WS 2004/05	Übungen Technische Informatik 1 (T1)	Musterlösung zu Aufgabenblatt 2

Lösung 2.1

a)

- $K_{10}(321) = 10.000 - 321 = 9679$

$$\begin{array}{r} 5250 \\ +9679 \\ \hline \ddot{u}\ 0100 \\ \hline \boxed{1}\,4929 \end{array}$$

→ Überlauf ignorieren!
(Positives Ergebnis)

$5250 - 321 = 4929$

- $K_9(321) = 10.000 - 1 - 321 = 9678$

$$\begin{array}{r} 5250 \\ +9678 \\ \hline \boxed{1}\,4928 \\ +\quad 1 \\ \hline \boxed{1}\,4929 \end{array}$$

b)

- $K_{10}(864) = 1000 - 864 = 136$

$$\begin{array}{r} 735 \\ +136 \\ \hline \boxed{0}\,871 \end{array}$$

→ Überlauf = 0 → negatives Ergebnis

Differenz ergibt sich als negatives 10-Komplement der Zwischensumme:

$$\begin{aligned} d &= -K_{10}(871) \\ d &= -(1000 - 871) \\ d &= -129 \end{aligned}$$

- $K_9(864) = 1000 - 1 - 864 = 135$

$$\begin{array}{r} 735 \\ +135 \\ \hline \boxed{0}\,870 \end{array}$$

→ Überlauf = 0 → negatives Ergebnis

Differenz ergibt sich als negatives 9-Komplement der Zwischensumme:

$$\begin{aligned} d &= -K_9(870) \\ d &= -(1000 - 1 - 870) \\ d &= -129 \end{aligned}$$

Lösung 2.2

a)

$$(5250)_{10} \Leftrightarrow (1.0100.1000.0010)_2 \quad (n = 13)$$
$$(321)_{10} \Leftrightarrow (1.0100.0001)_2$$

- $K_2(1.0100.0001) \Rightarrow$

$$\begin{array}{rr} & 0.0001.0100.0001 \\ & 1.1110.1011.1110 \\ + & \qquad\qquad 1 \\ \hline & 1.1110.1011.1111 \end{array}$$

$$\begin{array}{rl} & 1.0100.1000.0010 \\ + & 1.1110.1011.1111 \\ \hline \ddot{u} & 11.1000.0011.1100 \\ \hline & \boxed{1}1.0011.0100.0001 \end{array} \quad \rightarrow \text{ positives Ergebnis, da Überlauf=1 (ignorieren!)}$$

$(1.0011.0100.0001)_2 \Leftrightarrow (4929)_{10}$

- $K_1(1.0100.0001) = 1.1110.1011.1110$

$$\begin{array}{rll} & 1.0100.1000.0010 & \\ + & 1.1110.1011.1110 & \\ \hline \ddot{u} & 1.1000.0011.1100 & \\ \hline & \boxed{1}1.0011.0100.0000 & \rightarrow \text{ positives Ergebnis, da Überlauf=1} \\ + & \qquad\qquad 1 & \rightarrow \text{ Korrekturaddition} \\ \hline & 1.0011.0100.0001 & \end{array}$$

$(1.0011.0100.0001)_2 \Leftrightarrow (4929)_{10}$

b)

$$(735)_{10} \Leftrightarrow (10.1101.1111)_2$$
$$(864)_{10} \Leftrightarrow (11.0110.0000)_2$$

- $K_2(11.0110.0000) \Rightarrow$

$$\begin{array}{rr} & 00.1001.1111 \\ + & 1 \\ \hline & 00.0001.1110 \\ \hline & 00.1010.0000 \end{array}$$

$$\begin{array}{r} 10.1101.1111 \\ +00.1010.0000 \\ \hline \end{array}$$

$\boxed{0}11.0111.1111 \rightarrow$ negatives Ergebnis, da Überlauf=0

Differenz ergibt sich als negatives 2-Komplement

$$\begin{aligned} d &= -K_2(11.0111.1111) \\ &= -(00.1000.0001) \end{aligned}$$

- $K_1(11.0110.0000) = 00.1001.1111$

$$\begin{array}{r} 10.1101.1111 \\ +00.1001.1111 \\ \hline 01.0011.1110 \\ \hline \end{array}$$

$\boxed{0}11.0111.1110 \rightarrow$ wiederum negatives Ergebnis, da Überlauf=0

Differenz ergibt sich als negatives 1-Komplement

$$\begin{aligned} d &= -K_1(11.0111.1110) \\ &= -(1000.0001) \end{aligned}$$

Lösung 2.3

$$(135)_{10} \Leftrightarrow (1000.0111)_2$$
$$(21)_{10} \Leftrightarrow (1.0101)_2$$

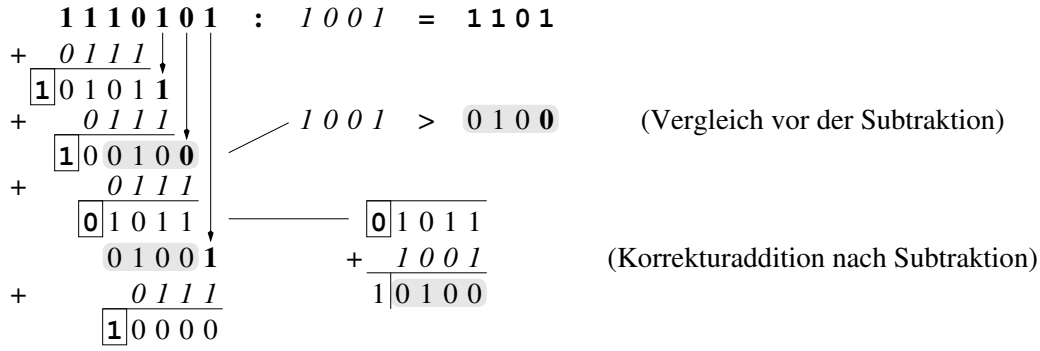
$$\begin{array}{r} 1000.0111 \cdot 10101 \\ \hline 1000.0111 \\ 100001.11\mathit{00} \\ 1000.0111.\mathit{0000} \\ \hline \ddot{u}\ 0001.1111.1000 \\ \hline 1011.0001.0011 \end{array}$$

$(1011.0001.0011)_2 \Leftrightarrow (2835)_{10}$

Lösung 2.4

$$(117)_{10} \Leftrightarrow (111.0101)_2$$
$$(9)_{10} \Leftrightarrow (1001)_2$$

$$K_2(9) = (0111)_2$$



$$(1101)_2 \Leftrightarrow (13)_{10}$$

Führt eine Subtraktion zu einem negativen Zwischenergebnis, so muß die betreffende Subtraktion rückgängig gemacht werden. In der dargestellten Musterlösung der "gespeicherte Wert" $\boxed{0100}$ einfach wieder hergestellt. Alternativen dazu sind der Vergleich ($1001 > 0100$) *vor* der Durchführung der Subtraktion oder die Wiederherstellung durch eine Korrekturaddition ($+1001$) *nach* einer Subtraktion mit negativem Ergebnis.

Lösung 2.5

a) 8-Bit-Darstellung:

$$
\begin{array}{rcl}
(114)_{10} & \Leftrightarrow & (0111.0010)_2 \\
(67)_{10} & \Leftrightarrow & (0100.0011)_2 \\
K_2(67) & = & \quad 1011.1101
\end{array}
$$

$$
\begin{array}{rr}
(114)_{10} & \quad 0111.0010 \\
(-67)_{10} & +1011.1101 \\
\hline
 & \quad 1.1110.0000 \quad \text{Übertrag (Carry)} \\
\hline
 & \quad 0010.1111
\end{array}
$$

$$(0010.1111)_2 \Leftrightarrow (47)_{10}$$

Interpretation:
- Überlauf $c_8 = 1 \quad \rightarrow$ positives Ergebnis
- $c_8 = c_7 = 1 \quad \rightarrow$ keine Zahlenbereichsüberschreitung

b) 8-Bit-Darstellung:

$$
\begin{array}{rcll}
(60)_{10} & \Leftrightarrow & (0011.1100)_2 & \\
K_2(60) & = & \quad 1100.0100 & \Leftrightarrow (-60)_{10} \\
(73)_{10} & \Leftrightarrow & (0100.1001)_2 & \\
K_2(73) & = & \quad 1011.0111 & \Leftrightarrow (-73)_{10}
\end{array}
$$

$$
\begin{array}{rr}
(-60)_{10} & \quad 1100.0100 \\
(-73)_{10} & + \ 1011.0111 \\
\hline
 & \ddot{u}\,10000.1000 \\
\hline
 & \quad 0111.1011
\end{array}
$$

$$(0111.1011)_2 \Leftrightarrow (123)_{10}$$

Interpretation:
- $c_8 \neq c_7 \quad \rightarrow$ Zahlenbereichsüberschreitung (Underflow)
  $\Rightarrow$ Ergebnis falsch!

Auswertung in einem Prozessor beispielsweise durch Auslösen eines internen Interrupts (Exception, Over- / Underflow).

Weiterführende Aufgabe: Fälle überlegen, die zu gültigen oder ungültigen Ergebnissen führen:
- Erkennung der Gültigkeit des Ergebnis anhand der Überträge
- Vermeidung von Overflow / Underflow durch Einschränkung des Definitionsbereichs

Lösung 2.6

a) $0,27374 \cdot 10^5$ **b)** $-0,1101111 \cdot 2^{-1}$ **c)** $-0,3A2 \cdot 16^A$

Lösung 2.7

a)

Darzustellende Zahl: $(10.0000)_2$

Positive Zahl	$\Longrightarrow$	sign-Bit s $= 0$
Normalisierung	$\Longrightarrow$	$10.0000 = 1,0 \cdot 2^5$
Charakteristik	$\Longrightarrow$	$(127)_{10} + (5)_{10} = (132)_{10} \Leftrightarrow (10000100)_2$
Fraktion	$\Longrightarrow$	000.0000.0000.0000.0000.0000

Resultierendes 32 Bit Wort gemäß IEEE 754:

Sign	Biased Exponent	Fraction
0	10000100	00000000000000000000000

b)

Darzustellende Zahl: $-(0100.1011,110)_2$

Negative Zahl	$\Longrightarrow$	sign-Bit s $= 1$
Normalisierung	$\Longrightarrow$	$0100.1011,110 = 1,00101111 \cdot 2^6$
Charakteristik	$\Longrightarrow$	$(127)_{10} + (6)_{10} = (133)_{10} \Leftrightarrow (10000101)_2$
Fraktion	$\Longrightarrow$	001.0111.1000.0000.0000.0000

Resultierendes 32 Bit Wort gemäß IEEE 754:

Sign	Biased Exponent	Fraction
1	10000101	00101111000000000000000

c)

Darzustellende Zahl: $(0,01100101)_2$

Positive Zahl	$\Longrightarrow$	sign-Bit s $= 0$
Normalisierung	$\Longrightarrow$	$0,01100101 = 1,100101 \cdot 2^{-2}$
Charakteristik	$\Longrightarrow$	$(127)_{10} + (-2)_{10} = (125)_{10} \Leftrightarrow (01111101)_2$
Fraktion	$\Longrightarrow$	100.1010.0000.0000.0000.0000

Resultierendes 32 Bit Wort gemäß IEEE 754:

Sign	Biased Exponent	Fraction
0	01111101	10010100000000000000000

Lösung 2.8

Unter Berücksichtigung der Steuerzeichen für Wagenrücklauf (`0D`) und Zeilenvorschub (`0A`) ergibt sich mit vorangehender Leerzeile der folgende Klartext:

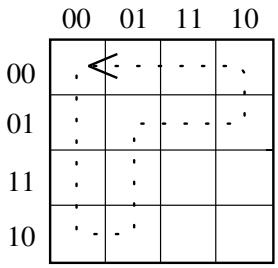
```
Viel
 Spaß
  beim
   Lösen
    der
     Aufgaben!
```

Lösung 2.9

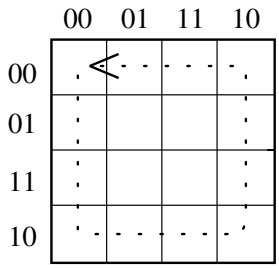
Bei einem einschrittigen Code wird beim Übergang von einem Codewort zum nächsten Codewort genau ein Bit verändert – daraus folgt, daß die Parität invertiert wird. Damit nach n solcher einschrittigen Übergänge wieder das anfängliche Codewort entsteht, muss insbesondere auch dessen Parität wieder erreicht werden. Deshalb muss in jedem Fall eine gerade Anzahl einschrittiger Übergänge stattgefunden haben, d. h. einen zyklisch-einschrittigen Code kann es nur für eine gerade Anzahl von Codewörtern geben.

Lösung 2.10

Im folgenden sind nur zwei von den zahlreichen möglichen Lösungen aufgeführt:



Codewort 0: 0 0 0 0
Codewort 1: 0 1 0 0
Codewort 2: 1 1 0 0
Codewort 3: 1 0 0 0
Codewort 4: 1 0 0 1
Codewort 5: 1 1 0 1
Codewort 6: 0 1 0 1
Codewort 7: 0 1 1 1
Codewort 8: 0 1 1 0
Codewort 9: 0 0 1 0
Codewort 10: 0 0 1 1
Codewort 11: 0 0 0 1
Codewort 0: 0 0 0 0



Codewort 0: 0 0 0 0
Codewort 1: 0 1 0 0
Codewort 2: 1 1 0 0
Codewort 3: 1 0 0 0
Codewort 4: 1 0 0 1
Codewort 5: 1 0 1 1
Codewort 6: 1 0 1 0
Codewort 7: 1 1 1 0
Codewort 8: 0 1 1 0
Codewort 9: 0 0 1 0
Codewort 10: 0 0 1 1
Codewort 11: 0 0 0 1
Codewort 0: 0 0 0 0