# 2 Zahlen codieren (fortgeschrittene Berechnungen)

## 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:

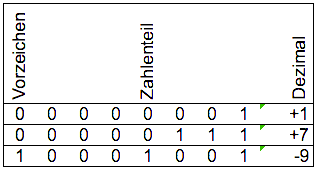


Abbildung 3.1-1

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)

|  |  |  |  |
| --- | --- | --- | --- |
|  | 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 ≙ 0\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_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:**

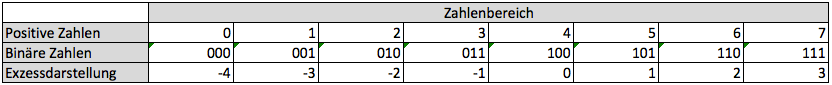
Kleinste Zahl: \_\_\_\_\_\_\_\_\_\_\_\_\_\_10 \_\_\_\_\_\_\_\_\_\_\_\_\_\_2

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!**

## Fliesskommazahlen (engl. floating point numbers)

Die Königsklasse der Zahlendarstellungen sind die Fliesskommazahlen. In der realen Welt treffen Sie diese viel öfters an, als einem bewusst ist.

* Oberfläche einer Kugel
* Grösse eines Baumes
* Fallzeit eines Blattes
* Zinsen pro Monat
* …

Fliesskommazahlen können sehr unterschiedlich daherkommen:

* Die Masse eines Kohlenstoffatoms beträgt etwa 0, 000 000 000 000 000 000 000 000 001 66 kg
* Der Eiffelturm in Paris hat eine Masse von ca. 7 300 000 kg
* Ein Virus hat einen Durchmesser von ca. 0, 000 000 1 m
* Die Geschwindigkeit des Lichts beträgt ca. 300 000 000 m/s
* Die Erde umläuft die Sonne in einem mittleren Abstand von etwa 150 000 000 000 m

Die sehr grossen bzw. sehr kleinen Masszahlen lassen sich in obiger Form gar nicht in den Taschenrechner eingeben, da er nicht genügend Stellen anzeigen kann! Deshalb bedient man sich einer anderen Zahlendarstellung, der sogenannten Exponentialschreibweise.

* Zur Erinnerung: 900000 = 9 × 100000 = 9 × 105 = 9.0 × 105 = 9.0 E5
* Der mittlere Sonnenabstand schreibt sich demzufolge als 1.5 E11
* und das Kohlenstoffatom würde so geschrieben: 1.66 E-27. mit dem Minus zwischen E und 27, wird gezeigt, dass sich das Komma nach links und nicht wie üblich nach rechts bewegt, die Zahl also kleiner als 1 ist.
* Eine negative Zahl wird so geschrieben: -9.0 E3 = -9000 bzw. -0.009 ist -9.0 E-3  
  NB: gewisse Programme/Rechner schreiben das als 9.e-3 bzw. -9.e-3

**Aufgabe Exponentialzahlen eingeben**

Rufen Sie den Windowsrechner auf, wählen Sie «Wissenschaftlich» als Darstellungsweise und geben Sie das obige Beispiel wie folgt ein:

**9.0 ± Exp 3 ±** Enter

Den umgekehrten Fall erreichen Sie mit F-E 🡪 -9000 F-E liefert -9.e+3

Begriffe: **Mantisse \* Basis Exponent**

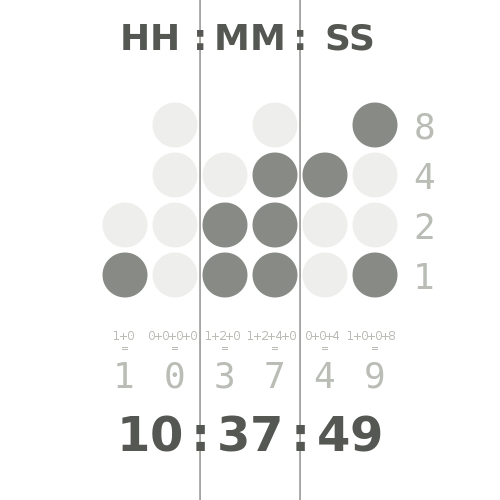
Ein Mikroprozessor rechnet grundsätzlich mit dem gleichen System, allerdings nicht dezimal, sondern binär (also Basis 2 statt 10). Das Rechnen im – für den Menschen nicht sehr anschaulichen – Binärsystem wird mit dem BCD-Code vereinfacht.

**Aufgabe sich ins Thema einarbeiten**

Arbeit Sie das Kapitel 2.4 in ihrem Lehrmittel durch. Arbeiten meint nicht einfach lesen, sondern die Erklärungen nachvollziehen und mit eigenen Beispielen ausprobieren!

Einen weiterführenden Artikel finden Sie in der Datei 🡪 **B21-114\_Gleitkommazahl.docx**

Betrachten Sie die Abbildung E-1 einer binären Uhr, die sexagesimale (60er-System) Werte dual darstellt. Jede Spalte bildet dabei eine dezimale Ziffer ab.

Man kann natürlich die Sekunden, Minuten und Stunden auch in 3 statt in 6 Spalten darstellen.

Überlegen Sie zuerst, wie viele Zeilen Sie dann benötigen.

Gehen Sie anschliessen auf die Webseite <http://binary.onlineclock.net/> auf der Sie eine Uhr in Echtzeit laufen sehen.

Beantworte Sie folgende Fragen:

* gibt es einen Fall, bei dem alle Felder rot sind?
* gibt es einen Fall, in dem eine Spalte komplett rot ist?
* Welche Uhrzeit belegt die meisten roten Felder?

Abbildung E-1

Damit alle Mikroprozessoren mit dem gleichen System arbeiten und die Zahlen unter den Rechnern austauschbar sind, wurde die Norm IEEE 754-2800 geschaffen. Die Norm deckt die 16, 32, 64 und 128 Bit-Version ab (wobei letztere im Alltag nur eine geringe Bedeutung hat).   
Die Abbildung E-2 zeigt die Paramter für 32 Bit (single) und 64 Bit (double).

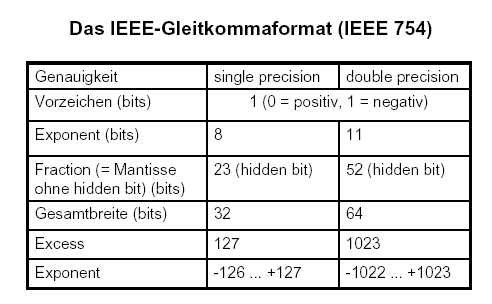


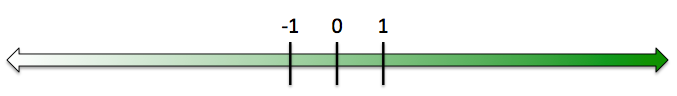
Abbildung E-2

Je nach Version wird eine andere Anzahl an Bits für Mantisse und Exponent verwendet.

**Aufgabe zu Fliesskommarechnung**

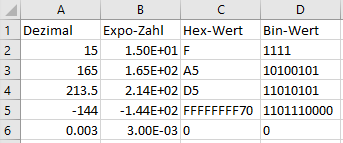
Ordnen Sie die Vorzeichenvarianten der entsprechenden Stelle auf dem Zahlenstrahl zu!

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Vorzeichen Mantisse |  |  |  |  |
| Vorzeichen Exponent |  |  |  |  |
| Zuordnen: | A | B | C | D |



**Aufgabe verschiedene Umrechnungsmethoden ausprobieren**

Öffnen Sie in Excel ein leeres Tabellenblatt und erstellen Sie die abgebildete Umrechnungstabelle



* Die Werte in Spalte B werden mit einer Formatierung gelöst.
* Die Werte in Spalte C erhalten Sie mit der Funktion =DEZINHEX().
* Die Werte in Spalte C erhalten Sie mit der Funktion =DEZINBIN().
* Alle Werte in den Spalten B, C und D, beziehen sich jeweils auf die Werte in Spalte A.

Spielen Sie mit den Formeln herum oder konsultieren Sie die Hilfe zu der jeweiligen Funktion, um die nachfolgenden Fragen zu beantworten:

* wo liegen die Grenzwerte bei der Hexadezimalumrechnung?
* wo liegen die Grenzwerte bei der Binärumrechnung?
* was passiert mit negativen Zahlen bei Hex- und Bin-Werten?
* was passiert mit Zahlen <1 bei Hex- und Bin-Werten?

Wenn Sie grössere Werte umrechnen wollen, funktioniert die obige Lösung nicht. Mit etlichen Zwischenschritten kann man das aber umgehen. In der Datei «**Basisumrechnung.xlsx**»   
finden Sie eine Lösung, bei der Sie eine (beinahe) beliebig grosse Zahl und die gewünschte Basis (2–36) eingeben können. Die Basis für Binärwerte wäre dann 2, das für Hexadezimalwerte 16.

Wenn Sie wissen wollen, wie das dahinterliegende System arbeitet, blenden Sie die ausgeblendeten Zeilen wieder ein. Geben Sie grosse Zahlen ein und beobachten Sie, was passiert.

Rufen Sie den Windowsrechner auf, wählen Sie «Programmierer» als Darstellungsweise und spielen Sie damit so lange herum, bis Sie in jede beliebige Richtung umrechnen können.

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung