

# GTI – ÜBUNG 4

BINÄR-, HEX- UND GLEITKOMMAZAHLEN-ARITHMETIK

# Aufgabe 1 – Bin- und Hex Arithmetik

2

- ▶ Führen Sie die folgenden Berechnungen **im angegebenen Zahlensystem aus**, ohne die Zahlen ins Dezimalsystem umzuwandeln:
- ▶ Hinweis: Arbeiten Sie mit dem Zweierkomplement und benutzen Sie 10 Bits für die Binärzahlendarstellung

▶  $111010100110_2 + 010101111110_2$  (Addition im Dualsystem)

▶  $B674FC12_{16} + 2DA9D4B2_{16}$  (Addition im Hexadezimalsystem)

▶  $11101010_2 \cdot 1011_2$  (Multiplikation im Dualsystem)

▶  $11010010_2 - 10110101_2$  (Subtraktion im Dualsystem)

▶  $01110110_2 - 10011001_2$  (Subtraktion im Dualsystem)

# Aufgabe 1 – Bin- und Hex Arithmetik

## ► Addition in fremden Systemen

**Ziel:** wir berechnen  $a + b = c$  im System  $d$

**Ansatz:** wir addieren jeweils stellenweise

A handwritten binary addition problem. The numbers are 1110101 and 0101011. The sum is 1010000. A red box highlights the 7th bit (1) of the first number and the 7th bit (1) of the second number. A yellow circle highlights the 7th bit (1) of the result, which is the carry from the 7th bit. The 8th bit of the result is 0, indicating a carry of 1 to the 8th bit.

$$\begin{array}{r} 1110101 \\ + 0101011 \\ + 1111111 \\ \hline = 1010000 \end{array}$$

Hierbei gilt:

- Ist die Rechnung zu kompliziert, wandle  $a$  und  $b$  in das Dezimalsystem und  $c$  wieder zurück
- Hat das Ergebnis der Addition mehrere Stellen, schreibe nur die letzte Stelle nieder
- Der Rest  $e$  wandert als Übertrag zur Berechnung der nächsten Stelle
- Die Berechnung der nächsten Stelle lautet dann:  $a + b + e = c$

Beispiel:  $1_2 + 1_2 + 1_2 = 11_2$ , d.h.  $1_2$  Übertrag  $1_2$  (siehe oben)

# Aufgabe 1 – Bin- und Hex Arithmetik

- ▶  $111010100110_2 + 010101111110_2$  (Addition im Dualsystem)

**Tipp:** Rechenregeln erfüllen bekannte Dezimaladditionsregeln, d.h. z.B.  
 $1_2+1_2+1_2+1_2+1_2 = 5_{10} = 10_2$  (wobei hier 10 ein Übertrag ist)

Mechanismus der Addition nach Grundschuladdition:

		1	1	1	0	1	0	1	0	0	1	1	0
+		0	1	0	1	0	1	1	1	1	1	1	0
+	1	1	1	1	1	1	1	1	1	1	1		
<hr/>													
=	1	0	1	0	0	0	0	1	0	0	1	0	0

Bei jeder Stelle nach dem Tipp das Ergebnis bestimmen.

# Aufgabe 1 – Bin- und Hex Arithmetik

- ▶  $B674FC12_{16} + 2DA9D4B2_{16}$  (Addition im Hexadezimalsystem)

**Tipp:** wer das Ergebnis einer Hexadezimaladdition  $a + b = c$  nicht direkt sieht, wandelt vorher  $a$  und  $b$  in das Dezimalsystem,  $c$  wieder zurück

Mechanismus der Addition nach Grundschuladdition:

	B	6	7	4	F	C	1	2
+	2	D	A	9	D	4	B	2
+	1	1		1	1			
<hr/>								
=	E	4	1	E	D	0	C	4

Bei jeder Stelle nach dem Tipp das Ergebnis bestimmen.

# Einschub: BCD-Addition

## Pseudotetrade:

Binärdarstellung von nicht dezimalen Ziffern, also 1010 bis 1111 (10 bis 15)

## Addition:

Addition funktioniert prinzipiell wie die Binäraddition, mit folgenden Abweichungen:

- Stellen (je 4 bit) werden getrennt addiert und der Übertrag in die nächste Stelle gezogen
- Pseudotetraden müssen korrigiert werden
- Überlauf muss korrigiert werden, wenn er nicht durch einen Korrekturschritt verursacht wurde
- Korrekturschritt ist die Addition von 6 (0110)

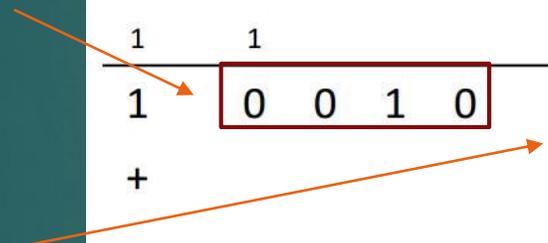
# Einschub: BCD-Addition

Beispiel:

Keine Korrektur:  
Verursacht durch  
Korrekturschritt

Korrektur:  
Da Überlauf

	0	1	1	1	0	1	1	1	0	1	0	1	
+	0	1	0	0	1	0	0	1	1	0	0	0	
	1	1	1	1	1	1	1						
<hr/>													
	1	1	0	0	0	0	0	0	1	1	0	1	Pseudotetrade
+	0	1	1	0					0	1	1	0	Korrektur
	1	1					1		1				
<hr/>													
1	0	0	1	0	0	0	0	1	0	0	1	1	Überlauf
+					0	1	1	0					Korrektur
<hr/>													
1	0	0	1	0	0	1	1	1	0	0	1	1	
<hr/>													
→ $775_{10} + 498_{10} = 1273_{10}$													



# Aufgabe 1 – Bin- und Hex Arithmetik

8

- ▶ Subtraktion im 2er-Komplement

**Ziel:** wir berechnen  $a - b = c$  im Binärsystem und nutzen das 2er-Komplement

**Ansatz:** wir bilden die Subtraktion auf die Addition ab

$a + (-b) = c$ ;  $-b$  bilden wir mit Hilfe des 2er-Komplements

Fallunterscheidung:

ist das erste Bit von  $c$  gesetzt  $\rightarrow c$  positiv, Bit vorne abschneiden

ist das erste Bit nicht gesetzt  $\rightarrow c$  negativ, noch 2er Komplement bilden

# Aufgabe 1 – Bin- und Hex Arithmetik

- ▶ Subtraktion  $11010010_2 - 10110101_2$

Ansatz:  $a + (-b) = c$ ;  $-b$  bilden wir mit Hilfe des 2er-Komplements

Wir rechnen  
mit 10 Bits

	0011010010	Minuend
-	0010110101	Subtrahend
	1101001010	Subtrahend (B-1)
+	0000000001	
	1101001011	Subtrahend (B-2)
+	0011010010	Minuend
=	10000011101	Übertrag = 1 Ergebnis positiv
	0000011101	Ergebnis

Bilden des  
2er -Komplements

Addition  
 $a + (-b)$

# Aufgabe 1 – Bin- und Hex Arithmetik

- ▶ Subtraktion  $01110110_2 - 10011001_2$

**Ansatz:**  $a + (-b) = c$ ;      $-b$  bilden wir mit Hilfe des 2er-Komplements

	0001110110	Minuend
-	0010011001	Subtrahend
<hr/>		
	1101100110	Subtrahend (B-1)
+	0000000001	
<hr/>		
	1101100111	Subtrahend (B-2)
+	0001110110	Minuend
<hr/>		
= 0	1111011101	Übertrag = 0 Ergebnis negativ
	1111011101	Ergebnis
<hr/>		
	0000100010	Betragsbildung
+	0000000001	zur Kontrolle
<hr/>		
	0000100011	Betrag

Bilden des  
2er -Komplements

Addition  
 $a + (-b)$

Bilden des  
2er -Komplements

# Aufgabe 1 – Bin- und Hex Arithmetik

- ▶ Multiplikation in fremden Systemen

**Ziel:** wir berechnen  $a \cdot b = c$  im System  $d$

**Ansatz:** wir bilden die Multiplikation auf die Addition ab

d.h. wir bilden die Summe aus  $a \cdot b_{\text{STELLE1}} + a \cdot b_{\text{STELLE2}} + \dots + a \cdot b_{\text{STELLEn}}$

$$\begin{array}{r} \begin{array}{cccc} a_0 & a_1 & a_2 & \dots & a_n \\ \hline a_0 & a_1 & a_2 & \dots & a_n & * & b_n \\ + & & a_0 & a_1 & a_2 & \dots & a_n & * & b_{n-1} \\ + & & & & & & & & & & \vdots \\ \vdots & & & & & & & & & & \vdots \\ + & & & & a_0 & a_1 & a_2 & \dots & a_n & * & b_1 \\ \hline c_0 & c_1 & \dots & & & & & & & & c_n \end{array} \end{array}$$

# Aufgabe 1 – Bin- und Hex Arithmetik

12

▶  $11101010_2 * 1011_2$

(Multiplikation im Dualsystem)

Mechanismus der Abbildung auf die Addition:

	1	1	1	0	1	0	1	0	x	1	0	1	1
+	1	1	1	0	1	0	1	0					
+		0	0	0	0	0	0	0					
+			1	1	1	0	1	0	1	0			
+				1	1	1	0	1	0	1	0		
+ 1	1		1	1		1	1						
<hr/>													
	1	0	1	0	0	0	0	0	1	1	1	0	

# Aufgabe 1 – Bin- und Hex Arithmetik

13

- ▶ Prozessoren arbeiten mit einer endlichen Wortbreite, was den darstellbaren Zahlenbereich einschränkt. Welche Auswirkungen kann dies bei der Subtraktion und Addition haben?
- ▶ Können Sie sich vorstellen, wie diese Probleme in der Praxis gelöst werden?

**Hinweis:** Was passiert beim Verlassen des Zahlenbereichs

# Aufgabe 1 – Bin- und Hex Arithmetik

14

- ▶ Auswirkungen bei der Subtraktion und Addition mit endlicher Wortbreite

Im Folgenden sei eine Wortbreite von 8 bit angenommen.

**Problem 1:** Addition vorzeichenloser Zahlen ( $[0,255]$ ):

$128 + 128 = 0!$

$$\begin{array}{r} 10000000 \\ 10000000 + \\ \hline 100000000 = \end{array}$$

**Lösung:** **Carry Flag** im Prozessor (Übertrag von der  $n$ -ten in die  $(n + 1)$ -te Stelle)

Programmierer kann Flag explizit überprüfen und verarbeiten

Dadurch lassen sich auch breitere Worte per Software emulieren (z. B. 64-Bit-Worte auf einem 32-Bit-Prozessor).

# Aufgabe 1 – Bin- und Hex Arithmetik

15

- Auswirkungen bei der Subtraktion und Addition mit endlicher Wortbreite

**Problem 2:** Addition im 2er-Komplement ( $[-128, 127]$ ):

$127 + 3 = -126!$  (Überlauf)

Überlauf nur bei: Addition von Zahlen gleiches Vorzeichens

Subtraktion von Zahlen unterschiedliches Vorzeichens

$$\begin{array}{r} 0\ 1111111 \\ 0\ 0000011 + \\ \hline 01\ 111111 \\ 1\ 0000010 = \end{array}$$

**Lösung:** **Overflow-Flag** im Prozessor (zeigt Überlauf an)

Berechnung:  $OF = c_{n-1} \oplus c_n$

Im Beispiel:  $0 \oplus 1 = 1$  (Overflow gesetzt)

Dadurch lassen sich auch breitere Worte per Software emulieren.

# Zurück zu Gleitkommazahlen



# Aufgabe 2 – Addition & Subtraktion

17

- ▶ Gegeben seien folgende Zahlen im IEEE Standard 754 (einfache Genauigkeit):

$$x_1 = 0\ 1001\ 1001\ 0000\ 1001\ 1001\ 0000\ 0000\ 000_2$$

$$x_2 = 0\ 1001\ 1001\ 1111\ 1111\ 0000\ 0000\ 0000\ 000_2$$

- ▶ Addieren Sie die beiden Zahlen.

# Aufgabe 2 – Addition & Subtraktion

18

## Addition im IEEE-Format:

### 1: Transformation

Rechtsschieben der kleineren Zahl auf den Exponenten der Größeren.

### 2: Addition der Mantissen

Falls Ergebnis  $< 0$ : setze Vorzeichenbit und bilde Zweierkomplement.

### 3: Normalisierung

A, Falls Ergebnis  $\geq 2$ : Rechtsschieben des Ergebnisses um eins (ggf. runden) und Inkrementierung des Exponenten.

B, Falls Ergebnis  $< 1$ : Linksschieben des Ergebnisses um eins und Dekrementierung des Exponenten.

C, Wiederhole A bzw. B bis Ergebnis = 0 oder  $1 \leq \text{Ergebnis} < 2$

### 4: Behandlung von Sonderfällen (Überlauf, Unterlauf, Null)

# Aufgabe 2 – Addition & Subtraktion

19

## ► Addition von

$$x_1 = 0\ 1001\ 1001\ 0000\ 1001\ 1001\ 0000\ 0000\ 000_2$$

$$x_2 = 0\ 1001\ 1001\ 1111\ 1111\ 0000\ 0000\ 0000\ 000_2$$

## 1: Transformation

$x_1$  und  $x_2$  besitzt den gleichen Exponenten, weswegen ein Verschieben entfällt.

$$E = 1001\ 1001$$

# Aufgabe 2 – Addition & Subtraktion

► Addition von

$$x_1 = 0\ 1001\ 1001\ 0000\ 1001\ 1001\ 0000\ 0000\ 000_2$$

$$x_2 = 0\ 1001\ 1001\ 1111\ 1111\ 0000\ 0000\ 0000\ 000_2$$

## 2: Addition der Mantissen

	$m_1:$	$1,0000\ 1001\ 1001_2$
+	$m_2:$	$1,1111\ 1111\ 0000_2$
<hr/>		
	$m_{\text{ges}}:$	$11,0000\ 1000\ 1001_2$

# Aufgabe 2 – Addition & Subtraktion

21

## ► Addition von

$$x_1 = 0\ 1001\ 1001\ 0000\ 1001\ 1001\ 0000\ 0000\ 0000_2$$

$$x_2 = 0\ 1001\ 1001\ 1111\ 1111\ 0000\ 0000\ 0000\ 0000_2$$

## 3: Normalisierung

$$11,0000\ 1000\ 1001_2 \rightarrow 1,10000\ 1000\ 1001_2 \cdot 2^1$$

Exponent anpassen

$$2^1 \cdot 2^E = 2^{E+1}$$

$$E = 1001\ 1001 + 1 = 1001\ 1010$$

# Aufgabe 2 – Addition & Subtraktion

22

## ► Addition von

$$x_1 = 0\ 1001\ 1001\ 0000\ 1001\ 1001\ 0000\ 0000\ 0000_2$$

$$x_2 = 0\ 1001\ 1001\ 1111\ 1111\ 0000\ 0000\ 0000\ 0000_2$$

## 4: Zusammenfassung

VB: 0 (Addition zweier positiver Zahlen)

$e_1$ : 1001 1010<sub>2</sub>

$m_{\text{ges}}$ : 1,1000 0100 0100 1000 0000 000<sub>2</sub>

Ergebnis  $x_3$ : 0 1001 1010 1000 0100 0100 1000 0000 000<sub>2</sub>

# Aufgabe 2 – Addition & Subtraktion

23

Subtraktion im IEEE-Format (Abbildung auf die Addition  $a + (-b) = c$ ):

1: Transformation

Rechtsschieben der kleineren Zahl auf den Exponenten der Größeren.

▶ 2: Zweierkomplement von  $b$  bilden

3: Addition der Mantissen

Falls Ergebnis  $< 0$ : setze Vorzeichenbit und bilde Zweierkomplement.

4: Normalisierung

A, Falls Ergebnis  $\geq 2$ : Rechtsschieben des Ergebnisses um eins (ggf. runden) und Inkrementierung des Exponenten.

B, Falls Ergebnis  $< 1$ : Linksschieben des Ergebnisses um eins und Dekrementierung des Exponenten.

C, Wiederhole A bzw. B bis Ergebnis = 0 oder  $1 \leq \text{Ergebnis} < 2$

5: Behandlung von Sonderfällen (Überlauf, Unterlauf, Null)

# Aufgabe 2 – Addition & Subtraktion

- ▶ „Subtraktion“  $x_1 + x_3$  von

$$x_1 = 0\ 1001\ 1001\ 0000\ 1001\ 1001\ 0000\ 0000\ 0000_2$$

$$x_3 = 1\ 1001\ 0100\ 0011\ 0110\ 0000\ 0000\ 0000\ 0000_2$$

## 1: Transformation

Die Exponenten sind unterschiedlich, also müssen wir den kleineren Exponenten auf den größeren schieben.

$$1001\ 1001_2 - 1001\ 0100_2 = 101_2 (= 5_{10})$$

Wir müssen also  $m_3$  5 Stellen nach rechts schieben ( $2^5$  ist positiv!)

$$m_3 \text{ (alt)} = 1,0011\ 0110\ 0000\ 0000\ 0000\ 0000_2$$

$$m_3 \text{ (neu)} = 0,0000\ 1001\ 1011\ 0000\ 0000\ 0000_2$$

# Aufgabe 2 – Addition & Subtraktion

25

- ▶ „Subtraktion“  $x_1 + x_3$  von

$$x_1 = 0\ 1001\ 1001\ 0000\ 1001\ 1001\ 0000\ 0000\ 0000_2$$

$$x_3 = 1\ 1001\ 0100\ 0011\ 0110\ 0000\ 0000\ 0000\ 0000_2$$

2: Zweierkomplement der Mantisse von  $x_3$  (VB = 1)

Wir erweitern die Mantissen von  $x_1$  und  $x_3$  um ein führendes **Vorzeichenbit**, um das Vorzeichen des Ergebnisses der Addition überprüfen zu können.

$m_3$ :                    **00**, 0000 1001 1011<sub>2</sub>

Einerkomplement:    **11**, 1111 0110 0100<sub>2</sub>

Zweierkomplement: **11**, 1111 0110 0101<sub>2</sub>

Erinnerung  
Erste 1 von rechts  
suchen. Linken Rest  
invertieren

# Aufgabe 2 – Addition & Subtraktion

- ▶ „Subtraktion“  $x_1 + x_3$  von

$$x_1 = 0\ 1001\ 1001\ 0000\ 1001\ 1001\ 0000\ 0000\ 0000_2$$

$$x_3 = 1\ 1001\ 0100\ 0011\ 0110\ 0000\ 0000\ 0000\ 0000_2$$

## 3: Addition der Mantissen

$$\begin{array}{r} m_1: \quad 01, 0000\ 1001\ 1001_2 \\ + m_{3, neu}: \quad 11, 1111\ 0110\ 0101_2 \\ \hline m_{ges}: \quad 00, 1111\ 1111\ 1110_2 \end{array}$$

Diesmal ist das Vorzeichenbit nicht gesetzt (deshalb auch Erweiterung um führende 0), also müssen wir vom Ergebnis nicht mehr das Zweierkomplement bilden.

# Aufgabe 2 – Addition & Subtraktion

27

- ▶ „Subtraktion“  $x_1 + x_3$  von

$$x_1 = 0\ 1001\ 1001\ 0000\ 1001\ 1001\ 0000\ 0000\ 000_2$$

$$x_3 = 1\ 1001\ 0100\ 0011\ 0110\ 0000\ 0000\ 0000\ 000_2$$

## 4: Normalisierung

Wir müssen unsere Zahl 1 nach links schieben, um auf das Format 1,... zu kommen.

$$0,1111\ 1111\ 1110_2$$

$$\rightarrow 1,111\ 1111\ 1110_2 \cdot 2^{-1}$$

Exponent anpassen

$$2^{-1} \cdot 2^E = 2^{E-1}$$

$$E = 1001\ 1001 - 1 = 1001\ 1000$$

# Aufgabe 2 – Addition & Subtraktion

28

- ▶ „Subtraktion“  $x_1 + x_3$  von

$$x_1 = 0\ 1001\ 1001\ 0000\ 1001\ 1001\ 0000\ 0000\ 0000_2$$

$$x_3 = 1\ 1001\ 0100\ 0011\ 0110\ 0000\ 0000\ 0000\ 0000_2$$

## 5: Zusammenfassung

VB: 0 (da Ergebnis positiv; vgl. vorheriger Schritt)

$e_1$ : 1001 1000<sub>2</sub>

$m_{\text{ges (2er)}}$ : 01, 1111 1111 1100 0000 0000 0000<sub>2</sub>

Ergebnis  $x_3$ : 0 1001 1000 1111 1111 1100 0000 0000 0000<sub>2</sub>

# Denkpause

Aufgabe:

Ein Bauer möchte sein Feld in vier gleich große, identisch geformte Teilstücke aufteilen, um sie seinen vier Kindern zu vermachen.

Wie muss aufgeteilt werden?



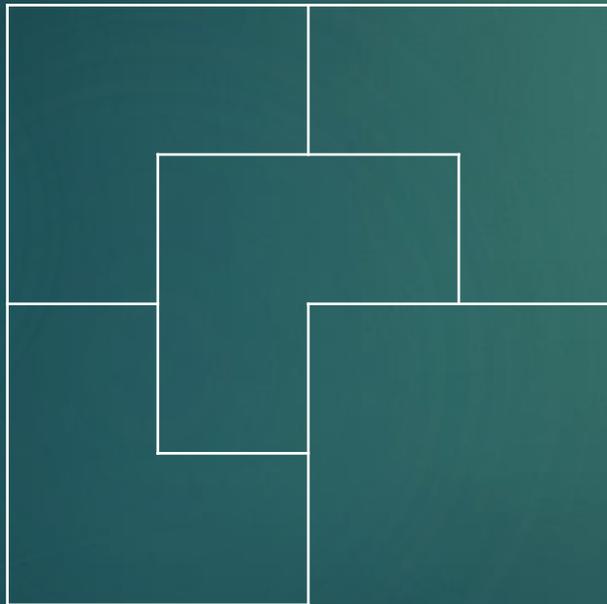
Skizze des Felds



# Denkpause

Lösung:

Folgende Aufteilung erfüllt alle Kriterien.



30

# Aufgabe 3 – Multiplikation

31

- ▶ Gegeben seien folgende Gleitkommazahlen:

$$x_1 = 0\ 1001\ 000\ 1001\ 1011_2$$

$$x_2 = 1\ 1001\ 010\ 1110\ 1000_2$$

- ▶ Für die beiden Zahlen gilt:

- ▶ Vorzeichen (V): 1 Bit breit (1: negativ)

- ▶ Exponent (E): 7 Bit breit

- ▶ Mantisse (M): 8 Bit breit (1,M wie beim IEEE Format üblich)

- ▶ Anordnung: VEM

- ▶ Berechnen Sie  $x_1 \cdot x_2$ .

# Aufgabe 3 – Multiplikation

32

## ► IEEE-Format im generellen Fall

Bisher: wir befinden uns auf einem 32 bit System, deshalb wird ein float mit 32 bit dargestellt

Es gibt nach IEEE aber auch Darstellungsmöglichkeiten für z.B. 8 bit, 16 bit und 64 bit Gleitkommazahlen.

Je größer die Anzahl der Bit, desto größer ist die Genauigkeit und die Auflösung der Gleitkommazahl.

Hier haben wir ein Beispiel für die 16 bit Darstellung:

0 1001 000 1001 1011<sub>2</sub>

Die Berechnung funktioniert analog in allen Fällen. Achtung beim BIAS!

# Aufgabe 3 – Multiplikation

33

## Multiplikation im IEEE-Format

### 1: Multiplikation der Mantissen

Mantissen beide im Format 1,...

### 2: Addition der Biased Exponenten

vgl. hier Potenzrechenregeln  $2^a \cdot 2^b = 2^{a+b}$

der Bias kommt als Hilfsgröße in beiden Faktoren vor, deshalb einmal den Bias (hier 63) abziehen.

### 3: Normalisierung der Mantisse

d.h. auf Form 1,MANTISSE bringen (ggf. Exponenten verschieben)

### 4: Vorzeichen getrennt behandeln

# Aufgabe 3 – Multiplikation

34

- ▶ Multiplikation folgender Gleitkommazahlen:

$$x_1 = 0\ 1001\ 000\ 1001\ 1011_2$$

$$x_2 = 1\ 1001\ 010\ 1110\ 1000_2$$

## 1: Multiplikation der Mantissen

Wir müssen beiden Mantissen um die **führende 1 des Formats erweitern**, um die Korrektheit der Multiplikation zu gewährleisten.

$$1,1001\ 1011_2 \cdot 1,1110\ 1000_2 \quad (\text{vgl. letzte Übung für ausführlichen Rechenweg})$$

$$= 11,0000111101111000_2$$

# Aufgabe 3 – Multiplikation

35

- ▶ Multiplikation folgender Gleitkommazahlen:

$$x_1 = 0\ 1001\ 000\ 1001\ 1011_2$$

$$x_2 = 1\ 1001\ 010\ 1110\ 1000_2$$

2: Addition der Biased Exponenten

$$\text{BIAS} = 2^{n-1} - 1 = 2^6 - 1 = 63_{10} = 111111_2$$

$$1001\ 000_2 + 1001\ 010_2 - \text{BIAS} \quad (\text{vgl. letzte Übung für ausführlichen Rechenweg})$$

$$= 10010\ 010_2 - 111111_2 \quad (\text{BIAS einsetzen})$$

$$= 1010011_2$$

# Aufgabe 3 – Multiplikation

- ▶ Multiplikation folgender Gleitkommazahlen:

$$x_1 = 0\ 1001\ 000\ 1001\ 1011_2$$

$$x_2 = 1\ 1001\ 010\ 1110\ 1000_2$$

## 3: Normalisierung der Mantisse

Mantisse:  $11,0000111101111000_2$  (Zielformat 1,... + 8 NKS)

$$1,10000111 \cdot 2^1$$

Exponent:  $1010011_2$

Wir verrechnen jetzt noch den Shift-Faktor der Mantisse mit dem Exponenten.

$$1010011_2 + 1_{10} = 1010011_2 + 1_2 = 1010100_2$$

# Aufgabe 3 – Multiplikation

37

- ▶ Multiplikation folgender Gleitkommazahlen:

$$x_1 = 0 \ 1001 \ 000 \ 1001 \ 1011_2$$

$$x_2 = 1 \ 1001 \ 010 \ 1110 \ 1000_2$$

4: Vorzeichen getrennt behandeln

0 = +, 1 = -      - · + = -      Vorzeichen folglich 1 (auch XOR möglich)

5: Zusammensetzen

VB: 1    Exponent:  $1010100_2$     Mantisse:  $1,1000 \ 0111_2$

Ergebnis:     $1 \ 1010100 \ 1000 \ 0111_2$

# Aufgabe 3 – Multiplikation

38

- ▶ Multiplikation folgender Gleitkommazahlen:

$$x_1 = 0 \ 1001 \ 000 \ 1001 \ 1011_2$$

$$x_2 = 1 \ 1001 \ 010 \ 1110 \ 1000_2$$

Grad des Fehlers

Unser Ergebnis weicht von dem wahren Ergebnis, wenn man  $x_1$  und  $x_2$  in Dezimalzahlen wandelt und diese verrechnet, ab. Dies hängt mit dem Verlust der Genauigkeit der IEEE-Darstellung zusammen. Denn nicht alle Zahlen lassen sich durch gerade Mal 16 Bit wiedergeben!

Wahres Ergebnis:  $822 \cdot -3904 = -3\ 209\ 088$

Unser Ergebnis:  $1 \ 1010100 \ 1000 \ 0111_2 = -2^{21} \cdot 1 \ 135/256 = -3\ 203\ 072$

# Aufgabe 4 - Assoziativität

39

- ▶ Die Operationen Addition und Multiplikation auf Operanden in einer Fließkommadarstellung sind normalerweise **nicht assoziativ**.
- ▶ Experimentieren Sie mit einem PC-Tabellenkalkulationsprogramm um dieses zu belegen. Bestimmen Sie dadurch auch die Anzahl der Bits, die zur Speicherung der Mantisse verwendet werden.
- ▶ (Beispiel: LibreOffice und Excel liefern bei der Berechnung von:  $10^{20} + 17 - 10 - 10^{20} + 130$  als Ergebnis 130 - dies übrigens ohne irgendeine Warnung.)

# Aufgabe 4 - Assoziativität

- Die folgenden Berechnungen sind mit Excel 2013 vorgenommen worden:

	<b>Formel</b>	<b>Ergebnis</b>
A	$10^{20} - 10^{20} + 17 - 10 + 130$	= 137
B	$10^{15} - 10^{15} + 17 - 10 + 130$	= 137
C	$10^{16} - 10^{16} + 17 - 10 + 130$	= 137
D	$10^{20} + 17 - 10 - 10^{20} + 130$	= 130
E	$10^{15} + 17 - 10 - 10^{15} + 130$	= 137
F	$10^{16} + 17 - 10 - 10^{16} + 130$	= 136
G	$10^{20} + 17 + 130 - 10^{20} - 10$	= -10
H	$10^{15} + 17 + 130 - 10^{15} - 10$	= 137
I	$10^{16} + 17 + 130 - 10^{16} - 10$	= 136

# Aufgabe 4 - Assoziativität

41

$$D: 10^{20} + 17 - 10 - 10^{20} + 130 = 130$$

Berechnung von D ist falsch, da die Zahl  $10^{20}$  zwar darstellbar ist, aber auf Grund der Inkontinuität der Gleitkommazahlen  $10^{20} + 17$  nicht mehr (auf  $10^{20}$  gerundet).

$$F: 10^{16} + 17 - 10 - 10^{16} + 130 = 136$$

Berechnung von F (bzw. auch I) ist falsch, da bei der Darstellung von  $10^{16}$  die niederwertigste 1 von  $17_{10} = 10001_2$  bei der Addition auf die 54te Stelle der Mantisse fallen würde (IEEE 754 – 64 bit hat 53 bit für die Mantisse) und somit abgeschnitten wird.

Somit ist klar, welche Genauigkeit (nämlich 64 bit) verwendet wurde.

A scenic view of Paris, France, featuring the Eiffel Tower in the background and a dense cityscape in the foreground. The scene is framed by vibrant autumn leaves in shades of yellow and orange in the upper left corner. The overall atmosphere is warm and nostalgic.

Vielen Dank für eure geschätzte  
Aufmerksamkeit!

42

Friedrich-Alexander-Universität Erlangen-Nürnberg  
Jan Spieck

"En un mot, l'homme doit se créer sa propre  
essence; c'est en se jetant dans le monde,  
en y souffrant, en y luttant qu'il se définit peu  
à peu; et la définition demeure toujours  
ouverte"

*A propos de l'existentialisme - Mise au point. Action, no. 17, 29 décembre 1944.*