

**Allgemeine Hinweise:** Versuchen Sie beim Lösen der Beispiele *keine elektronischen Hilfsmittel* zu verwenden – beim Test werden Sie diese nicht zur Verfügung haben.

Damit ein Beispiel anerkannt wird, muss ein Lösungsweg erkennbar sein und es müssen alle enthaltenen Teilaufgaben gelöst sein. Ein korrektes Endergebnis ist nicht zwingend erforderlich!

Deadline für das Ankreuzen und Hochladen der Lösungen in TUWEL: Mittwoch, 26.10.2016, 13:00 Uhr (Toleranzzeit ohne Gewähr, verspätete Abgaben per E-Mail werden ausnahmslos nicht akzeptiert!)

### Aufgabe 1: Binäre Gleitpunkt-Arithmetik – Addition & Subtraktion

Gegeben sind die Zahlen  $A = (0.00131470)_8$  und  $B = (-0.09A10)_{16}$ .

Es gilt folgendes Gleitpunktformat:

$\mathbb{F}(2, 11, -14, 15, true)$  mit Formatbreite 16 Bit und *impliziter* Darstellung des ersten Bits. Mit Ausnahme der kleineren Formatbreite ist dieses Gleitpunktformat analog zum IEEE 754 *Single Precision*-Format aufgebaut.

- a) Stellen Sie  $A$  und  $B$  in diesem Gleitpunktformat dar! Verwenden Sie Guard- und Round-Digit sowie das Sticky-Bit zur Vermeidung von numerischen Ungenauigkeiten (vgl. *Informatik Grundlagen*, 5. Auflage, Kapitel 8.6.4). Runden Sie mittels *round to nearest* zusammen mit *round to even*.

- b) Berechnen Sie anschließend  $A+B$  sowie  $A-B$  und stellen Sie das Ergebnis wieder als Gleitpunktzahl im angegebenen Format dar. Runden Sie die Ergebnisse wieder mittels *round to nearest* in Kombination mit *round to even*.

## Aufgabe 2: Binäre Gleitpunkt-Arithmetik – Multiplikation & Division

Gegeben sind die folgenden, im 16-Bit-Gleitpunktformat aus Aufgabe 1 codierten Zahlen:

$$A = 0\ 11011\ 0111110011$$

$$B = 1\ 01110\ 1010110000$$

$$C = 1\ 10101\ 0111110000$$

Führen Sie die nachfolgenden Berechnungen durch! Verwenden Sie Guard- und Round-Digit sowie das Sticky-Bit zur Vermeidung von numerischen Ungenauigkeiten. Runden Sie mittels *round to nearest* zusammen mit *round away from zero*.

*Hinweis:* Beachten Sie das implizite erste Bit.

a)  $A * B$

b)  $\frac{A}{C}$

### **Aufgabe 3: Binäre Gleitpunkt-Arithmetik – Sonderfälle**

Gegeben sind die folgenden im 16-Bit-Gleitpunktformat aus Aufgabe 1 codierten Zahlen:

$$\begin{aligned} A &= 0\ 00011\ 0100000000 \\ B &= 0\ 00001\ 0010101101 \\ C &= 1\ 11101\ 1001100111 \\ D &= 1\ 00001\ 0010100001 \end{aligned}$$

Führen Sie mit den Zahlen folgende Berechnungen durch und codieren Sie das Ergebnis jeweils im angegebenen Gleitpunktformat! Runden Sie mittels round toward plus infinity (= gerichtetes Aufrunden)!

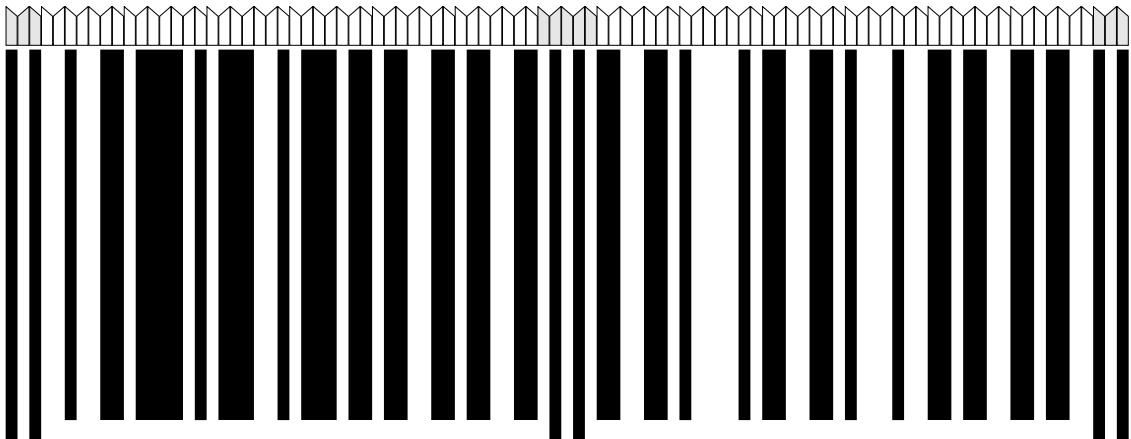
a)  $A * B$

b)  $B + D$

c)  $C/D$

**Aufgabe 4: EAN-13-Code**

a) Decodieren Sie die nachfolgende EAN!



b) Codieren Sie die EAN 1 782008 17201! Berechnen Sie hierzu die Prüfziffer und tragen Sie den resultierenden Code in den vorgedruckten Raster ein.

*Hinweis: Rand- und Trennzeichen sind grau hinterlegt.*



**Aufgabe 5: NRZ-Code**

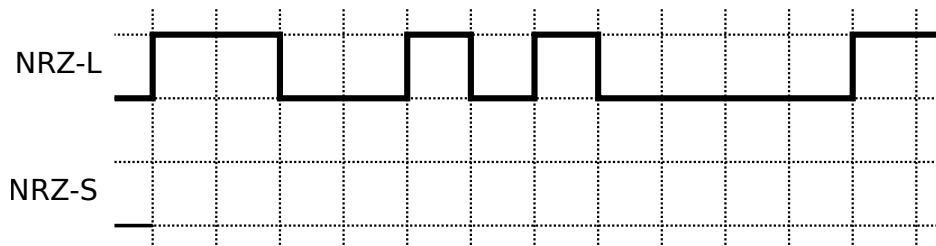
Gegeben ist der nachfolgende Signalverlauf mit Pegel 1 (*high*) und Pegel 2 (*low*).



a) Interpretieren Sie den Signalverlauf in NRZ-M-Codierung und geben Sie die decodierte 0/1-Folge an!

b) Interpretieren Sie den Signalverlauf in NRZ-L-Codierung und geben Sie die decodierte 0/1-Folge an!

c) Zeichnen Sie zum nachfolgend gegebenen Signalverlauf in NRZ-L-Codierung darunter den entsprechenden Signalverlauf in NRZ-S-Codierung!



### Aufgabe 6: CRC-Code

a) Sie sollen das ASCII-Zeichen  $d$  mittels CRC-16 absichern. Welche Bitfolge ergibt das?

b) Sie haben ein mit CRC-CCITT codiertes Zeichen als Bitfolge '0000 0101 0101 0000 1010 0101' empfangen. Begründen Sie, ob ein Fehler vorliegt oder nicht!

### Aufgabe 7: Hamming-Distanz

Gegeben ist ein Code mit fünf Codewörtern: 1101001, 1110001, 0001100, 1001100 und 1010101

a) Berechnen Sie die Hamming-Distanz zwischen den einzelnen Codewörtern und vervollständigen Sie die nachfolgende Distanz-Matrix!

	1101001	1110001	0001100	1001100	1010101
1101001					
1110001					
0001100					
1001100					
1010101					

b) Geben Sie den Hamming-Abstand  $D$  des Codes an!

c) Wie viele Bits braucht man mindestens, um einen Code für acht Codewörter zu entwerfen, der einen Hamming-Abstand von  $D = 2$  aufweist?

### Aufgabe 8: Hamming-Code

Ein Hamming-Code hat 4 Prüfbit.

- a) Wie lang (in Bit) können die Codewörter maximal sein? Wieviele Bit stehen maximal für Daten zur Verfügung?
  
- b) Wie lauten die Gleichungen für die nötigen Prüfbits dieses Codes?

- c) Listen Sie alle gültigen Codewörter eines Hamming-Code mit 7 Codebit auf!

*Hinweis: Es werden möglicherweise nicht alle Zeilen benötigt.*

$c_1$	$c_2$	$c_3$	$c_4$	$c_5$	$c_6$	$c_7$

- d) Überprüfen Sie anhand von zwei Beispielen exemplarisch, ob es sich bei dem Code aus c) um einen linearen Code handeln könnte!
  
  
- e) Decodieren und ggf. korrigieren Sie das empfangene Codewort  $c_1c_2c_3c_4c_5c_6c_7 = 1101110$  unter der Annahme, dass es sich beim Absenden um ein Codewort aus dem Code aus c) handelte und maximal ein Bit gestört wurde!