Institut für Informatik
Heinrich-Heine-Universität Düsseldorf
Prof. Dr. Alice McHardy
Robin Küppers
Sebastian Konietzny
Aaron Weimann

WS 2013/2014
29.10.2013
Übungsblatt 1

Einführung in die Informatik I
Übungen zur Vorlesung
Musterlösung

1.1 Umrechnung zwischen Zahlensystemen (6 Punkte)

Vervollständigen Sie die folgende Tabelle, indem Sie in jeder Zeile zwischen den Zahlensystemen umrechnen.

Binärsystem	*Hexadezimalsystem*	*Dezimalsystem*
		$(76)_{10}$
	$(feba)_{16}$	
$(0001\ 0010\ 0110\ 0111)_2$		

Musterlösung

Binärsystem	*Hexadezimalsystem*	*Dezimalsystem*
$(0000\ 0000\ 0100\ 1100)_2$	$(4c)_{16}$	$(76)_{10}$
$(1111\ 1110\ 1011\ 1010)_2$	$(feba)_{16}$	$(65210)_{10}$
$(0001\ 0010\ 0110\ 0111)_2$	$(1267)_{16}$	$(4711)_{10}$

Hinweis zur Benotung: Je ein Punkt für jede berechnete Zahl in der Tabelle.

Mögliche Rechenwege:

- Binärsystem → Dezimalsystem: Zweierpotenzen aufsummieren
- Dezimalsystem → Binärsystem: Division durch 2 mit Rest

- Binärsystem → Hexadezimalsystem: vier Binärziffern entsprechen einer Hexadezimalziffer (Nibble)
- Hexadezimalsystem → Binärsystem: analog
- Dezimalsystem → Hexadezimalsystem: Division durch 16 mit Rest
- Hexadezimalsystem → Dezimalsystem: Sechzehnerpotenzen aufsummieren

Beispielrechnungen:

Beispielrechnung (Dezimal → Binärsystem) für $(76)_{10}$:

$76/2 = 38 + 0$	0
$38/2 = 19 + 0$	00
$19/2 = 9 + 1$	100
$9/2 = 4 + 1$	1100
$4/2 = 2 + 0$	01100
$2/2 = 1 + 0$	001100
$1/2 = 0 + 1$	$(1001100)_2$

Beispielrechnung (Hex → Dezimalsystem) für $(feba)_{16}$:

$$\begin{aligned}(feba)_{16} &= (a * 16^0) + (b * 16^1) + (e * 16^2) + (f * 16^3)\\ &= 10 + (11 * 16) + (14 * 256) + (15 * 4096)\\ &= (65210)_{10}\end{aligned}$$

Beispielrechnung (Bin → Dezimalsystem) für $(0001001001100111)_2$:

$$\begin{aligned}(0001\ 0010\ 0110\ 0111)_2 &= 2^0 + 2^1 + 2^2 + 2^5 + 2^6 + 2^9 + 2^{12}\\ &= 1 + 2 + 4 + 32 + 64 + 512 + 4096\\ &= (4711)_{10}\end{aligned}$$

1.2 Unicode und UTF-8 (6 Punkte)

1. Gegeben sei das folgende UTF-8 codierte Zeichen (als Hexadezimalzahl): E2 98 BA
   Bestimmen Sie das zugehörige Unicodezeichen und geben Sie es als Hexadezimalzahl an.

Schreiben Sie die Hexadezimalzahl zunächst binär auf und suchen Sie anschließend die "Nutz-Bits". Geben Sie alle Zwischenschritte Ihrer Berechnung an. (3 Punkte)

2. Gegeben sei das folgende Unicodezeichen (als Hexadezimalzahl): 26 7F
   Codieren Sie das Zeichen mit UTF-8 (als Hexadezimalzahl). Schreiben Sie die Hexadezimalzahl zunächst binär auf. Wandeln Sie dann die Zahl in die UTF-8 Codierung um und markieren Sie die Bits, die hinzugefügt werden müssen. Geben Sie alle Zwischenschritte Ihrer Berechnung an. (3 Punkte)

Musterlösung

1. $(E2\ 98\ BA)_{16} \hat{=} 11100010\ 10011000\ 10111010_2$
   $111000101001100010111010_2$
   $00100110\ 00111010_2 \hat{=} (26\ 3A)_{16}$ (WHITE SMILING FACE)

2. $(26\ 7F)_{16} \hat{=} 00100110\ 01111111_2$
   $111000101001100110111111_2$
   $1110\ 0010\ 1001\ 1001\ 1011\ 1111_2 \hat{=} (E2\ 99\ BF)_{16}$ (WHEELCHAIR SYMBOL)

1.3 Rechnen in der Zweierkomplement-Darstellung (5 Punkte)

1. Nennen Sie zwei Vorteile der Zweierkomplement-Darstellung gegenüber der Vorzeichen-Betrags-Darstellung. (1 Punkt)

2. Wandeln Sie die folgenden Ausdrücke zunächst in 8-Bit-Zweierkomplement-Darstellung um. Berechnen Sie anschließend den Wert der Ausdrücke in Zweierkomplement-Darstellung[1]. (1 + 1 + 2 Punkte)

   (a) $(15)_{10} - (8)_{10}$

   (b) $(3)_{10} - (10)_{10}$

   (c) $(8)_{10} - (39)_{10} - (6)_{10} + (50)_{10}$

Musterlösung

[1] Eine führende Eins (an 9. Stelle) wird verworfen, wenn sie bei der Addition durch einen Übertrag entsteht. (Sehen sie auch: http://de.wikipedia.org/wiki/Zweierkomplement#Rechenoperationen)

1. Nennen Sie zwei Vorteile der Zweierkomplement-Darstellung gegenüber der Vorzeichen-Betrags-Darstellung. (1 Punkt)

   - benötigt keine Fallunterscheidung, ob mit negativen oder mit positiven Zahlen gerechnet wird; d.h. im Rechenwerk muss keine Subtraktion implementiert sein
   - genau eine Darstellung für Null

2. Wandeln Sie die folgenden Ausdrücke zunächst in 8-Bit-Zweierkomplement-Darstellung um. Berechnen Sie anschließend den Wert der Ausdrücke in Zweierkomplement-Darstellung. (1 + 1 + 2 Punkte)

   Rechenoperationen:

   - Addition wie gewohnt (Überlauf ignorieren)
   - Subtraktion als Negation gefolgt von Addition
   - Negation als Komplementbildung und Addition von 1

   (a) $(8)_{10} = (00001000)_2$, invertieren: $(11110111)_2$, Eins addieren: $(11111000)_{2k}$

   $$\begin{aligned} & (15)_{10} - (8)_{10} \\ =& (00001111)_{2k} + (11111000)_{2k} \\ =& (00000111)_{2k} \\ =& (7)_{10} \end{aligned}$$

   (b) $(10)_{10} = (00001010)_2$, invertieren: $(11110101)_2$, Eins addieren: $(11110110)_{2k}$

   $$\begin{aligned} & (3)_{10} - (10)_{10} \\ =& (00000011)_{2k} + (11110110)_{2k} \\ =& (11111001)_{2k} \\ =& (-7)_{10} \end{aligned}$$

   (c) $(39)_{10} = (00100111)_2$, invertieren: $(11011000)_2$, Eins addieren: $(11011001)_{2k}$
   $(6)_{10} = (00000110)_2$, invertieren: $(11111001)_2$, Eins addieren: $(11111010)_{2k}$

   $$\begin{aligned} & (8)_{10} + (-39)_{10} + (-6)_{10} + (50)_{10} \\ =& (00001000)_{2k} + (11011001)_{2k} + (11111010)_{2k} + (00110010)_{2k} \\ =& (11100001)_{2k} + (11111010)_{2k} + (00110010)_{2k} \\ =& (111011011)_{2k} + (00110010)_{2k}, \text{ Überlauf ignorieren: } (11011011)_{2k} + (00110010)_{2k} \\ =& (100001101)_{2k}, \text{ Überlauf ignorieren: } (00001101)_{2k} \\ =& (13)_{10} \end{aligned}$$

1.4 Darstellung von Festkomma- und Gleitkommazahlen (8 Punkte)

1. Beschreiben Sie die Bedeutung des ersten Bits und der beiden darauffolgenden Bitfolgen in der kompakten Gleitkommadarstellung. (2 Punkte)
2. Wieso ist die Normalisierung von Gleitkommazahlen sinnvoll? (2 Punkte)
3. Welche Dezimalzahlen werden durch die IEEE 754 Single-Precision-Gleitkommadarstellungen 0 10001011 00000000000000001000100 und 1 10000001 01000000000000000000000 kodiert? (4 Punkte)

Musterlösung

1. Beschreiben Sie die Bedeutung des ersten Bits und der beiden darauffolgenden Bitfolgen in der kompakten Gleitkommadarstellung. (2 Punkte)
   - erstes Bit: Vorzeichen (0 für positiv, 1 für negativ)
   - Mantisse: enthält die Ziffern der Gleitkommazahl, durch geeignete Wahl des Exponenten wird eine führende Eins erzwungen, die in der kompakten Darstellung eingespart werden kann.
   - Exponent: speichert die Stelle des Kommas bzw. die Größenordnung der Zahl
2. Wieso ist die Normalisierung von Gleitkommazahlen sinnvoll? (2 Punkte)
   - Normalisierte Zahlen sind direkt vergleichbar, denn jede darstellbare Zahl hat genau eine normalisierte Kodierung.
   - Das erste Bit ist implizit bekannt und kann weggelassen werden.
3. Welche Dezimalzahlen werden durch die IEEE 754 Single-Precision-Gleitkommadarstellungen 0 10001011 00000000000000001000100 und 1 10000001 01000000000000000000000 kodiert? (4 Punkte)

   0 10001011 00000000000000001000100

   $exp : 2^7 + 2^3 + 2^1 + 2^0 = 139$
   $exp - bias = 139 - 127 = 12$
   $mantisse : 2^{-17} + 2^{-21}$
   $\Rightarrow (1.0 + 2^{-17} + 2^{-21}) * 2^{12}$
   $= 2^{12} + 2^{-5} + 2^{-9}$
   $= 4096 + 0,03125 + 0,001953125 = 4096,033203125$

1 10000001 01000000000000000000000

$exp: 2^7 + 2^0 = 129$
$exp - bias = 129 - 127 = 2$
$mantisse: 2^{-2}$
$\Rightarrow (1.0 + 2^{-2}) * 2^2 = 5, Vorzeichenbit = 1 => -5$