GDS – Übungsblatt 2

Aufgabe 1a

```
Länge des Exponenten = 16 (Gesamt) - 10 (Mantisse) - 1 (Vz) = 5 Bit,
```
daher Exzess = $2^{5-1} - 1 = 15_{10} = 01111_2$

A:

```
0.   | 0   | 0   | 1   | 3   | 1   | 4   | 7
000.  000   000   001   011   001   100   111

Normalisierung:
```
$$(0.000\,000\,001\,011\,001\,100\,111)*2^0=(1.011\,001\,100\,111)*2^{-9}$$
```
Exponent:
 01111
-01001
 00110

Darstellung ohne Rundung, mit GRS:
Vz  Exp   Mantisse   G R S
 0  00110 0110011001 1 1 0

Darstellung mit Rundung (round to nearest, Mantisse++):
Vz  Exp   Mantisse
 0  00110 0110011010
```

B:

```
0.   | 0    | 9    | A    | 1    | 0
0000. 0000   1001   1010   0001   0000

Normalisierung:
```
$$(0.0000\,1001\,1010\,0001)_2*2^0=(1.001\,1010\,0001)*2^{-5}$$
```
Exponent:
 01111
-00101
 01010

Darstellung ohne Rundung, mit GRS:
Vz  Exp   Mantisse   G R S
 1  01010 0011010000 1 0 0
```

Darstellung mit Rundung (Grenzfall => round to even):

```
Vz  Exp   Mantisse
 1  01010 0011010000
```

Aufgabe 1b

Nachfolgend wird die Mantisse immer mit explizitem ersten Bit I angeschrieben. Am Ende der Berechnung wird das Bit wieder entfernt.

```
  Vz  Exp   I Mantisse
A: 0  00110 1 0110011010
B: 1  01010 1 0011010000
```

Angleichen der Exponenten (A: exp + 4):

```
  Vz  Exp   I Mantisse   G R S
A: 0  01010 0 0001011001 1 0 1
B: 1  01010 1 0011010000
```

A + B:

|B| > |A|, daher Mantisse A von Mantisse B abziehen.
Vz auf 1 setzen, da B negativ.

```
      Vz  Exp   I Mantisse   G R S
    B: 1  01010 1 0011010000
    A: 0  01010 0 0001011001 1 0 1
A + B: 0  00000 1 0001110110 1 0 1
```

I ist 1, daher normalisiert.
Gerundet und mit implizitem ersten Mantissenbit:

```
A + B: 1  01010 0001110110
```

A - B:

B negativ, daher Mantissen addieren.
Vz auf 0, da A − B = A + |B| > 0

```
      Vz  Exp   I Mantisse   G R S
    A: 0  01010 0 0001011001 1 0 1
    B: 1  01010 1 0011010000
A − B: 0  01010 1 0100101001 1 0 1
```

I ist 1, daher normalisiert.
Gerundet und mit implizitem ersten Mantissenbit:

```
A − B: 0  01010 0100101010
```

Aufgabe 2

Zahlen mit explizitem ersten Mantissenbit I:

```
   Vz Exp   I Mantisse
A = 0 11011 1 0111110011
B = 1 01110 1 1010110000
C = 1 10101 1 0111110000
```

Aufgabe 2a: A * B

Multiplikation der Mantissen (einige Zwischensummen):

```
 1,0111110011 * 1,1010110000
 1 0111110011
   10111110011
     10111110011
       10111110011
        10111110011
                   0000
10,0111110010010001000 0
```

Addition der Exponenten:

```
 11011
 01110
101001
 -1111  <= Weniger ein Exzess
011010
    +1  <= Normalisierung
011011
```

Zusammensetzen:

```
Vz   Exp I Mantisse   G R S
 1 11011 1 0011111001 0 0 1 00010000
 1 11011 1 0011111001 0 0 1
 1 11011 1 0011111001                <= Runden
```

Ohne explizitem ersten Mantissenbit:

```
 1 11011 0011111001
```

Aufgabe 2b: A / C

```
Division der Mantissen (A / C):
10111110011 : 10111110000 = 1,0000000010000001...
         11000000000
        -10111110000
         0000001000000000000
               -10111110000
               001000010000...

G = R = 0
S = 1

Subtraktion der Exponenten:
  exp_A: 11011
- exp_C: 10101
        00110
        +1111  <= +Exzess
        10101

Zusammensetzen (eh normalisiert):

Vz    Exp I Mantisse   G R S
 1  10101 1 0000000010 0 0 1
 1  10101 1 0000000010        <= Runden

Ohne explizitem ersten Mantissenbit:
 1  10101 0000000010
```

Aufgabe 3

```
  Vz   Exp I Mantisse   G R S
A: 0 00011 1 0100000000
B: 0 00001 1 0010101101
C: 1 11101 1 1001100111
D: 1 00001 1 0010100001
```

Aufgabe 3a: A * B

Multiplikation der Mantissen:

```
 1,0010101101 * 1,0100000000
 10 010101101
    10010101101
1,0 11101100001

 00011
 00001
   100
-01111  <= -Exponent
```

$-01011_2 = -11_{10}$

Summe der Exponenten (minus 1 Exzess) ergibt Unterlauf. Unterschied zwischen e_{min} und der Summe ist 12_{10}.

```
            1,011101100001 =>
0,00000000000101110110001
```

Neuer Exponent = e_{min}, aber denormalisierte Zahl, "Exponent" in der Repräsentation daher e_{min} - 1

```
  Vz   Exp I Mantisse   G R S
A: 0 00000 0 0000000000 0 1 1
```

Aufrunden:

```
  Vz   Exp I Mantisse
A: 0 00000 0 0000000001
```

Aufgabe 3b: B + D

```
  Vz   Exp I Mantisse
B: 0 00001 1 0010101101
D: 1 00001 1 0010100001
```

D < 0, daher Mantisse D von Mantisse B abziehen:

```
 1,0010101101
-1,0010100001
 0,0000001100
```

Beide Exponenten sind bereits e_{min}, daher Unterlauf bei Exponenten bei Normalisierung, daher Denormalisierung benötigt.

```
Vz    Exp I Mantisse
 0 00000  0 0000001100
```

Runden hat sich auch erledigt.

Aufgabe 3c: C / D

```
  Vz   Exp I Mantisse
C: 1 11101 1 1001100111
D: 1 00001 1 0010100001
```

Mantissen dividieren:

```
 11001100111 : 10010100001 = 1,01100010000101...
-10010100001
   11100011000
  -10010100001
    10011101110
   -10010100001
        10011010000
       -10010100001
             10111100000
            -10010100001
               10011111100
              -10010100001
                   1011011
```

Exponenten subtrahieren:

```
 11101
-00001
 11100
+01111  <= +Exzess
101011
```

Überlauf bei Exponent und Mantisse schon normalisiert, daher ist diese Zahl in diesem Gleitpunktformat nicht darstellbar => +infinity

```
Vz   Exp Mantisse
 0 11111 0000000000
```

Aufgabe 4: EAN-13-Code

a) Decodieren Sie die nachfolgende EAN!

1 2A 3A 5B 7A 1B 7B 1C 3C 1C 7C 1C 1C

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

AABABB CCC

Hinweis: Rand- und Trennzeichen sind grau hinterlegt.

7 8 2 0 0 8 1 7 2 0 1 P=7

$1 + 7\cdot 3 + 8 + 2\cdot 3 + 0 + 0\cdot 3 + 8 + 1\cdot 3 + 7 + 2\cdot 3 + 0 + 1\cdot 3 + p \equiv 0 \mod 10$

$63 + p \equiv 0 \mod 10$

$p = 7$

Aufgabe 5: NRZ-Code

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

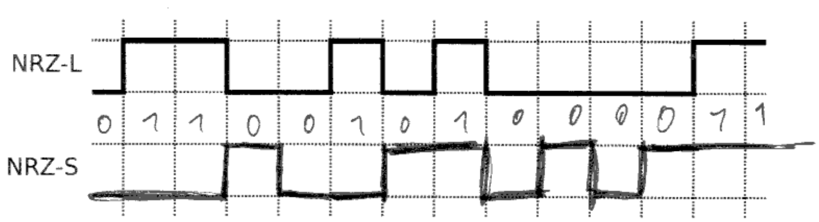
a) Interpretieren Sie den Signalverlauf in NRZ-M-Codierung und geben Sie die decodierte 0/1-Folge an!

? 1 0 0 1 0 1 1 1 0 0 0 1 1

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

1 0 0 0 1 1 0 1 0 0 0 0 1 0

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



Aufgabe 6a

ASCII 'd' = 64_{10}
6   4
0110 0100

$G(x)=x^{16}+x^{15}+x^{2}+x^{1}+1$
$M(x)=x^{6}+x^{5}+x^{2}$
$r=16$

$x^{r}*M(x)=x^{16}(x^{6}+x^{5}+x^{2})=x^{22}+x^{21}+x^{18}$

$(x^{22}+x^{21}+x^{18}):(x^{16}+x^{15}+x^{2}+x^{1}+1)=x^{6}+x^{2}+x^{1}+1$
$-\underline{x^{22}+x^{21}+x^{8}+x^{7}+x^{6}}$
$x^{18}+x^{8}+x^{7}+x^{6}$
$-\underline{x^{18}+x^{17}+x^{4}+x^{3}+x^{2}}$
$x^{17}+x^{8}+x^{7}+x^{6}+x^{4}+x^{3}+x^{2}$
$-\underline{x^{17}+x^{16}+x^{3}+x^{2}+x}$
$x^{16}+x^{8}+x^{7}+x^{6}+x^{4}+x$
$-\underline{x^{16}+x^{15}+x^{2}+x+1}$
$x^{15}+x^{8}+x^{7}+x^{6}+x^{4}+x^{2}+1$

$x^{22}+x^{21}+x^{18}$
$-\underline{x^{15}+x^{8}+x^{7}+x^{6}+x^{4}+x^{2}+1}$
$x^{22}+x^{21}+x^{18}+x^{15}+x^{8}+x^{7}+x^{6}+x^{4}+x^{2}+1$

11001001000000111010101

Aufgabe 6b

$G(x)=x^{16}+x^{12}+x^{5}+1$
$T(x)=x^{18}+x^{16}+x^{14}+x^{12}+x^{7}+x^{5}+x^{2}+1$

$(x^{18}+x^{16}+x^{14}+x^{12}+x^{7}+x^{5}+x^{2}+1):(x^{16}+x^{12}+x^{5}+1)=x^{2}+1$
$\underline{x^{18}+x^{14}+x^{7}+x^{2}}$
$x^{16}+x^{12}+x^{5}+1$

Kein Rest, kein erkennbarer Fehler.

Aufgabe 7a

	1101001	1110001	0001100	1001100	1010101
1101001	0	2	4	3	4
1110001	2	0	6	5	2
0001100	4	6	0	1	4
1001100	3	5	1	0	3
1010101	4	2	4	3	0

Aufgabe 7b

= Kleinster Wert aus der Tabelle = 1

Aufgabe 7c

3 Bit / Wort => D = 1
Parity Bit dazu: 4 Bit / Wort und D = 2

Aufgabe 8a

Letztes Prüfbit auf Position 2^3, Codewort hört an Position 2^4-1 auf,
daher maximale Länge 2^4-1 = 15.
Vier Prüfbits, Länge 15 (maximal),
daher 15 – 4 = 11 Datenbits (maximal).

Aufgabe 8b

$$p_1 = c_3 \otimes c_5 \otimes c_7 \otimes c_9 \otimes c_{11} \otimes c_{13} \otimes c_{15}$$
$$p_2 = c_3 \otimes c_6 \otimes c_7 \otimes c_{10} \otimes c_{11} \otimes c_{14} \otimes c_{15}$$
$$p_3 = c_5 \otimes c_6 \otimes c_7 \otimes c_{12}$$
$$p_4 = c_9 \otimes c_{10} \otimes c_{11} \otimes c_{12}$$

Aufgabe 8c

$c_1 = p_1$	$c_2 = p_2$	c_3	$c_4 = p_3$	c_5	c_6	c_7
0	0	0	0	0	0	0
1	1	0	1	0	0	1
0	1	0	1	0	1	0
1	0	0	0	0	1	1
1	0	0	1	1	0	0
0	1	0	0	1	0	1
1	1	0	0	1	1	0
0	0	0	1	1	1	1
1	1	1	0	0	0	0
0	0	1	1	0	0	1
1	0	1	1	0	1	0
0	1	1	0	0	1	1
0	1	1	1	1	0	0
1	0	1	0	1	0	1
0	0	1	0	1	1	0
1	1	1	1	1	1	1

Aufgabe 8d

- (0000000 + 1111111) mod 2 = 1111111
- (0001111 + 1110000) mod 2 = 1111111

Aufgabe 8e

Prüfbit

```
 n: 1 2 3 4 5 6 7
```
c_n: 1 1 0 1 1 1 0

$p_1 = c_3 \otimes c_5 \otimes c_7 = 0 \otimes 1 \otimes 0 = 1$ ... Passt
$p_2 = c_3 \otimes c_6 \otimes c_7 = 0 \otimes 1 \otimes 0 = 1$ ... Passt
$p_3 = c_5 \otimes c_6 \otimes c_7 = 1 \otimes 1 \otimes 0 = 0$ ... Fehler!

Mehrere Möglichkeiten für Fehler:

- c_5 ist gestört
- c_6 ist gestört
- c_7 ist gestört
- p_3 ist gestört

Da alle Datenbits durch andere Prüfbits "abgedeckt" sind, ist der Fehler in p_3.

Korrigiertes Codewort: 1100110