darstellung $\langle\cdot\rangle$!
Gehen Sie davon aus, dass der Addierer korrekt implementiert ist. Des Weiteren dürfen Sie das Lemma $2^n = 1 + \sum_{i=0}^{n-1} 2^i$ und die Definition $\neg x = (1-x)$ verwenden.

## Aufgabe 7 Zweierkomplement-Darstellung und Overflow — 04

10

Ein Ripple-Carry-Addierer, der zwei $n$-Bit-Binärzahlen $a = \langle a_{n-1}\, a_{n-2} \dots a_0\rangle$ und $b = \langle b_{n-1}\, b_{n-2} \dots b_0\rangle$ summiert, liefert als Ergebnis eine $n+1$-Bit-Zahl $e = \langle c\, e_{n-1}\, e_{n-2}, \dots e_0\rangle$, wobei $e = a+b$. Für den Fall, dass $c$ gleich Eins ist, spricht man von einem *Overflow*.

a) Verwendet man einen derartigen Ripple-Carry-Addierer für die Addition zweier vorzeichenbehafteten $n$-Bit-Binärzahlen $a$ und $b$ in Zweierkomplement-Darstellung, so bedeutet ein Ergebnis mit $c = 1$ nicht notwendigerweise, dass bei der Addition ein Overflow aufgetreten ist, also dass $[\![s_a\, a_{n-2} \dots a_0]\!] + [\![s_b\, b_{n-2} \dots b_0]\!] \notin \{-2^{n-1}, \dots, 2^{n-1}-1\}$.

   Geben Sie für jeden der folgenden zwei Fälle $n$-Bit-Zahlen $a$ und $b$ an, für die der Ripple-Carry-Adder ein Ergebnis mit $c = 1$ liefert, obwohl bei der Addition kein Overflow auftritt!

   1. $[\![s_a\, a_{n-2} \dots a_0]\!] > 0, [\![s_b\, b_{n-2} \dots b_0]\!] < 0$
   2. $[\![s_a\, a_{n-2} \dots a_0]\!] < 0, [\![s_b\, b_{n-2} \dots b_0]\!] < 0$

   Warum tritt dieses Phänomen im Fall $[\![s_a\, a_{n-2} \dots a_0]\!] > 0, [\![s_b\, b_{n-2} \dots b_0]\!] > 0$ nie auf?

b) Es seien $c_n, \dots, c_0$ die Carry-Bits des $n$-Bit-Ripple-Carry-Adders.

   Zeigen Sie, dass
$$[\![s_a\, a_{n-2} \dots a_0]\!] + [\![s_b\, b_{n-2} \dots b_0]\!] \notin \{-2^{n-1}, \dots, 2^{n-1}-1\} \iff c_{n-1} \oplus c_n$$
   für den Fall $[\![s_a\, a_{n-2} \dots a_0]\!] > 0$ und $[\![s_b\, b_{n-2} \dots b_0]\!] > 0$ !