

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.

Deadline für das Ankreuzen und Hochladen der Lösungen in TUWEL: Montag, 04.04.2016, 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 $A = (-0.0014BA26C)_{16}$ und $B = (0.025764)_8$.

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 $B-A$ 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:

$$A = 0\ 10011\ 0011110111$$

$$B = 1\ 00011\ 0010011000$$

$$C = 1\ 10101\ 1110110000$$

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:

$$\begin{aligned} A &= 0\ 00011\ 0100000000 \\ B &= 0\ 01010\ 1110101101 \\ C &= 0\ 11110\ 0000000000 \\ D &= 0\ 00110\ 0100101100 \end{aligned}$$

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)!

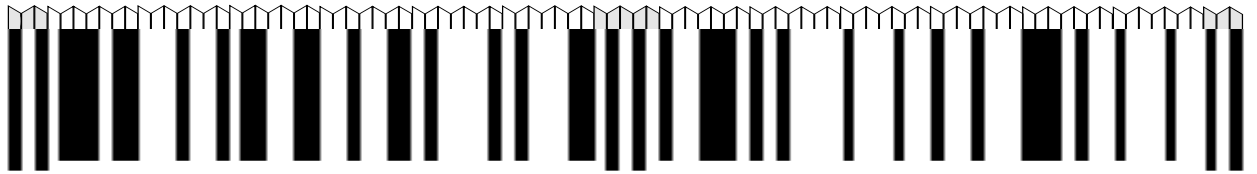
a) $A * B$

b) $B + D$

c) A/C

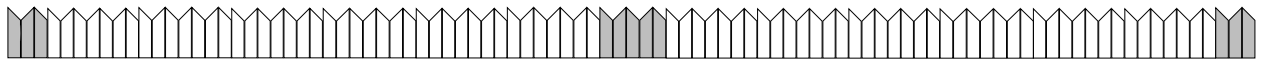
Aufgabe 4: EAN-13-Code

- a) Decodieren Sie die nachfolgende EAN!



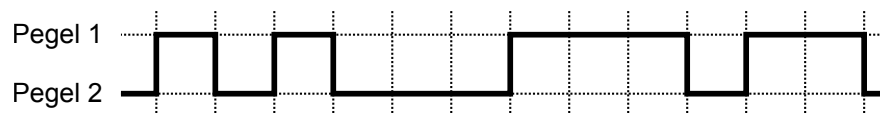
- b) Codieren Sie die EAN 7 235953 52525! Berechnen Sie hierzu die Prüfziffer und tragen Sie den resultierenden Code in den vorgedruckten Raster ein.

Hinweis: Rand- und Trennzeichen sind grau hinterlegt.

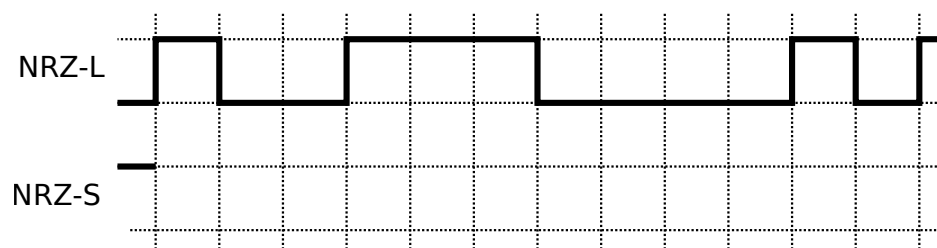


Aufgabe 5: NRZ-Code

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! Gehen Sie davon aus, dass die Folge bei Pegel *low* mit Wert 0 startet.
- 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-CCITT absichern. Welche Bitfolge ergibt das?

b) Sie haben ein mit CRC-12 codiertes Zeichen als Bitfolge '00 0010 1000 0001 0001' empfangen. Begründen Sie, ob ein Fehler vorliegt oder nicht!

Aufgabe 7: Hamming-Distanz

Gegeben ist ein Code mit fünf Codewörtern: 0000000, 0101010, 1010101, 0000111 und 1111111

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

	0000000	0101010	1010101	0000111	1111111
0000000					
0101010					
1010101					
0000111					
1111111					

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

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

Aufgabe 8: Hamming-Code

Es soll ein Hamming-Code für 5 Datenbits konstruiert werden.

- a) Wie viele Prüfbits werden benötigt? Wie hoch ist die Anzahl der resultierenden Codebits?
- b) Wie lauten die Gleichungen für die nötigen Prüfbits dieses Codes?
- c) Überprüfen Sie mittels der Prüfgleichungen die in der Tabelle angegebenen Codewörter und tragen Sie den errechneten Korrekturindikator in die Tabelle ein!

c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9	Korrekturindikator	Datenwort
0	0	0	0	0	0	0	0	0		
0	1	1	1	0	0	0	1	1		
1	1	1	1	1	0	1	0	0		
0	1	0	1	0	0	1	1	0		

- d) Decodieren und ggf. korrigieren Sie die Codewörter aus Punkt c) unter der Annahme, dass maximal ein Bit gestört wurde! Tragen Sie die resultierenden Datenwörter in der Spalte *Datenwort* ein.
- e) Handelt es sich bei diesem Hamming-Code um einen linearen Code? Überprüfen Sie die Linearität anhand von zwei gültigen Codewörtern exemplarisch!