bbs.eins.mainz
Berufsbildende Schule Technik

3 Rechnen mit Binärzahlen

In der Mathematik können wir beliebig große Zahlen darstellen. Innerhalb eines Computers ist der Speicherplatz aber beschränkt und somit auch die Anzahl der möglichen Stellen einer Binärzahl. Betrachten wir folgendes Beispiel:

# Bit	Dez	Bin
3	0	000
	1	001
	2	010
	3	011
	4	100
	5	101
	6	110
	7	111

Bei drei zur Verfügung stehenden Bits lassen sich also insgesamt nur acht verschiedene Zahlen (0-7) darstellen.

Wie werden Binärzahlen addiert?

In einem Stellenwertsystem, unabhängig von der Basis, können wir addieren, wie wir es aus der Grundschule kennen.
Dabei dürfen wir aber lediglich die zur Verfügung stehenden Ziffern verwenden.

Im *Oktal* können wir nur die Ziffern $0 - 7$ verwenden.

Für das Binärsystem ergeben sich somit folgende Regeln:

$$\begin{array}{cccc} 0 & + & 0 & = & 0 \\ 0 & + & 1 & = & 1 \\ 1 & + & 0 & = & 1 \\ 1 & + & 1 & = & 0 \end{array}$$

Man mag sich fragen, was ist da bei $1 + 1 = 0$ passiert?
Wie auch im Dezimalsystem kommt es zu einem **Übertrag**, wenn unsere Summe den Wert der *Basis* des Stellenwertsystems annimmt.

Wir betrachten das folgende Beispiel:

Dezimal: $7 + 11 = 18$
Binär: $111 + 1011 = 10010$

Wie kommen wir darauf:

$$\begin{array}{lccccc} & & 0 & 1 & 1 & 1 \\ + & & 1 & 0 & 1 & 1 \\ \text{Übertrag} & 1 & 1 & 1 & 1 & \\ \hline & 1 & 0 & 0 & 1 & 0 \end{array}$$

Was fällt bezüglich der Anzahl der Stellen der beiden Summanden und der Summe auf?

Die Regeln der Addition lassen sich wie folgt festhalten:

x_1	x_2	E	O
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

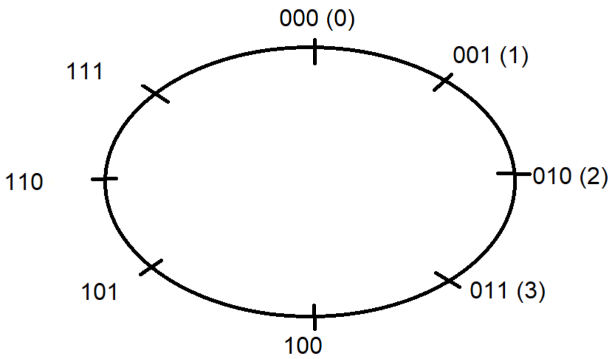
Wir haben nun die Addition von Binärzahlen kennengelernt und können diese ziemlich souverän anwenden.
Was wir aber noch nicht können, ist die Subtraktion. Der Computer selbst ist auch nicht in der Lage zu subtrahieren. Was intern passiert, wenn $3 - 2$ gerechnet wird, ist $3 + (-2)$. Der Computer addiert also die -2 zur 3 hinzu. Um das genauer zu verstehen, schauen wir uns die Negativdarstellung von Binärzahlen an.

4 Darstellung negativer Zahlen

Ausgangsproblem: *Wie stellt man eine negative Zahl im Binärsystem dar? z.B.: -5*

4.1 Negative Zahlen

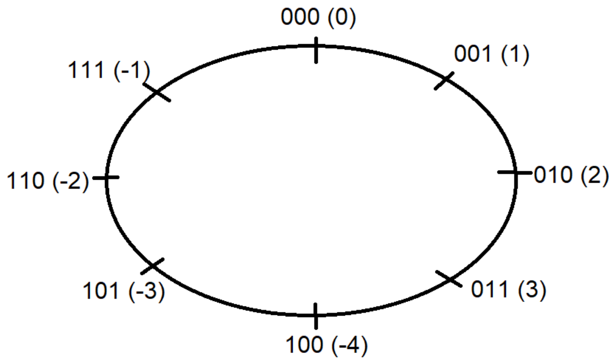
Im Allgemeinen haben wir für die Darstellung einer Zahl n Bit zur Verfügung. Das heißt, haben wir 3 Bit, so können wir die Zahlen von $0 - 7(2^3 - 1)$ darstellen.
Um nun aber positive und negative Zahlen darstellen zu können, reservieren wir das vorderste (**M**ost **S**ignificant **B**it) Bit als Vorzeichen. Hier gilt die Zuordnung:
$MSB = 0 \Rightarrow$ Zahl ist positiv.
$MSB = 1 \Rightarrow$ Zahl ist negativ.



Bleibt herauszufinden, welche Zahlen mit der $1\ldots$ Darstellung gegeben sind.
Hierfür betrachten wir Beispielhaft eine der vier Zahlen.
Beispiel: Welche Zahl müssen wir addieren, um wieder zur Null (0) zu gelangen:

?		1	0	1
+ 3		0	1	1
0	(1)	0	0	0

Auf die gleiche Weise können wir herausfinden, welche Zahlen die anderen $1\ldots$ Darstellen repräsentieren.



Bleibt zu klären, wie man die Negativdarstellung erhält, ohne jedes Mal zu zeichnen oder auszuprobieren.
Hierfür müssen wir zwei Schritte ausführen, welche nachfolgend kurz erklärt werden.

Einerkomplement

Wie eben bereits erwähnt, müssen die Zahl und ihre negative Zahl zusammen Null (0) ergeben. Zunächst bestimmen wir also die Binärzahl, so dass wir bei der größten mit n Bit darstellbaren Zahl sind.

Im Fall von 3 Bit ist das 111_2, bei 4 Bit dann 1111_2.

Diese ominöse Zahl erhält man, indem man **alle Bit invertiert** (also umdreht). Diese Darstellung nennt man auch **Einerkomplement**.
Beispiel: $101_2 \Rightarrow 010_2$.

Zahldarstellung				Einerkomplement			
Dez	Bin			Bin			Dez
0	0	0	0	1	1	1	7
1	0	0	1	1	1	0	6
2	0	1	0	1	0	1	5
3	0	1	1	1	0	0	4
4	1	0	0	0	1	1	3
5	1	0	1	0	1	0	2
6	1	1	0	0	0	1	1
7	1	1	1	0	0	0	0

Zweierkomplement

In der Tabelle ist erkennbar, eine Zahl und ihre Gegenkomponente ergeben die größte mit n Bit darstellbare Zahl. Um nun alle n auf Null (0) zu bringen und einen Übertrag an der höchsten Stelle zu erhalten, also aus dem darstellbaren Zahlenbereich raus zu fallen, müssen wir 1 addieren.

Beispiel: Wir haben 3 Bit und möchten das Zweierkomplement von 2.

2	0	1	0	*Zahl*
	1	0	1	*Einerkomplement*
$+$			1	*Addiere 1*
		1		*Übertrag*
	1	1	0	*Zweierkomplement*

Es gilt also folgendes:
Für das **Zweierkomplement** invertieren wir alle Bit (*Einerkomplement*) und *addieren dann 1*.

Möchten wir die **Negativdarstellung einer Zahl**, so stellen wir die Zahl zunächst auf bekannte Weise binär dar. Im Anschluss bilden wir das Zweierkomplement.

Beinhaltet die **Zahldarstellung ein VZ** (ist in der Regel angegeben), so entscheidet diese VZ über unser weiteres Vorgehen.
Wenn $\mathbf{VZ} = \mathbf{0}$, dann ist die dargestellte Zahl positiv. Wir bestimmen also *aus der Zahldarstellung die Dezimalzahl*.
Ist $\mathbf{VZ} = \mathbf{1}$, handelt es sich um eine negative Zahl. Wir bilden also das *Zweierkomplement und bestimmen anschließend die Dezimaldarstellung*.

Beispiel:
Bilden Sie das Zweierkomplement von 100110_2

	0	1	1	0	0	1	*Einerkomplement*
$+$						1	*Addiere 1*
					1		*Übertrag*
	0	1	1	0	1	0	*Zweierkomplement*

Stellen Sie $-\mathbf{9_{10}}$ mit 5 Bit dar.
Bestimme zunächst die Dezimaldarstellung von 9 mit 5 Bit $\Rightarrow 01001_2$

	1	0	1	1	0	*Einerkomplement*
$+$					1	*Addiere 1*
						Übertrag
	1	0	1	1	1	*Zweierkomplement*

Damit folgt: $-9_{10} = 10111_2$

Geben Sie zu folgender Darstellung **mit** Vorzeichenbit an, um welche Zahl es sich handelt.
$1\ 00110_2$: $VZ = 1 \Rightarrow$ negative Zahl.

	0	1	1	0	0	1	*Einerkomplement*
$+$						1	*Addiere 1*
					1		*Übertrag*
	0	1	1	0	1	0	*Zweierkomplement*
		16	8		2		$=$ 26

Damit folgt: $101101_2 = -26_{10}$

Geben Sie zu folgender Darstellung **mit** Vorzeichenbit an, um welche Zahl es sich handelt.
$0\ 01101_2$: $VZ = 0 \Rightarrow$ positive Zahl.

0	0	1	1	0	1	
		8	4		1	$=$ 13

Damit folgt: $001101_2 = 13_{10}$

4.2 Wertebereich

Durch die Reservierung des MSB als Vorzeichenbit (VZ), bleiben uns nur noch 2 Bit für die Zahldarstellung. Wir können also **im positiven** von $0 - 3(2^2 - 1)$ und **im negativen** von $-1 - (-4)(2^2)$.

Zahldarstellung				Zweierkomplement			
Dez	Bin			Bin			Dez
0	0	0	0	0	0	0	0
1	0	0	1	1	1	1	-1
2	0	1	0	1	1	0	-2
3	0	1	1	1	0	1	-3
-4	1	0	0	1	0	0	-4
-3	1	0	1	0	1	1	3
-2	1	1	0	0	1	0	2
-1	1	1	1	0	0	1	1

Bei der Zahldarstellung mit n Bit **mit** Vorzeichenbit gilt für die darstellbaren Zahlen:

(-2^{n-1}) ist die kleinste darstellbare Zahl

$(2^{n-1} - 1)$ ist die größte darstellbare Zahl

Beispiel: Wir haben *4 Bit* **mit** Vorzeichenbit.
Die kleinste darstellbare Zahl ist: $-2^3 = -16$.
Die größte darstellbare Zahl ist: $2^3 - 1 = 15$