Lösungsvorschlag zur Sitzung am 23.11.2020 – Grundlagen IV: Zweierkomplementdarstellung, Subtraktion im Binärsystem

Abgabe auf Ilias bis 30.11.2020, 10.00 Uhr

Aufgabe 1

Wie werden folgende Zahlen im Rechner dargestellt? Verwenden Sie falls nötig das Zweierkomplement. Notieren Sie alle Zwischenschritte und Rechenwege. Die Ergebnisse allein reichen nicht.

a) 11_{10}

b) -11_{10}

c) -51_{10}

d) -256_{10}

Lösung:

a) 11_{10}

Rechnung:

11 : 2 = 5 Rest: 1
5 : 2 = 2 Rest: 1
2 : 2 = 1 Rest: 0
1 : 2 = 0 Rest: 1

$11_{10} = 1011_Z$

b) -11_{10}

11_{10} = 0000 1011_2

Rechnung: siehe a)

Einerkomplement:	1111 0100
Zweierkomplement:	1111 0100
	+ 0000 0001
	1111 0101

-11_{10} = 1111 0101_Z

c) -51_{10}

$51_{10} = 0011\ 0011_2$

Rechnung:

51 : 2 = 25 Rest: 1
25 : 2 = 12 Rest: 1
12 : 2 = 6 Rest: 0
6 : 2 = 3 Rest: 0
3 : 2 = 1 Rest: 1
1 : 2 = 0 Rest: 1

EK: 1100 1100
ZK: 1100 1100
+ 0000 0001
1100 1101

$-51_{10} = 1100\ 1101_Z$

d) -256_{10}

$256_{10} = 0001\ 0000\ 0000_2$

Rechnung:

256 : 2 = 128 Rest: 0
128 : 2 = 64 Rest: 0
64 : 2 = 32 Rest: 0
32 : 2 = 16 Rest: 0
16 : 2 = 8 Rest: 0
8 : 2 = 4 Rest: 0
4 : 2 = 2 Rest: 0
2 : 2 = 1 Rest: 0
1 : 2 = 0 Rest: 1

EK: 1110 1111 1111
ZK: 1110 1111 1111
+ 0000 0000 0001
1 1111 111
1111 0000 0000

$-256_{10} = 1111\ 0000\ 0000_Z$

Aufgabe 2

Berechnen Sie im Binärsystem. Verwenden Sie falls nötig das Zweierkomplement. Notieren Sie alle Zwischenschritte und Rechenwege. Die Ergebnisse allein reichen nicht.

a) $10_{10} - 3_{10}$

b) $21_{10} - 9_{10}$

Lösung:

a) 10_{10} - 3_{10}

$3_{10} = 0000\ 0011_2$

Rechnung:

3 : 2 = 1 Rest: 1
1 : 2 = 0 Rest: 1

$10_{10} = 0000\ 1010_2$

Rechnung:

10 : 2 = 5 Rest: 0
5 : 2 = 2 Rest: 1
2 : 2 = 1 Rest: 0
1 : 2 = 0 Rest: 1

EK: 1111 1100

ZK: 1111 1100
+ 0000 0001
1111 1101

$-3_{10} = 1111\ 1101_Z$

Rechnung für: $10_{10} + (-3_{10})$

```
  0000 1010
+ 1111 1101
 1 1 1 1 1
 10000 0111
```

Da beim Rechnen nur 8 Bit verwendet wurden, darf das Ergebnis auch nur 8 Bit groß sein. Das Ergebnis hat 9 Bit, weshalb die Stelle ganz links wegfällt und somit ist das Ergebnis:

$10_{10} + (-3_{10}) = 7_{10}$

$(0000\ 1010_Z) + (1111\ 1101_Z) =$ **$0000\ 0111_Z$**

b) 21_{10} - 9_{10}

9_{10} = 0000 1001_2

Rechnung:

9 : 2 = 4 Rest: 1
4 : 2 = 2 Rest: 0
2 : 2 = 1 Rest: 0
1 : 2 = 0 Rest: 1

EK: 1111 0110

ZK: 1111 0110
+ 0000 0001
1111 0111

-9_{10} = 1111 0111_Z

21_{10} = 0001 0101_2

Rechnung:

21 : 2 = 10 Rest: 1
10 : 2 = 5 Rest: 0
5 : 2 = 2 Rest: 1
2 : 2 = 1 Rest: 0
1 : 2 = 0 Rest: 1

Rechnung für: 21_{10} + (-9_{10})

  0001 0101
+ 1111 0111
 1 111 111
1 0000 1100

Da beim Rechnen nur 8 Bit verwendet wurden, darf das Ergebnis auch nur 8 Bit groß sein. Das Ergebnis hat 9 Bit, weshalb die Stelle ganz links wegfällt und somit ist das Ergebnis:

21_{10} + (-9_{10}) = 12_{10}

(0001 0101_Z) + (1111 0111_Z) = **0000 1100_Z**

Aufgabe 3

Berechnen Sie im Binärsystem. Wie werden die folgenden Zahlen im Rechner abgebildet, wenn 16 Bit zur Zahldarstellung zur Verfügung stehen? Verwenden Sie falls nötig das Zweierkomplement. Notieren Sie alle Zwischenschritte und Rechenwege. Die Ergebnisse allein reichen nicht.

a) $-56_{10} + 99_{10}$

b) $1025_{10} + 187_{10}$

c) $15_{10} - 21_{10}$

d) $-100_{10} - 256_{10}$

Lösung:

a) $-56_{10} + 99_{10}$

$56_{10} = 0000\ 0000\ 0011\ 1000_2$

Rechnung:

56 : 2 = 28 Rest: 0
28 : 2 = 14 Rest: 0
14 : 2 = 7 Rest: 0
7 : 2 = 3 Rest: 1
3 : 2 = 1 Rest: 1
1 : 2 = 0 Rest: 1

$99_{10} = 0000\ 0000\ 0110\ 0011_2$

Rechnung:

99 : 2 = 49 Rest: 1
49 : 2 = 24 Rest: 1
24 : 2 = 12 Rest: 0
12 : 2 = 6 Rest: 0
6 : 2 = 3 Rest: 0
3 : 2 = 1 Rest: 1
1 : 2 = 0 Rest: 1

EK: 1111 1111 1100 0111
ZK: 1111 1111 1100 0111
+ 0000 0000 0000 0001
1111 1111 1100 1000

$-56_{10} = 1111\ 1111\ 1100\ 1000_Z$

Rechnung für: $(-56)_{10} + 99_{10}$

1111 1111 1100 1000
+ 0000 0000 0110 0011
1 0000 0000 0010 1011

Da beim Rechnen nur 16 Bit verwendet wurden, darf das Ergebnis auch nur 16 Bit groß sein. Das Ergebnis hat 17 Bit, weshalb die Stelle ganz links wegfällt und somit ist das Ergebnis:

$(-56)_{10} + 99_{10} = 43_{10}$

$(1111\ 1111\ 1100\ 1000_Z) + (0000\ 0000\ 0110\ 0011_Z) =$ **$0000\ 0000\ 0010\ 1011_Z$**

Probe für **$0000\ 0000\ 0010\ 1011_Z$: Umrechnung ins Dezimalsystem**

$1*2^5 + 0*2^4 + 1*2^3 + 0*2^2 + 1*2^1 + 1*2^0$

$= 32 + 0 + 8 + 0 + 2 + 1$

$= 43_{10}$

b) $1025_{10} + 187_{10}$

$1025_{10} = 10000000001_2$

Rechnung:
1025 : 2 = 512 Rest: 1
512 : 2 = 256 Rest: 0
256 : 2 = 128 Rest: 0
128 : 2 = 64 Rest: 0
64 : 2 = 32 Rest: 0
32 : 2 = 16 Rest: 0
16 : 2 = 8 Rest: 0
8 : 2 = 4 Rest: 0
4 : 2 = 2 Rest: 0
2 : 2 = 1 Rest: 0
1 : 2 = 0 Rest: 1

$187_{10} = 1011\ 1011_2$

Rechnung:
187 : 2 = 93 Rest: 1
93 : 2 = 46 Rest: 1
46 : 2 = 23 Rest: 0
23 : 2 = 11 Rest: 1
11 : 2 = 5 Rest: 1
5 : 2 = 2 Rest: 1
2 : 2 = 1 Rest: 0
1 : 2 = 0 Rest: 1

Rechnung für: $1025_{10} + 187_{10}$

```
  100 0000 0001
+     1011 1011
           1 1
---------------
  100 1011 1100
```

Auffüllen auf 16Bit: 0000 0100 1011 1100

c) $15_{10} - 21_{10}$

$21_{10} = 0000\ 0000\ 0001\ 0101_2$

Rechnung: siehe 2.a)

$15_{10} = 0000\ 0000\ 0000\ 1111_2$

Rechnung:
15 : 2 = 7 Rest: 1
7 : 2 = 3 Rest: 1
3 : 2 = 1 Rest: 1
1 : 2 = 0 Rest: 1

```
EK:  1111 1111 1110 1010
ZK:  1111 1111 1110 1010
   + 0000 0000 0000 0001
     1111 1111 1110 1011
```

$-21_{10} = 1111\ 1111\ 1110\ 1011_Z$

Rechnung für: $15_{10} + (-21)_{10}$

```
  0000 0000 0000 1111
+ 1111 1111 1110 1011
              1 1 1 1
---------------------
  1111 1111 1111 1010
```

Probe für **$1111\ 1111\ 1111\ 1010_Z$: Umrechnung ins Dezimalsystem**

```
EK: 0000 0000 0000 0101
ZK: 0000 0000 0000 0101
  + 0000 0000 0000 0001
                    110
```

$1*2^2 + 1*2^1 + 0*2^0$
$= 4 + 2 + 0$
$= 6_{10}$

d) -100_{10} - 256_{10}

100_{10} = 0000 0000 0110 0100_2

Rechnung:

100 : 2 = 50 Rest: 0
50 : 2 = 25 Rest: 0
25 : 2 = 12 Rest: 1
12 : 2 = 6 Rest: 0
6 : 2 = 3 Rest: 0
3 : 2 = 1 Rest: 1
1 : 2 = 0 Rest: 1

EK: 1111 1111 1001 1011
ZK: 1111 1111 1001 1011
+ 0000 0000 0000 0001
1111 1111 1001 1100

-100_{10} = 1111 1111 1001 1100_Z

Rechnung für: $(-100)_{10} + (-256)_{10}$

1111 1111 1001 1100
+ 1111 1111 0000 0000
1111 1110 1001 1100

Probe für **1111 1110 1001 1100_Z: Umrechnung ins Dezimalsystem**

EK: 000 0001 0110 0011
ZK: 000 0001 0110 0011
+ 000 0000 0000 0001
1 0110 0100

$1*2^8 + 0*2^7 + 1*2^6 + 1*2^5 + 0*2^4 + 0*2^3 + 1*2^2 + 0*2^1 + 0*2^0$
= 256 + 0 + 64 + 32 + 0 + 0 + 4 + 0 + 0
= 356_{10}

-256_{10} = **1111 0000 0000_Z**

Rechnung: siehe 1.d)

Auf 16 Bit auffüllen: 1111 1111 0000 0000_z