Rechnen mit negativen Zahlen

Die Komplementärbildung

ARJ/August 2015

Binärcode

Die 16 verschiedenen
Bitkombinationen
beim 4-Bit-Code.
Es sind hier nur positive
Zahlen vorgesehen!

2^3	2^2	2^1	2^0
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1
1	0	1	0
1	0	1	1
1	1	0	0
1	1	0	1
1	1	1	0
1	1	1	1

0000 = 0
0001 = 1
0010 = 2
0011 = 3
0100 = 4
0101 = 5
0110 = 6
0111 = 7
1000 = 8
1001 = 9
1010 = 10 = A
1011 = 11 = B
1100 = 12 = C
1101 = 13 = D
1110 = 14 = E
1111 = 15 = F

$$\begin{array}{rl} 0100 & (4) \\ +\ 0110 & (6) \\ \hline 1010 & (10) \end{array}$$

$$\begin{array}{rl} 1010 & (10) \\ -\ 0110 & (6) \\ \hline 0100 & (4) \end{array}$$

Binärcode mit negativen Zahlen:

Idee: Das erste Bit (links) stellt das Vorzeichen dar!

Doch funktioniert das?

0000	=	−8
0001	=	−7
0010	=	−6
0011	=	−5
0100	=	−4
0101	=	−3
0110	=	−2
0111	=	−1
1000	=	+0
1001	=	+1
1010	=	+2
1011	=	+3
1100	=	+4
1101	=	+5
1110	=	+6
1111	=	+7

0000	= -8
0001	= -7
0010	= -6
0011	= -5
0100	= -4
0101	= -3
0110	= -2
0111	= -1
1000	= 0
1001	= 1
1010	= 2
1011	= 3
1100	= 4
1101	= 5
1110	= 6
1111	= 7

$$
\begin{array}{rrr}
 & 1001 & (1) \\
+ & 1100 & (4) \\
\hline
 & 0101 & (-3)
\end{array}
$$

$$
\begin{array}{rrr}
 & 1011 & (3) \\
- & 0101 & (-3) \\
\hline
 & 0110 & (-2)
\end{array}
$$

Diese Idee funktioniert offensichtlich nicht richtig!

Binärcode mit negativen Zahlen:

Vorgehen: Die negativen Zahlen ergeben sich durch bitweises Invertieren des entsprechendes Betrags!
Man nennt dies
1-er Komplement!
Doch funktioniert das?

0000	= 0
0001	= 1
0010	= 2
0011	= 3
0100	= 4
0101	= 5
0110	= 6
0111	= 7
1000	= -7
1001	= -6
1010	= -5
1011	= -4
1100	= -3
1101	= -2
1110	= -1
1111	= 0

0000	= 0
0001	= 1
0010	= 2
0011	= 3
0100	= 4
0101	= 5
0110	= 6
0111	= 7
1000	= -7
1001	= -6
1010	= -5
1011	= -4
1100	= -3
1101	= -2
1110	= -1
1111	= 0

$$\begin{array}{rrr} & 0001 & (1) \\ + & 0100 & (4) \\ \hline & 0101 & (5) \end{array}$$

$$\begin{array}{rrr} & 1010 & (-5) \\ - & 1101 & (-2) \\ \hline & 1101 & (-2) \end{array}$$

1-er Komplement

Problem:

Sobald ein Übertrag erzeugt wird, funktioniert es nicht mehr korrekt!

$$\begin{array}{rrr} & 1010 & (-5) \\ - & \underline{1101} & (-2) \\ & 1101 & (-2) \end{array}$$

0000	= 0
0001	= 1
0010	= 2
0011	= 3
0100	= 4
0101	= 5
0110	= 6
0111	= 7
1000	= -7
1001	= -6
1010	= -5
1011	= -4
1100	= -3
1101	= -2
1110	= -1
1111	= 0

```
  1010   (-5)
- 1101   (-2)
  ----
  1101   (-2)

  1101   (-2)
- 0001   (1)
  ----
  1100   (-3)
```

Korrektur wegen Übertrag

```
  1010   (-5)
+ 0111   (7)
  ----
  0001   (1)

  0001   (1)
+ 0001   (1)
  ----
  0010   (2)
```

Korrektur wegen Null-durschschreitung

1-er Komplement

Vorzeichenbehaftete Zahlen mit 1-er Komplement

- Nachteil: Redundanz, weil Null doppelt vorhanden
- Nachteil: Korrektur des Rsultats bei Übertrag

1-er Komplement-Variante verbessern:

Die negativen Zahlen ergeben sich durch bitweises Invertieren des entsprechendes Betrags und Addition von 1!
Man nennt dies
2-er Komplement!
Wie zuverlässig funktioniert nun dies?

0000	=	0
0001	=	1
0010	=	2
0011	=	3
0100	=	4
0101	=	5
0110	=	6
0111	=	7
1000	=	-8
1001	=	-7
1010	=	-6
1011	=	-5
1100	=	-4
1101	=	-3
1110	=	-2
1111	=	-1

2-er Komplement

0000	= 0
0001	= 1
0010	= 2
0011	= 3
0100	= 4
0101	= 5
0110	= 6
0111	= 7
1000	= -8
1001	= -7
1010	= -6
1011	= -5
1100	= -4
1101	= -3
1110	= -2
1111	= -1

$$
\begin{array}{rrr}
 & 1011 & (-5) \\
- & 1110 & (-2) \\
\hline
 & 1101 & (-3)
\end{array}
$$

$$
\begin{array}{rrr}
 & 1011 & (-5) \\
+ & 0111 & (7) \\
\hline
 & 0010 & (2)
\end{array}
$$

Schlussfolgerung zum 2-er Komplement

- Die Grundrechenoperationen funktionieren
- Keine Redundanz wie beim 1-er Komplement
- keine Behandlung des Übertrags wie beim 1-er Komplement