# 2 Zahlen codieren (fortgeschrittene Berechnungen) (Kap. 1 + 2)

|  |  |
| --- | --- |
| **erwünschtes Ergebnis** | Sie wissen, wie Computer mit Fliesskommaberechnungen beliebige Werte berechnen. |
| **Zeitaufwand** | 75 Min. |
| **Ausgangslage** | Alle Berechnungen, die Sie bisher unternommen haben, geschah mit natürlichen Zahlen. Für weiterführenden Berechnungen benötigen Sie jedoch auch die reellen und negative Zahlen. |
| **Aufgabe** | Arbeiten Sie die nachfolgende Theorie durch und lösen Sie die zugehörigen Aufgaben. Machen Sie Notizen zum Thema in ihrem Moduljournal |
| **Hinweis** | ggf. Internet |
| **Ergebnis** | Sie haben im Moduljournal festgehalten, wie Fliesskommaberechnungen funktionieren und negative Zahlen im Computer gehandhabt werden. |

## Negative Zahlen

Alle bisherigen Zahlen die wir betrachtet haben waren positiv. Nun gibt es aber unendlich viele Anwendungen für negative Zahlen, wie stellen wir diese in der Informatik dar?

Eine naheliegende Möglichkeit negative Zahlen darzustellen wäre, die bestehenden Zahlen mittels eines Bits die Information «Positiv» oder «Negativ» mitzugeben. Beispielsweise so:

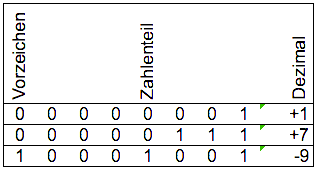


Abbildung 3.1-1

Allerdings ergibt sich bei dieser Lösung ein Problem bei der Berechnung. Im Dezimalsystem wird bspw. durch die Addition von negativen Zahlen eine Subtraktion erreicht. 55 - 13 ≙ 55 + (-13) = 42. Damit kann bei einem Prozessor eine Operation (Minusrechnen) eingespart werden, indem vor dem Addieren die zu subtrahierende Zahl negiert wird. Damit das funktioniert, wird die **Einerkomplementdarstellung** gebraucht.

## Einerkomplement: Wie funktioniert die Negierung einer Zahl?

1. Die Zahl als Binärzahl darstellen, dabei eine genügend grosse Länge auswählen 1310 = 000011012

2. Das Vorzeichen wechseln, indem alle Bits umgekehrt (invertiert) werden -1310 = 111100102

Hier eine Beispielberechnung mit 5510 - 1310

001101112 (Zahl 1)

+ 111100102 (Zahl 2)

111 11 (Übertrag, «behalte»)

|  |  |  |  |
| --- | --- | --- | --- |
|  | Bsp 1 | Bsp 2 | Bsp 3 |
| Zahl 1 | 0 | 01 | 011 |
| Zahl 2 | 1 | 01 | 011 |
| *Übertrag* | *0* | *10* | *110* |
| Summe | 1 | 10 | 110 |

-----------

001010012 (Summe)

Zur Erinnerung; So funktioniert die Addition von Binärzahlen:  
Die Summenbildung erfolgt wie bei den Dezimalzahlen von rechts nach links und zwar nach den Regeln die Sie rechts abgebildet sehen.

Was fällt Ihnen bei der obigen Berechnung im Einerkomplement auf?

**Aufgaben** **negative Zahlen im 1er Komplement**

Rechnen Sie die Zahlen links (1er Komplement) in Dezimalzahlen um!

000000002 ≙ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_10



111111112 ≙ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_10



000000012 ≙ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_10



111111102 ≙ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_10



Wie Sie aus Übung entnehmen, können Zahlen von -127 bis +127 dargestellt werden, aber es gibt zwei Darstellungsarten von 0 und die Addition ist fehlerhaft. Das Einerkomplement muss verbessert werden!

## Zweierkomplement: Wie funktioniert die Negierung einer Zahl?

1. Die Zahl als Binärzahl darstellen, dabei eine genügend grosse Länge auswählen 1310 = 000011012

2. Das Vorzeichen wechseln, indem alle Bits umgekehrt (invertiert) werden -1310 = 111100102

3. Eine 1 dazurechnen -1310 = 1111001**1**2

Nochmals die Beispielberechnung mit 5510 - 1310, diesmal aber mit der zusätzlichen 1

001101112 (55)

+ 1111001**1**2 (13)

111 11 (Übertrag)

-----------

001010102 (42)

Sie können feststellen, dass die Subtraktion bzw. die Addition im Zweierkomplement das richtige Resultat ergeben. Auch die Problematik mit den zwei Darstellungen von 0 fällt weg.

**Aufgabe zu negative Zahlen im 2er Komplement**

Lösen Sie folgende Aufgaben mit 8 Bit Dualzahlen im 2-Komplement:

|  |  |
| --- | --- |
| +3210  +1510 | +3210  -1510 |
| -3210  +1510 | -3210  -1510 |

**Aufgabe: Nennen Sie die grösste und kleinste Zahl (Dezimal) welche mit dem 2-Komplement mit 8 Bits dargestellt werden kann:**

Kleinste Zahl: \_\_\_\_\_\_\_\_\_\_\_\_\_\_10 \_\_\_\_\_\_\_\_\_\_\_\_\_\_2



Grösste Zahl: \_\_\_\_\_\_\_\_\_\_\_\_\_\_10 \_\_\_\_\_\_\_\_\_\_\_\_\_\_2

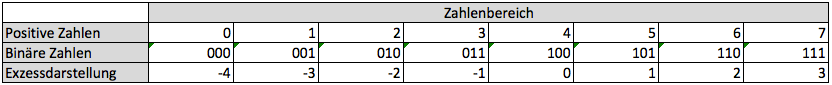


Was passiert, wenn Sie die Zahl 010 mittels 8 Bit ins 2-Komplement umrechnen?



## Exzessdarstellung (engl. excess oder offset binary)

Eine in der Praxis oft vorkommende Variante der Zahlendarstellung basiert auf der Verschiebung des Zahlenbereichs:



Wie einfach zu erkennen ist, wird der Zahlenbereich der positiven Zahlen bei der Exzessdarstellung verschoben. Dabei wird eine negative Zahl mehr dargestellt als positive. (-4 bis +3).

Sie können nun einfach vorhersehen, welche Zahlen mit 8 Bits dargestellt würden:

**28 = 256 -> es würden die Zahlen -128 bis +127 dargestellt.**

Die Umrechnung von Dezimalzahlen in die Exzessdarstellung und umgekehrt ist denkbar einfach; Es wird immer 2n-1 addiert oder subtrahiert.

**Aufgabe Dezimalzahlen 🡨 🡪 Exzessdarstellung umrechnen**

Berechnen Sie im Kopf die jeweils fehlenden Werte in der Tabelle!   
Es wird immer in einem 8 Bit System gerechnet.

|  |  |
| --- | --- |
| DEZIMAL | EXZESSDARSTELLUNG |
| Bsp: -34 | 01011110 |
| 100 |  |
| -63 |  |
|  | 11001100 |
|  | 00101101 |
| 1 |  |
| -1 |  |

Zur Kontrolle Ihrer Berechnungen in der Exzessdarstellung können Sie Formeln in Excel erstellen.

**Aufgabe: Wie können Sie in der Exzessdarstellung zwischen positiven und negativen Zahlen unterscheiden? Begründen Sie!**