Zahlendarstellung

Binär, Oktal, Hexadezimal, Rechnen, Vorzeichen

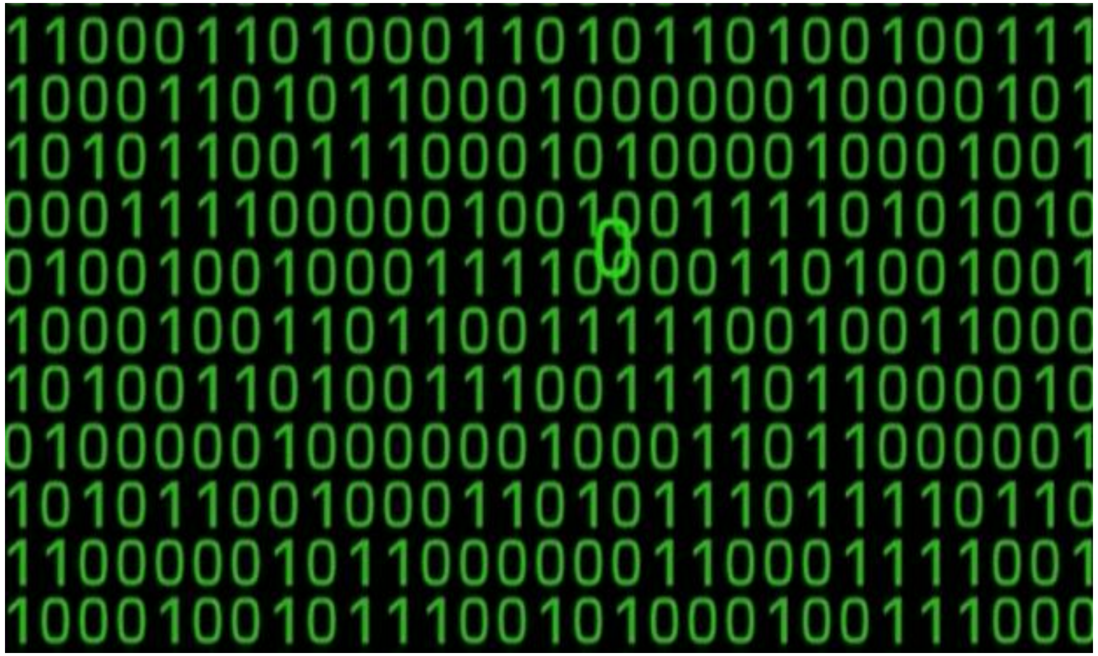
Zahlendarstellung

Natürliche Zahlen können in jeder Basis repräsentiert werden:

$$a_{n-1}a_{n-2} \dots a_1 a_{0(Basis\ B)} = a_{n-1}B^{n-1} + \cdots + a_1 B^1 + a_0 B^0 = \sum_{i=0}^{n-1} a_i B^i$$

- Computer benutzen Basis 2
- Auch 8 und 16 zur Darstellung
- Leichte Darstellung:

-> CPU unterscheidet zwischen 2 Spannungen



Binärzahlen

- Stelle der Ziffer bestimmt Gewicht/Wert

Binärzahl:		1	1	1	0	0	1	0	1
Wert:	256	128	64	32	16	8	4	2	1

11100101 = 128+64+32 + 4 + 1 = ?

Dezimal -> Binär

$167_D \rightarrow 167 / 2 = 83$ Rest 1

$83 / 2 = 41$ Rest 1

$41 / 2 = 20$ Rest 1

$20 / 2 = 10$ Rest 0

$10 / 2 = 5$ Rest 0

$5 / 2 = 2$ Rest 1

$2 / 2 = 1$ Rest 0

$1 / 2 = 0$ Rest 1

$167_D = 10100111_B$

Niederwertigstes Bit
(least significant bit (LSB))

Höchstwertigstes Bit
(most significant bit (MSB))

Oktal und Hexadezimal

- Kürzere Darstellung als Binärzahlen
- -> jede Ziffer geht bis 7 / 15 anstatt 1
- Sehr leichte Konvertierung
- Wie viele Binärstellen brauch man, um 0 bis 7 oder 0 bis 15 darzustellen?

- Konvertierung mit Dezimal: Wie bei Binärzahlen mit Basis 8 / 16

Vorzeichendarstellung

Mehrere Ansätze (einer hat sich durchgesetzt)

- **Vorzeichen-/Betrags-Zahlen** (*Sign-magnitude numbers*)
  - MSB zeigt Vorzeichen (sign) an (0: positiv, 1: negativ).

1-Komplement-Zahlen (*One's complement numbers*)

- Zahl wird durch die Invertierung aller Bits negiert.
- MSB *impliziert* das Vorzeichen.

2-Komplement-Zahlen (*Two's complement numbers*)

MSB hat ein negatives Gewicht (-2^{n-1}).

$b_{n-1}b_{n-2}\ldots b_1b_0$ (binär) = $-b_{n-1}2^{n-1}+b_{n-2}2^{n-2}+\ldots+b_12^1+b_02^0$ (dezimal)

MSB *impliziert* das Vorzeichen.

Overflow

- Deutsch: Überlauf / Bereichsüberschreitung

  Überlauf (***overflow***):
  - Das Ergebnis ist zu groß für ein endliches Computer-Wort
  - Z. B. Addition von zwei n-Bit-Zahlen muss keine n-Bit-Zahl ergeben

- Für Rechnen mit 2k-Zahlen:

$$\boxed{\text{carry in}_{MSB} \neq \text{carry out}_{MSB}}$$