

## Zahlensysteme und Codes:

Computer arbeiten intern mit zwei leicht zu unterscheidenden Zuständen

0 = keine Spannung

1 = Spannung

Wichtigste Aufgabe von Computern: RECHNEN !!

=> Zahlen müssen mit zwei Zuständen repräsentiert (=kodiert) werden

### Unser Zahlensystem: Dezimalsystem

Bsp.:  $37 = 3 \cdot 10 + 7 \cdot 1$

- Ziffern (Koeffizienten)  $\in \{0,1,2,3,4,\dots,9\}$
- jede Ziffer hat eine Wertigkeit ( $10^0, 10^1, 10^2, \dots$ )
  - die zur Stelle passende Zehnerpotenz
- höchstwertigste Stelle steht ganz links
- niederwertige Stellen in absteigender Reihenfolge rechts davon

### Zahlensystem des Computers: Binär-(Dual) System

Bsp.: 1011

- Ziffern (Koeffizienten)  $\in \{0,1\}$  heißen Bits
- jede Ziffer hat eine Wertigkeit ( $2^0, 2^1, 2^2, \dots$ )
  - die zur Ziffer passende Zweierpotenz
- höchstwertigste Stelle steht ganz links
- niederwertige Stellen in absteigen der Reihenfolge rechts davon

	Dezimal		Binär			
Stellenwert	$10^1$	$10^0$	$2^3$	$2^2$	$2^1$	$2^0$
Ziffer	3	7	1	0	1	1

### Umrechnen von Binär- in Dezimalsystem

Jedes Bit (Ziffer) mit ihrer Stellenwertigkeit multiplizieren und aufaddieren.

Bsp.:

$$\begin{aligned}[1011]_2 &= 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 \\ &= 8 + 0 + 2 + 1 \\ &= 11\end{aligned}$$

## Umrechnen von Dezimal- in Binärsystem

gegeben  $a \in \mathbb{N}_0$

- 1) größte Zweierpotenz finden ( $2^i$  mit  $a \geq 2^i$ )
- 2) setze  $a \leftarrow a - 2^i$  ( $2^i$  abziehen von  $a$ )
- 3) Setze die Stelle  $i$  in der Binärdarstellung auf 1
- 4) Schritte 1-3 wiederholen, solange  $a > 0$  ist
- 5) Am Schluss alle nicht auf 1 gesetzten Stellen auf 0 setzen

Bsp.:

$$a = 37$$

$$2^5 = 32 < 37 \Rightarrow a = 37 - 32 = 5, \quad 5. \text{ Stelle} = 1$$

$$2^2 = 4 < 5 \Rightarrow a = 5 - 4 = 1, \quad 2. \text{ Stelle} = 1$$

$$2^0 = 1 = 1 \Rightarrow a = 1 - 1 = 0, \quad 0. \text{ Stelle} = 1$$

$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$
1	0	0	1	0	1

## Alternative zur Umrechnung Dezimal $\rightarrow$ Binär

gegeben  $a \in \mathbb{N}_0$

- 1)  $a \leftarrow \frac{a}{2}$  ( $a$  durch 2 teilen (Ganzzahldivision mit Rest))
- 2) Stelle  $i$  wird der Rest der Ganzzahl
- 3) erhöhe  $i$
- 4) Schritte 1-3 wiederholen, solange  $a > 0$

Bsp.:

$$a = \frac{69}{2} = 34 \quad \text{Rest } 1 \Rightarrow 0. \text{ Stelle} = 1$$

$$a = \frac{34}{2} = 17 \quad \text{Rest } 0 \Rightarrow 1. \text{ Stelle} = 0$$

$$a = \frac{17}{2} = 8 \quad \text{Rest } 1 \Rightarrow 2. \text{ Stelle} = 1$$

$$a = \frac{8}{2} = 4 \quad \text{Rest } 0 \Rightarrow 3. \text{ Stelle} = 0$$

$$a = \frac{4}{2} = 2 \quad \text{Rest } 0 \Rightarrow 4. \text{ Stelle} = 0$$

$$a = \frac{2}{2} = 1 \quad \text{Rest } 0 \Rightarrow 5. \text{ Stelle} = 0$$

$$a = \frac{1}{2} = 0 \quad \text{Rest } 1 \Rightarrow 6. \text{ Stelle} = 1$$

$$\Rightarrow 69 = [1000101]_2$$

### Elementare Rechenoperationen mit Binärzahlen

Addition: - man arbeitet von rechts nach links  
- man erzeugt nach jedem Zwischenschritt eine Ergebnisstelle und eine „Merke“- Stelle

Bsp.:

$$[10011010]_2 = 154$$

$$[110110]_2 = 54$$

$$\begin{array}{r} 10\ 01\ 1010 \\ +\ 00\ 11\ 0110 \\ \hline = 1\ 1\ 01\ 00\ 00 = 16 + 64 + 128 = 208 \end{array}$$

### Vorgehensweise für die Addition von Binärzahlen

Diese Vorgehensweise lässt sich verallgemeinern, indem alle Möglichkeiten für Eingangsbits und Merkerbits in einer Tabelle zusammengefasst werden:

<b>A</b>	<b>B</b>	<b>Merker</b>	<b>Ergebnis</b>	<b>Merker (neu)</b>
0	0	0	0	0
0	0	1	1	0
1	0	0	1	0
1	0	1	0	1
0	1	0	1	0
0	1	1	0	1
1	1	0	0	1
1	1	1	1	1

Diese Art der Beschreibung einer digitalen (bool'schen) Funktion durch eine Tabelle nennt man Funktionstabelle oder Wahrheitstabelle.

### Darstellung negativer Zahlen:

Bei der Darstellung negativer Zahlen in Binärformat ist die Zweierkomplement-Darstellung allgemein üblich.

Umwandlung einer positiven Binärzahl in eine negative Binärzahl in Zweierkomplement-Darstellung.

- 1) - Bitweises Invertieren
- 2) - Addition von 1

Bsp.: -18

$$18 = 2^4 + 2^1 \Rightarrow [10010]_2$$

$$\text{in 8 Bit} = [00010010]_2$$

1) invertieren:  $[11101101]_2$

2) +1:  $+ [00000001]_2$

$$\Rightarrow -18 = [11101110]_2 \text{ (im Zweierkomplement, sonst ist } [11101110] = 238)$$

Das erste Bit ist in dieser Darstellung das Vorzeichenbit. Ist es gesetzt, ist die Binärzahl negativ.

Mit 8 Bit können die Zahlen von

$$[0000.0000]_2 = 0$$

$$\text{bis } [1111.1111]_2 = 255$$

dargestellt werden.

Bei Verwendung des Zweierkomplements wird der Zahlenbereich aufgeteilt:

positiven Zahlen von 0 bis  $[0111.1111]_2 = 127$

negative Zahlen von  $[1000.0000]_2 = 128$  (-128) bis 256 (-1)

Betrachtung von -1

$$1 = 0000.0001$$

$$\text{inv.} = 1111.1110$$

$$+1 = 1111.1111 = -1$$

Betrachtung von -128

$$128 = 1000.0000$$

$$\text{inv.} = 0111.1111$$

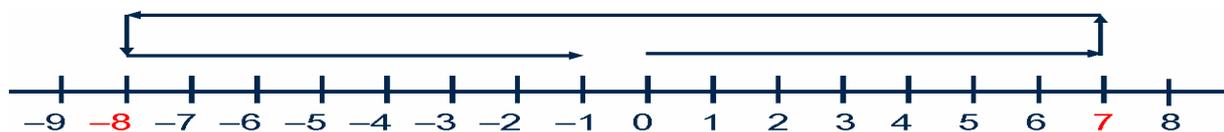
$$+1 = 1000.0000 = -128$$

=> negative Zahlen von

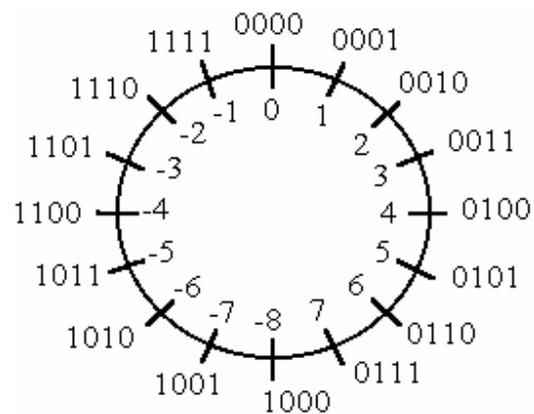
$$[1000.0000]_2 = -128(128)$$

$$\text{bis } [1111.1111]_2 = -1(255)$$

Darstellung auf Zahlengerade



## Zahlenkreis für 4-Bit



Quelle: <http://www.etch.fh-trier.de/>

## Addition von Binärzahlen im Zweierkomplement (Subtraktion von Binärzahlen)

Bsp.:

$$26 - 18 = 26 + (-18)$$

$$\begin{aligned} 26 &= [0001.1010]_2 \\ +(-18) &= [1110.1110]_2 \\ &= [00001000]_2 = 8 \end{aligned}$$