Aufgabe 29

Binärwort	Interpretation als			
	Binärzahl	Zweierkomplement	Einerkomplement	Betrag-Vorzeichen
0000	0	0	0	0
0001	1	1	1	1
0010	2	2	2	2
0011	3	3	3	3
0100	4	4	4	4
0101	5	5	5	5
0110	6	6	6	6
0111	7	7	7	7
1000	8	-8	-7	-0
1001	9	-7	-6	-1
1010	10	-6	-5	-2
1011	11	-5	-4	-3
1100	12	-4	-3	-4
1101	13	-3	-2	-5
1110	14	-2	-1	-6
1111	15	-1	-0	-7

Aufgabe 30

Für alle $a \in \{0,\ldots,2^{n-1}-1\}$ und alle $b \in \{-2^{n-1},\ldots,-1\}$ gilt:

(1) $a > a+b$, d.h. $(a+b) \in \{-2^{n-1},\ldots,2^{n-1}\}$

(2) sowie

$$K_2^{-1}\left(\left(K_2(a)+K_2(b)\right) \bmod 2^n\right)$$
$$= K_2^{-1}\left(\left(a+\left(2^n-|b|\right)\right) \bmod 2^n\right)$$
$$= K_2^{-1}\left(\left(2^n+a-|b|\right) \bmod 2^n\right)$$

Nun unterscheide ich zwei Fälle:

a) $a \geq |b|$: Es gilt: $(a-|b|) \in \{0,\ldots,2^{n-1}-1\}$ und dadurch

$$\left(2^n+a-|b|\right) \bmod 2^n = \left(a-|b|\right) \bmod 2^n = a-|b| = a+b,$$

mit $(a+b) \in \{0,\ldots,2^{n-1}-1\}$, was ergibt:

$K_2^{-1}(a+b) = a+b$ (siehe (1))

b) $a < |b|$: Es gilt: $(a-|b|) \in \{-2^{n-1},\ldots,-1\}$ sowie $2^{n-1} < 2^n+a-|b| < 2^n$ und dadurch

$$\left(2^n+a-|b|\right) \bmod 2^n = 2^n+a-|b|,$$

mit $(a+b) \in \{-2^{n-1},\ldots,-1\}$, was ergibt:

$K_2^{-1}\left(2^n+a-|b|\right) = a-|b| = a+b$

Damit wurde gezeigt, dass in der Zweierkomplementdarstellung die Addition einer positiven mit einer negativen Zahl (mit Einschränkung $a \in \{0,\ldots,2^{n-1}-1\}$ und $b \in \{-2^{n-1},\ldots,-1\}$) stets erlaubt ist.

Aufgabe 31

a) Das Zweierkomplement überdeckt die Zahlenbereiche:
8 Bit: -128 ... +127

16 Bit: -32.768 ... +32.767
32 Bit: -2.147.483.648 ... +2.147.483.647

b) 0: `0000 0000`
-7: `1111 1001`
–13: `1111 0011`

c) Man untergliedert das 8-Bit-Zweierkomplement in zwei Teile, so dass es die Form $ab_6\ b_5\ b_4\ b_3\ b_2\ b_1\ b_0$ annimmt. Es errechnet sich der Dezimalwert aus:

$$K_2^{-1}\left(ab_6b_5b_4b_3b_2b_1b_0\right)=a\cdot\left(-2^7\right)+\sum_{n=0}^{6}2^n\cdot b_n$$

Da a=1 ist, genügt es, die Binärzahl, die die letzten 7 Bits darstellen (also b_6 bis b_0) auszurechnen und davon 2^7=128 zu subtrahieren:

$1111\ 0101_2 = 0111\ 0101_2 – 2^7{}_{10} = 117 – 128 = -11$
$1111\ 1010_2 = 0111\ 1010_2 – 2^7{}_{10} = 122 – 128 = -6$

Man kann sich noch etwas Rechenaufwand sparen, wenn man die Beziehung

$$K_2^{-1}\left(11\ldots11b_mb_{m-1}\ldots b_1b_0\right)=K_2^{-1}\left(1b_mb_{m-1}\ldots b_1b_0\right)$$

ausnutzt und so von den beiden gegebenen 8-Bit-Zweierkomplementen eigentlich nur die niederwertigsten fünf betrachten muss:

$10101_2 = 0101_2 – 2^4{}_{10} = 5 – 16 = -11$
$11010_2 = 1010_2 – 2^4{}_{10} = 10 – 16 = -6$

<u>Aufgabe 32</u>

a) Die Zweierkomplementdarstellung hat einen Zahlenbereich, der immer um 1 größer ist als der des Einerkomplementes, da letzteres zwei Darstellungen der Zahl Null vorsieht: $K_1(+0)\neq K_1(-0)$. Diese Nicht-Eindeutigkeit ist auf die Symmetrie des Einerkomplementes zurückzuführen.

b) Eine negative Zahl hat stets des MSB gesetzt, eine positive dagegen nie. Somit entspricht eine 1 im MSB stets einem negativen Vorzeichen, eine 0 stets dem positiven.

c) Ich beschränke mich auf ein Beispiel für 4-Bit-Zahlen:

$$\begin{array}{rr} K_2(6_{10}) = & 0110_2 \\ \underline{+\ K_2(4_{10}) =} & \underline{0100_2} \\ & 1010_2 \rightarrow K^{-1}{}_2(1010_2) = -6 \end{array}$$

Scheinbar ergibt 6 + 4 = -6, was aber falsch ist. Der Overflow ist immer an einem Vergleich der MSB der Summanden und der Summe zu erkennen. Gilt MSB(Summand 1) = MSB(Summand 2) = 0 und gleichzeitig MSB(Summe) = 1, so trat ein Überlauf auf und das Ergebnis ist ungültig.

d) Wiederum ein 4-Bit-Beispiel:

$$\begin{array}{rr} K_2(-6_{10}) = & 1010_2 \\ \underline{+\ K_2(-4_{10}) =} & \underline{1100_2} \\ & 1\ \ 0110_2 \rightarrow K^{-1}{}_2(0110_2) = +6 \end{array}$$

(die führende 1 beim Ergebnis rutscht als Carry-Out aus der Rechnung heraus und gehört nicht zur Summe)

Das scheinbare Ergebnis ist erneut falsch. Der Underflow lässt sich, genau wie der Overflow, an der Konstellation der MSB erkennen. Das hierfür zutreffende Muster ist MSB(Summand 1) = MSB(Summand 2) = 1 und gleichzeitig MSB(Summe) = 0.