Nils Ulrich
William Djalal

Rechnerarchitekturen
Übung 4

1.1)

1. 23 + 81

B+V:

```
  0 0 1 0 1 1 1
  0 0 0 1 0 1 1
+ 1 0 1 0 0 0 1
    ^ ^ ^ ^
  1 0 1 1 1 0 0
  1 1 0 1 0 0 0
```

Zweierdarstellung:

```
  0 0 0 1 0 1 1 1
+ 0 1 0 1 0 0 0 1
      ^ ^ ^ ^
  0 1 1 0 1 0 0 0
```

Exzessdarstellung:

```
  0 0 0 1 0 1 1 1
+ 0 1 0 1 0 0 0 1
      ^ ^ ^ ^
  0 1 1 0 1 0 0 0
```

2. 36 − 14

B+V:

```
  0 1 0 0 1 0 0
- 0 0 0 1 1 1 0
      ^ ^
  0 1 0 0 1 1 0
```

Zweierkomplement:

36 + (−14) ⇒ 14 in −14 ⇒ alle Bits flippen +1

−14 =

```
  1 1 1 1 0 0 1
+ 0 0 0 0 0 0 1
      ^
  1 1 1 1 0 0 1 0
```

⇒

```
  0 0 1 0 0 1 0 0
+ 1 1 1 1 0 0 1 0
    ^ ^ ^
  0 0 0 1 0 1 1 0
```

negative Zahl ⇒ Bits flippen +1

```
   11101001
+  00000001
         1
-----------
   11101010
```

Exzessdarstellung:

36 - 14   Offset = 14
               => damit 22 - 0 gerechnet wird
                  => einfacher

```
  00010110
- 00000000
----------
  00010110
```

3. 72 - 87

B+V

```
  1001000
- 1010111
 111 111
---------
 11110001
```

Zweierkomplement

72 + (-87) => Bits flippen + 1

```
-87 =   10101000
      + 00000001
      ----------
        10101001
```

=> 72 + (-87):

```
   01001000
 + 10101001
       1
 ----------
   11110001
```

Exzessdarstellung: offset = ~~-87~~ 72

=> ~~159 - 0~~ (vereinfacht)

~~10011111~~
~~- 00000000~~

```
     0 - 15
   00000000
=> - 00001111
   11111111
   ---------
   11110001
```

```
   00001110
 + 00000001
 ----------
   00001111
```

4. -113 -37 ← Zweierkomplement

-113 :

00001110
+ 00000001
00001111

-37 :

00100101
+ 00000001
1
00100110

=> (-113) + (-37) :

00001111
+ 00100110
11
00101101

B + V :

0001111
- 0100101
11
11101010

=>

0010101
+ 0000001
0010100

Exzessdarstellung:

Offset ~~-37~~ +113

=> ~~76~~ 0

=> 0 + 76

00000000
+ 01001100
01001100

3)

=> 76 - 113

01001100
- 11110001
1111 11
101011011