TI - 2. Übung

Henning Lehmann
Parya Nemtsava

Afg. 1

1. $\forall i \in \mathbb{N} \setminus \{0\}: 2^i \bmod 2 = 0$, da jede Zweierpotenz trivialerweise ohne Rest durch 2 teilbar ist. Die Summe mehrerer Zweierpotenzen ist ebenfalls durch 2 teilbar: $\left(\sum_{i=1}^{n-1} 2^i\right) \bmod 2 = 0$ und ebenfalls $\left(\sum_{i=1}^{n-1} b_i \cdot 2^i\right) \bmod 2 = 0$. Da $b_0 \in \{0, 1\}$, ist $b_0 \bmod 2 = b_0$. Zusammengefasst:

$$\left(\sum_{i=0}^{n-1} b_i\, 2^i\right) \bmod 2 = \left(\sum_{i=0}^{0} b_i\, 2^i + \sum_{i=1}^{n-1} b_i\, 2^i\right) \bmod 2$$

$$= (\underbrace{b_0}_{\bmod 2 = b_0} + \underbrace{\sum_{i=1}^{n-1} b_i\, 2^i}_{\bmod 2 = 0}) \bmod 2$$

$$= b_0$$

2. Eingabe: Dezimalzahl x

~~Ausgabe~~: Binärstellen $b_0 \dots b_{n-1}$

```
n ← ⌊log_2(x)⌋
for i=0 to n:
    b_i ← x mod 2
    x ← ⌊x/2⌋
return b_0 ... b_n-1
```

3. Zeile 2: $18 : 2 = 9$ Rest 0

Es fehlt am Ende:

$1 : 2 = 0$ Rest 1

Zum Schluss wurden die Binärstellen in der verkehrten Reihenfolge zusammengesetzt. Nach Korrektur dieser drei Fehler: $(37)_{10} = (100101)_2$

Afg. 2

Dezimal	Binär [*]	Oktal [*]	Hexadezimal [*]	9-Bit Zweierkomplement
198	1100 0110	306	C6	011000110
173	10101101	255	AD	010101101
349	1 0101 1101	535	15D	–
1196	100 1010 1100	2254	4AC	–
-171	–	–	–	101010101

[*] Vorzeichenlose Zahlen

Afg. 3

1. i. 1000.0000 iii. 0000.10000 v. 0000.00011

   ii. 1001.1000 iv. 0000.00100 vi. keine Darstellung, da zu groß

2. i. $-(2^{-6}) = 0{,}015625$ iii. 9,6640625 v. 7,96875

   ii. 0 iv. -9,6796875 vi. -8

Afg. 4

$a = 11111{,}101_2 = 1{,}\underline{1111101}_2 \cdot (2^4)_{10}$

Exponent:

$$\begin{array}{r} 100 \\ +\ 1111111 \\ \hline =\ \underline{10000011} \end{array}$$

↓

IEEE 754: 0 10000011 11111010000000000000000 = a

$b = 32{,}7_{10} = 100000{,}1\overline{0110} = 1{,}\underline{000001\overline{0110}}_2 \cdot (2^5)_{10}$

Exponent:

$$\begin{array}{r} 101 \\ +\ 1111111 \\ \hline \underline{10000100} \end{array}$$

Aufgerundet

IEEE 754: 0 10000100 00000101100110011001101

$c = 111111,1010000$

$+ 1000000,1011001$

$= 10000000,0101001$

Afg. 5

1.

01001000	H
01100101	e
01101100	l
01101100	l
01101111	o
10100000	␣
01110111	w
01101111	o
01110010	r
01101100	l
11100100	d
00100001	!

2. Info ist toll!

Afg. 6

1. "e" : 01100101

   "€" : 11100010 10000010 10101100

2. Die Binärsequenz kodiert den Buchstaben "a". Dieser Buchstabe könnte aber auch als einzelnes Byte (01100001) kodiert werden, da nur die letzten 7 Bits Einsen enthalten.

3. Nur dann, wenn die ersten 128 Zeichen des Latin-1-Zeichensatzes verwendet werden, damit das erste Bit jedes Bytes 0 ist (wie bei der UTF-8-Kodierung von 7-Bit-Zeichen).

4. Für die BMP müssen im schlimmsten Fall 16 Bits kodiert werden. Diese werden in UTF-16 mit 2 Bytes, und in UTF-8 mit 3 Bytes kodiert. Daher ist eine UTF-8-Datei im worst case 1,5-mal so groß wie eine UTF-16-Datei.

Afg. 7

1. Code-Distanz $= n$, da sich z.B. zwischen dem a und dem b nur die letzten n Bits verändern.

2. Es können bis zu $n-1$ Bitfehler erkannt, und bis zu $\lfloor (n-1)/2 \rfloor$ Bitfehler korrigiert werden.