

Digitaltechnik

Wintersemester 2017/2018

2. Vorlesung



TECHNISCHE
UNIVERSITÄT
DARMSTADT





1. Einleitung
2. Darstellung von natürlichen Zahlen
3. Umrechnen zwischen Zahlensystemen
4. Addition von vorzeichenlosen Binärzahlen
5. Vorzeichenbehaftete Binärzahlen
6. Weitere Rechenbeispiele
7. Logikgatter
8. Zusammenfassung

Einleitung



TECHNISCHE
UNIVERSITÄT
DARMSTADT

1001010001100001010001111011011000010010
0001001011011101101001110111110111111100
0101101011010110001010110010101111101111
1011111110110100000000110000111011111110
00001011010011101101011101100000000001111
1010001110101111000110010110010111000111
1000010100011000011000100110000101001110
101110001110111010000100010000100100000
0000101100101001110000100000010101100110
1100010011110000000010100001100111010000
1001011110001000101100101110110000110001
1011001011011100001001001000101000101100
0101111011100011010011100010000110001100
0111011001011110111110111000001001000101
0001000110101011100000000111111000100110
1010001110111101101010110000011110000101



- ▶ Übungsbetrieb angelaufen
 - ▶ bisher 732 Anmeldungen im Moodle
 - ▶ 650 Zuordnungen zu Übungsgruppen
 - ▶ G22 und G23 (Mo 14:25-16:05) kaum nachgefragt
 - ▶ mehr Interessenten für G24 (Fr 11:40-13:20)?

- ▶ Testate erst ab KW 44

- ▶ Übungsausfall am 31.10.2017 (Reformationstag)
 - ▶ betrifft G05, G08, G09
 - ▶ kein Ersatztermin
 - ▶ Lösungsvorschlag und Sprechstunde nutzen
 - ▶ zugehörige Testate (T2) wie geplant in KW 45



- ▶ Beherrschen von Komplexität
 - ▶ Abstraktion
 - ▶ Disziplin
 - ▶ Hierarchie
 - ▶ Modularität
 - ▶ Regularität
- ▶ Digitale Abstraktion
 - ▶ Bits und Bitfolgen
 - ▶ binäre Größenfaktoren (KiBi, MeBi, GiBi, TeBi)
 - ▶ Nibble, Bytes, Wort

Wiederholung:

Zweierpotenzen schnell schätzen



TECHNISCHE
UNIVERSITÄT
DARMSTADT

- ▶ etwa wie viele Farben definiert
 - ▶ 15 bit Real Color?
 - ▶ 24 bit True Color?
 - ▶ 42 bit Deep Color?
- ▶ wie viele Bits nötig zur Repräsentation von
 - ▶ 24 Übungsgruppen
 - ▶ 750 Studierenden (DT)
 - ▶ 26 360 Studierenden (TU Da)

Überblick der heutigen Vorlesung



TECHNISCHE
UNIVERSITÄT
DARMSTADT

► Zahlensysteme: Bitfolgen \leftrightarrow (ganze) Zahlen

- Dezimal-, Binär-, Hexadezimalzahlen
- Darstellung
- Umrechnung
- Addition von Binärzahlen
- Vorzeichenbehaftete Binärzahlen

► Logikgatter: Einfache Boolesche Funktionen

- Wahrheitswertetabellen
- Symbole und Schreibweisen
- Anwendung



Harris 2013
Kap. 1.4 + 1.5
Seite 9 - 22

Darstellung von natürlichen Zahlen



TECHNISCHE
UNIVERSITÄT
DARMSTADT

1101100001001110011110010000101111110000
1110111000100011010110010100111001110010
0001001001100100101111100011010001000001
0010111101101000101110000000000011100010
0011110011110001001011010100011100100111
0100101111000000000011110110000101001000
1111001011111111001001001101000001111111
01000111000001011100100101111101110
1101010110111110000010011101101100111100
10110010100101100101011110011101011001
0110001101001000101110110111101001000001
0111001110100101100100111100011100001011
0011110111010011010100110001111111001010
0111111000100110011010110001101011010110
0111001001000011001101000001011111110111
1100010010000111000101111101100101111011

Definition von Zahlenmengen



TECHNISCHE
UNIVERSITÄT
DARMSTADT

- ▶ natürliche Zahlen $\mathbb{N} = \{0, 1, 2, \dots\}$
- ▶ ganze Zahlen $\mathbb{Z} = \{\dots, -2, -1, 0, 1, 2, \dots\}$
- ▶ rationale Zahlen $\mathbb{Q} = \{\frac{a}{b} : a \in \mathbb{Z} \wedge b \in \mathbb{N} \wedge b \neq 0\}$
- ▶ reelle Zahlen \mathbb{R}
- ▶ komplex, transzendent, algebraisch, ...
- ▶ $|\mathbb{N}| = |\mathbb{Z}| = |\mathbb{Q}| < |\mathbb{R}|$
- ▶ $\infty \notin \mathbb{N}$

Darstellungen von natürlichen Zahlen — Beispiele



TECHNISCHE
UNIVERSITÄT
DARMSTADT

dezimal: 5347 $= 7 \cdot 1 + 4 \cdot 10 + 3 \cdot 100 + 5 \cdot 1000$
 $= 7 \cdot 10^0 + 4 \cdot 10^1 + 3 \cdot 10^2 + 5 \cdot 10^3$
 $=: 5347_{10}$

binär: 1101₂ $= 1 \cdot 2^0 + 0 \cdot 2^1 + 1 \cdot 2^2 + 1 \cdot 2^3$
 $= 1 \cdot 1 + 0 \cdot 2 + 1 \cdot 4 + 1 \cdot 8$
 $= 13_{10}$

hexadezimal: 1F3A₁₆ $= 10 \cdot 16^0 + 3 \cdot 16^1 + 15 \cdot 16^2 + 1 \cdot 16^3$
 $= 10 \cdot 1 + 3 \cdot 16 + 15 \cdot 256 + 1 \cdot 4096$
 $= 7994_{10}$



Definition: vorzeichenloses Stellenwertsystem

Für eine Basis $b \in \mathbb{N} \wedge b \geq 2$ ist $Z_b := \{0, 1, \dots, b-1\}$ die Menge der verfügbaren Ziffern. Die Funktion $u_{b,k}$ bildet eine Ziffernfolge der Breite $k \in \mathbb{N}$ auf eine natürliche Zahl ab:

$$u_{b,k} : (a_{k-1} \dots a_1 a_0) \in Z_b^k \mapsto \sum_{i=0}^{k-1} a_i \cdot b^i \in \mathbb{N}$$



- ▶ polyadisches Zahlensystem
- ▶ niedrigstwertige Stelle (LSB): a_0
- ▶ höchstwertige Stelle (MSB): a_{k-1}
- ▶ kleinste darstellbare Zahl: $\sum_{i=0}^{k-1} 0 \cdot b^i = 0$
- ▶ größte darstellbare Zahl: $\sum_{i=0}^{k-1} (b-1) \cdot b^i = b^k - 1$
- ▶ Anzahl der darstellbaren Werte: $|Z_b^k| = |Z_b|^k = b^k$
- ▶ eineindeutig (bijektiv) auf Wertebereich $\{0, \dots, b^k - 1\}$ für festes k

Häufig verwendete Basen



TECHNISCHE
UNIVERSITÄT
DARMSTADT

	dual/binär	oktal	dezimal	hexadezimal
b	2	8	10	16
Z_b	$\{0, 1\} := \mathbb{B}$	$\{0, \dots, 7\}$	$\{0, \dots, 9\}$	$\{0, \dots, 9, A, B, C, D, E, F\}$
Literale	1101 0011 ₂ 0b11010011	323 ₈ 0o323 0323	211 ₁₀ 0d211 211	D3 ₁₆ 0xD3

► weniger gebräuchlich:

- $b = 20$ wenn man mit Händen *und* Füßen rechnet
- $b = 60$ zur Angabe von Zeit bzw. Längen-/Breitengrade
- $b = 12$ ein „Dutzend“

Umrechnen zwischen Zahlensystemen



TECHNISCHE
UNIVERSITÄT
DARMSTADT

0101111110000101010011000010101110110001
1110110101011010010111001111111010111110
0101010000001011011110100111001111111110
1000010010110100111010010010110010000110
1010110001000010101000000001100011111111
1100011001110111010000011110100111010001
1110101000111011111101000000111111010011
100010110010010011001110001011111100000
01101000111111100101101111000101011110100
1100000001000010100011100100001000001000
0001000000000101101110101111100111101111
1001011011110110000101110011001101100101
1001011101001101001001011000110111100100
1011000001010110100101011100011101111100
1011111011100000011001100111001101111110
0111010010101001011110011000010100001101

Handwerkszeug: Zweiterpotenzen



TECHNISCHE
UNIVERSITÄT
DARMSTADT

$$16^0 = 8^0 = 2^0 = 1$$

$$2^1 = 2$$

$$2^2 = 4$$

$$8^1 = 2^3 = 8$$

$$16^1 = 2^4 = 16$$

$$2^5 = 32$$

$$8^2 = 2^6 = 64$$

$$2^7 = 128$$

$$16^2 = 2^8 = 256$$

$$8^3 = 2^9 = 512$$

$$2^{10} = 1 \text{ Ki}$$

$$2^{11} = 2 \text{ Ki}$$

$$16^3 = 8^4 = 2^{12} = 4 \text{ Ki}$$

$$2^{13} = 8 \text{ Ki}$$

$$2^{14} = 16 \text{ Ki}$$

$$8^5 = 2^{15} = 32 \text{ Ki}$$

$$16^4 = 2^{16} = 64 \text{ Ki}$$

$$16^5 = 2^{20} = 1 \text{ Mi}$$

$$8^{10} = 2^{30} = 1 \text{ Gi}$$

$$16^{10} = 2^{40} = 1 \text{ Ti}$$

Handwerkszeug: Nibble-Werte



TECHNISCHE
UNIVERSITÄT
DARMSTADT

$0000_2 =$	0_{10}	$= 0_{16}$
$0001_2 =$	1_{10}	$= 1_{16}$
$0010_2 =$	2_{10}	$= 2_{16}$
$0011_2 =$	3_{10}	$= 3_{16}$
$0100_2 =$	4_{10}	$= 4_{16}$
$0101_2 =$	5_{10}	$= 5_{16}$
$0110_2 =$	6_{10}	$= 6_{16}$
$0111_2 =$	7_{10}	$= 7_{16}$
$1000_2 =$	8_{10}	$= 8_{16}$
$1001_2 =$	9_{10}	$= 9_{16}$
$1010_2 =$	10_{10}	$= A_{16}$
$1011_2 =$	11_{10}	$= B_{16}$
$1100_2 =$	12_{10}	$= C_{16}$
$1101_2 =$	13_{10}	$= D_{16}$
$1110_2 =$	14_{10}	$= E_{16}$
$1111_2 =$	15_{10}	$= F_{16}$

Binär/Hexadezimal \rightarrow Dezimal



TECHNISCHE
UNIVERSITÄT
DARMSTADT

- ▶ polyadische Abbildung anwenden:
- ▶ $u_{2,5}(1\ 0011_2) = 2^4 + 2^1 + 2^0 = 19_{10}$
- ▶ $u_{16,3}(4AF_{16}) = 4 \cdot 16^2 + 10 \cdot 16^1 + 15 \cdot 16^0 = 1199_{10}$



- ▶ Nibble-weise umwandeln
- ▶ bei least significant bit beginnen
- ▶ führende Nullen weglassen oder ergänzen (je nach geforderter Bitbreite)
- ▶ $11\ 1010\ 0110\ 1000_2 = 3A68_{16}$
- ▶ $7BF_{16} = 111\ 1011\ 1111_2$

Dezimal \rightarrow Binär

(Prinzip auch für größere Basen anwendbar)



TECHNISCHE
UNIVERSITÄT
DARMSTADT

- ▶ Methode 1
(links nach rechts):
maximale Zweierpotenzen
abziehen

$$\begin{aligned} & 53_{10} \\ &= 32 + 21 \\ &= 32 + 16 + 5 \\ &= 32 + 16 + 4 + 1 \\ &= 2^5 + 2^4 + 2^2 + 2^0 \\ &= 11\ 0101_2 \end{aligned}$$

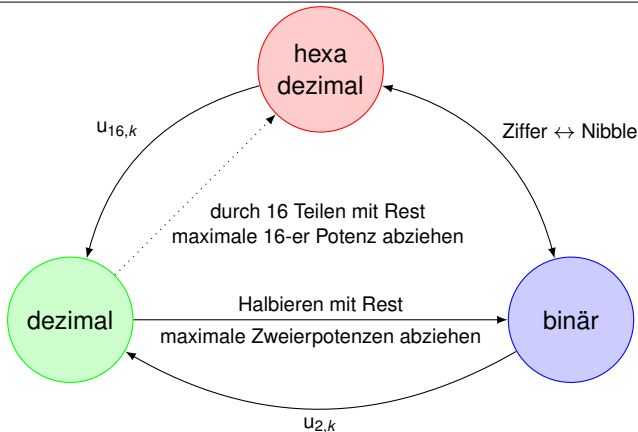
- ▶ Methode 2
(rechts nach links):
Halbieren mit Rest

$$\begin{aligned} & 53_{10} \\ &= 2 \cdot 26 + 1 \\ &= 2 \cdot (2 \cdot 13 + 0) + 1 \\ &= 2 \cdot (2 \cdot (2 \cdot 6 + 1) + 0) + 1 \\ &= 2 \cdot (2 \cdot (2 \cdot (2 \cdot 3 + 0) + 1) + 0) + 1 \\ &= 2 \cdot (2 \cdot (2 \cdot (2 \cdot (2 \cdot 1 + 1) + 0) + 1) + 0) + 1 \end{aligned}$$

Umrechnen zwischen Zahlensystemen



TECHNISCHE
UNIVERSITÄT
DARMSTADT



Zweierpotenzen verinnerlichen!

Addition von vorzeichenlosen Binärzahlen



TECHNISCHE
UNIVERSITÄT
DARMSTADT

0110011000011000101100101010011101101000
1001011000110101011011100001001100110010
0010110000110100000111111111011011111110
00000101001010000101100000010110000011011
10000111001110101110100011100000101110110
010111111111110100100101011000001100100101
111111000101111000000110110110110000001010
011001010011011100010011010100110011101
01011100010000010110110001010100110001010
101010100000101011111111011000000100011000
1010010111001010011110110101011100101110
00110100100010011011011110011010000110000
1001111001000000110000010110010010000010010
0100100111110100000011111110100001000010011
10110110110010111000011011111101001111001
101110000010110100000111001010110111011010

Schriftliche Addition



TECHNISCHE
UNIVERSITÄT
DARMSTADT

► Dezimal:

		1	1		Übertrag
	3	7	3	4	Summand
+	5	1	6	8	Summand
<hr/>					
=	8	9	0	2	Summe

► Binär:

		1	1		Übertrag
	1	0	1	1	Summand
+	0	0	1	1	Summand
<hr/>					
=	1	1	1	0	Summe

Addition mit Überlauf



► Binär:

	1	1			Übertrag	
		1	0	1	1	Summand
+		0	1	1	0	Summand
=	1	0	0	0	1	Summe

Überlauf

- Digitale Systeme arbeiten i.d.R. mit festen Bitbreiten
 - Langzahlarithmetik nur in Software (Bitbreite nur durch verfügbaren Arbeitsspeicher beschränkt)
 - Overflow-flag zum Signalisieren arithmetischer Ausnahmen in Hardware
- Operation (bspw. Addition) läuft über, wenn Ergebnis nicht mit der verfügbaren Bitbreite dargestellt werden kann
- für 4 bit Addierer gilt: $11 + 6 = 1$

Vorzeichenbehaftete Binärzahlen



TECHNISCHE
UNIVERSITÄT
DARMSTADT

001111110011111111110110111111001001110000
1101111110011100101100100011001001011100
1000011101000000001011110010011111111010
0111011110011001010110100000101001010000
0010000110100111010101010101010110010010
1011111001011110010011011000011000110001
001111111010101010100110100010100100011010
111100100111101000001011111110100000000
10101011111010100010101000110001101111000
0110110101101011111101011011010011001000
1101001011010111110000111001010100110011
1001011101100110110101011100001011100001
0000100000111110110000011011011011000000
1110001110000110111110100110000101111001
0001110011101011000010100001101100011110
1011000000001011011101011101110000011011

Darstellungen von ganzen Zahlen — Dezimal



TECHNISCHE
UNIVERSITÄT
DARMSTADT

$$\begin{aligned}-5347_{10} &= (7 \cdot 1 + 4 \cdot 10 + 3 \cdot 100 + 5 \cdot 1000) \cdot -1 \\ &= (7 \cdot 10^0 + 4 \cdot 10^1 + 3 \cdot 10^2 + 5 \cdot 10^3) \cdot (-1)^1\end{aligned}$$

$$\begin{aligned}+5347_{10} &= (7 \cdot 1 + 4 \cdot 10 + 3 \cdot 100 + 5 \cdot 1000) \cdot 1 \\ &= (7 \cdot 10^0 + 4 \cdot 10^1 + 3 \cdot 10^2 + 5 \cdot 10^3) \cdot (-1)^0\end{aligned}$$

► Vorzeichen

- spezielle Ziffer an höchstwertiger Stelle
- kann auch als 0/1 repräsentiert werden

Darstellung von ganzen Zahlen — Verallgemeinerung (Abstraktion)

Definition: Betrag und Vorzeichen

Für eine Basis $b \in \mathbb{N} \wedge b \geq 2$ ist $Z_b := \{0, 1, \dots, b-1\}$ die Menge der verfügbaren Ziffern. Die Funktion $\text{bv}_{b,k}$ bildet eine Ziffernfolge der Breite $k \in \mathbb{N}$ auf eine ganze Zahl ab:

$$\text{bv}_{b,k} : (a_{k-1} \dots a_1 a_0) \in \{0, 1\} \times Z_b^{k-1} \mapsto (-1)^{a_{k-1}} \cdot \sum_{i=0}^{k-2} a_i \cdot b^i \in \mathbb{Z}$$

Ganze Zahlen als Betrag und Vorzeichen



TECHNISCHE
UNIVERSITÄT
DARMSTADT

- ▶ niedrigstwertige Stelle: a_0
- ▶ höchstwertige Stelle: a_{k-1}
- ▶ kleinste darstellbare Zahl: $(-1)^1 \cdot \sum_{i=0}^{k-2} (b-1) \cdot b^i = -(b^{k-1} - 1)$
- ▶ größte darstellbare Zahl: $(-1)^0 \cdot \sum_{i=0}^{k-2} (b-1) \cdot b^i = +(b^{k-1} - 1)$
- ▶ Anzahl der darstellbaren Werte: $2 \cdot b^{k-1} - 1$
- ▶ nicht eindeutig (doppelte Darstellung für Null: ± 0)

Binärdarstellung mit Betrag und Vorzeichen



TECHNISCHE
UNIVERSITÄT
DARMSTADT

► Beispiele

$$\text{bv}_{2,4}(11110_2) = (0 \cdot 2^0 + 1 \cdot 2^1 + 1 \cdot 2^2) \cdot (-1)^1 = -6_{10}$$

$$\text{bv}_{2,4}(01110_2) = (0 \cdot 2^0 + 1 \cdot 2^1 + 1 \cdot 2^2) \cdot (-1)^0 = +6_{10}$$

► *inkompatibel* mit binärer (unsigned) Addition:

$$\begin{array}{rcccccc} & & \textcolor{red}{1} & 1 & 1 & & \\ & & & 1 & 1 & 1 & 0 = -6_{10} \\ + & & & 0 & 1 & 1 & 0 = +6_{10} \\ \hline = & \textcolor{red}{1} & 0 & 1 & 0 & 0 & = 4_{10} \not\equiv \end{array}$$

Definition: Zweierkomplement

Die Funktion s_k bildet eine Bitfolge der Breite $k \in \mathbb{N}$ auf eine ganze Zahl ab:

$$s_k : (a_{k-1} \dots a_1 a_0) \in \mathbb{B}^k \mapsto a_{k-1} \cdot (-2^{k-1}) + \sum_{i=0}^{k-2} a_i \cdot 2^i \in \mathbb{Z}$$

- ▶ auch für Basen $b > 2$ verallgemeinerbar: $s_{b,k}$
- ▶ wird aber heute kaum noch verwendet

Ganze Zahlen als Zweierkomplement



TECHNISCHE
UNIVERSITÄT
DARMSTADT

- ▶ niedrigstwertige Stelle: a_0
- ▶ höchstwertige Stelle: a_{k-1}
- ▶ kleinste darstellbare Zahl: $1 \cdot (-2^{k-1}) + \sum_{i=0}^{k-2} 0 \cdot 2^i = -2^{k-1}$
- ▶ größte darstellbare Zahl: $0 \cdot (-2^{k-1}) + \sum_{i=0}^{k-2} 1 \cdot 2^i = 2^{k-1} - 1$
- ▶ Anzahl der darstellbaren Werte: 2^k
- ▶ eineindeutig (bijektiv) auf Wertebereich $\{-2^{k-1}, \dots, 2^{k-1} - 1\}$ für festes k

Binärdarstellung im Zweierkomplement



TECHNISCHE
UNIVERSITÄT
DARMSTADT

► Beispiele

$$s_4(1010_2) = 0 \cdot 2^0 + 1 \cdot 2^1 + 0 \cdot 2^2 + 1 \cdot -2^3 = -6_{10}$$

$$s_4(0110_2) = 0 \cdot 2^0 + 1 \cdot 2^1 + 1 \cdot 2^2 + 0 \cdot -2^3 = +6_{10}$$

► *kompatibel* mit binärer (unsigned) Addition:

$$\begin{array}{rcccccc} & & \textcolor{red}{1} & 1 & 1 & & \\ & & & 1 & 0 & 1 & 0 = -6_{10} \\ + & & & 0 & 1 & 1 & 0 = +6_{10} \\ \hline = & \textcolor{red}{1} & 0 & 0 & 0 & 0 & = 0_{10} \checkmark \end{array}$$

► *kein Überlauf* bei Addition positiver und negativer Zahl gleicher Breite

Dezimal \rightarrow Zweierkomplement



- ▶ Methode 1 (links nach rechts):
Größtmögliche Zweierpotenzen
abziehen

$$\begin{aligned}-53_{10} &= -64 + 11 \\ &= -64 + 8 + 3 \\ &= -64 + 8 + 2 + 1 \\ &= -2^6 + 2^3 + 2^1 + 2^0 \\ &= 100\ 1011_2\end{aligned}$$

- ▶ Methode 2 (rechts nach links):
Betrag negieren =
Komplement (bitweise $\bar{a} = 1 - a$)
und Inkrement (+1)
(Reihenfolge beachten!)

$$\begin{aligned}-53_{10} &= \overline{53_{10}} + 1 \\ &= \overline{011\ 0101_2} + 1 \\ &= 100\ 1010_2 + 1 \\ &= 100\ 1011_2\end{aligned}$$

- ▶ in beiden Fällen auf korrekte/geforderte Bitbreite achten
- ▶ ggf. müssen führende Null(en) schon für Betragsdarstellung eingefügt werden

Negieren = Komplement und Inkrement



TECHNISCHE
UNIVERSITÄT
DARMSTADT

$$s_k(\overline{a_{k-1} \dots a_0}) = \overline{a_{k-1}} \cdot (-2^{k-1}) + \sum_{i=0}^{k-2} \overline{a_i} \cdot 2^i$$

$$= (1 - a_{k-1}) \cdot (-2^{k-1}) + \sum_{i=0}^{k-2} (1 - a_i) \cdot 2^i$$

$$= \underbrace{\left(1 \cdot (-2^{k-1}) + \sum_{i=0}^{k-2} 1 \cdot 2^i \right)}_{-1} - \underbrace{\left((a_{k-1}) \cdot (-2^{k-1}) + \sum_{i=0}^{k-2} a_i \cdot 2^i \right)}_{s_k(a_{k-1} \dots a_0)}$$

- ▶ notwendig, um unterschiedliche breite Bitfolgen zu addieren
- ▶ *zero extension*:
 - ▶ Auffüllen mit führenden Nullen für vorzeichenlose Darstellung

$$u_{2,k+1}(0a_{k-1} \dots a_0) = 0 \cdot 2^k + \sum_{i=0}^{k-1} a_i \cdot 2^i = u_{2,k}(a_{k-1} \dots a_0)$$

- ▶ *signed extension*:
 - ▶ Auffüllen mit Wert des Vorzeichen-Bits für Zweierkomplement Darstellung

$$\begin{aligned} s_{k+1}(a_{k-1} a_{k-1} \dots a_0) &= a_{k-1} \cdot \underbrace{(-2^k)}_{2 \cdot (-2^{k-1})} + a_{k-1} \cdot 2^{k-1} + \sum_{i=0}^{k-2} a_i \cdot 2^i \\ &= a_{k-1} \cdot \left(-2^{k-1} - 2^{k-1} + 2^{k-1} \right) + \sum_{i=0}^{k-2} a_i \cdot 2^i \\ &= s_k(a_{k-1} \dots a_0) \end{aligned}$$

Bitbreitenerweiterung — Beispiel



TECHNISCHE
UNIVERSITÄT
DARMSTADT

- ▶ -5_{10} im Zweierkomplement von 4 auf 8 Bit erweitern:

$$\begin{aligned} 5_{10} &= 0101_2 \\ \Rightarrow -5_{10} &= \overline{0101_2} + 1 \\ &= 1010_2 + 1 \\ &= 1011_2 \\ &= \mathbf{1111} \ 1011_2 \end{aligned}$$

$$\begin{aligned} \overline{1111 \ 1011_2} + 1 &= 0000 \ 0100_2 + 1 \\ &= 0000 \ 0101_2 \\ &= 5_{10} \end{aligned}$$

Vergleich der binären Zahlendarstellungen für $k=4$



TECHNISCHE
UNIVERSITÄT
DARMSTADT

\mathbb{Z}	Vorzeichenlos: $u_{2,k}$ $\{0, \dots, 2^k - 1\}$	Betrag/Vorzeichen: $bv_{2,k}$ $\{-2^{k-1} + 1, \dots, 2^{k-1} - 1\}$	Zweierkomplement: s_k $\{-2^{k-1}, \dots, 2^{k-1} - 1\}$
15	1111		
14	1110		
13	1101		
12	1100		
11	1011		
10	1010		
9	1001		
8	1000		
7	0111	0111	0111
6	0110	0110	0110
5	0101	0101	0101
4	0100	0100	0100
3	0011	0011	0011
2	0010	0010	0010
1	0001	0001	0001
0	0000	0000 1000	0000
-1		1001	1111
-2		1010	1110
-3		1011	1101
-4		1100	1100
-5		1101	1011
-6		1110	1010
-7		1111	1001
-8			1000

Weitere Rechenbeispiele



TECHNISCHE
UNIVERSITÄT
DARMSTADT

0110000110101100101100000100100011110100
1000101011100010001010101010110000000101
1111101100001001011110100011101011011010
0101001000000101010111100011101100011011
0001000001101110001110010111010000110000
1000101011011111001011111001010110100101
0101010101101111010010110100111010001011
101110110000000100010001011110100111
1000011001010000010001000101010111000110
0111101010100101110000101110010000100100
0101010001011101001010101001001101101111
1101000010010100000110011111011010010011
1110010010011001001001101101000000100111010
1111001101000101000101101011100111010111
1110001000101111111101000111111101010100
1010100001101001111001100010010100011101



▶ $U_{2,7}$

▶ $bv_{2,6}$

▶ $s_{2,10}$

Binär → (Hexa-)Dezimal



TECHNISCHE
UNIVERSITÄT
DARMSTADT

► $u_{2,6}(11\ 0011_2) =$

► $bv_{2,6}(11\ 0011_2) =$

► $s_{2,6}(11\ 0011_2) =$

► $hex(11\ 0011_2) =$

Dezimal \rightarrow Zweierkompliment, Addition



TECHNISCHE
UNIVERSITÄT
DARMSTADT

- ▶ 8 Bit Zweierkomplement von $60_{10} =$
- ▶ 6 Bit Zweierkomplement von $-20_{10} =$
- ▶ binär addieren:
- ▶ Überlauf?



`https://nabla.algo.informatik.tu-darmstadt.de`

Binäre Addition

Addition und Subtraktion von Bits

Binary number: addition

Bereitgestellt von Wikipedia.org



- ☒ Leicht
- ☐ Mittel
- ☐ Schwer

0 0

0 0

0 0

Start

Erstellungsdatum: Freitag, 26. Mai 2017

Autor: Lukas Weber



0100101001111110111001110001001011100001
000100101001100011000111111011010100011
1100101000010011001010111010001000101101
1111010010110101101100011001010110011010
0100001001001000010100101001000001101000
1111011110010010000101011000101101011010
1110100010000101100011011101100000100000
10011010111001010001101011111010101110
1011100110111110000001000100110100010010
1001011001100001001101000100101000010100
1101010111111100001111100010110111111100
1010111101010111110100100011000100000100
01011001111001001100000010011100000110001
1010011100111011001111110000011010100011
0010110110101101110011111011110101100110
0001101110111111001110110111110001000010

Schichtenmodell eines Computers



TECHNISCHE
UNIVERSITÄT
DARMSTADT

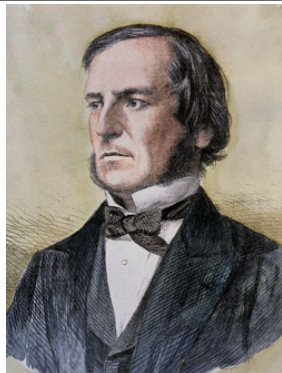
Anwendungs- software	Programme
Betriebs- systeme	Gerätetreiber
Architektur	Befehle Register
Mikro- architektur	Datenpfade Steuerung
Logik	Addierer Speicher
Digital- schaltungen	UND Gatter Inverter
Analog- schaltungen	Verstärker Filter
Bauteile	Transistoren Dioden
Physik	Elektronen

George Boole, 1815 - 1864



TECHNISCHE
UNIVERSITÄT
DARMSTADT

- ▶ in einfachen Verhältnissen geboren
 - ▶ brachte sich selbst Mathematik bei
 - ▶ Professor am Queen's College in Irland
 - ▶ „An Investigation of the Laws of Thought“ (1854)
- ⇒ grundlegende logische Variablen und Operationen



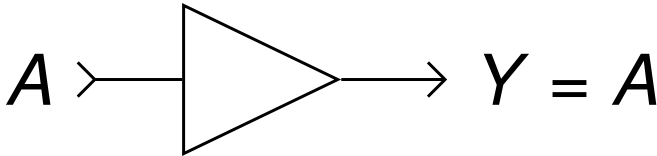


- ▶ verknüpfen binäre Werte: $\mathbb{B}^n \rightarrow \mathbb{B}^k$
- ▶ zunächst $k = 1$
- ▶ Beispiele für
 - ▶ $n = 1$: NOT
 - ▶ $n = 2$: AND, OR, XOR
 - ▶ $n = 3$: MUX
- ▶ Charakterisierung durch Wahrheitwertetabellen

BUF : $\mathbb{B} \rightarrow \mathbb{B}$



TECHNISCHE
UNIVERSITÄT
DARMSTADT

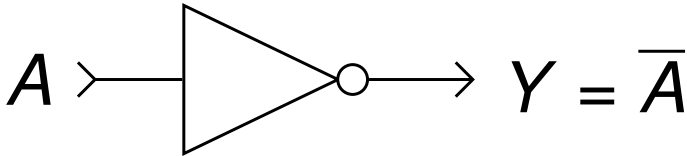


A	Y
0	0
1	1

NOT : $\mathbb{B} \rightarrow \mathbb{B}$



TECHNISCHE
UNIVERSITÄT
DARMSTADT



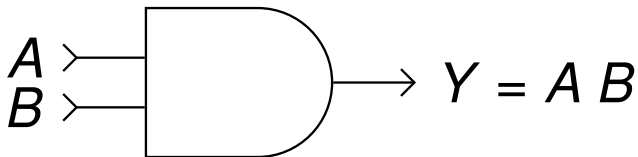
A	Y
0	1
1	0

alternativ: $Y = !A = \sim A$

AND : $\mathbb{B}^2 \rightarrow \mathbb{B}$



TECHNISCHE
UNIVERSITÄT
DARMSTADT



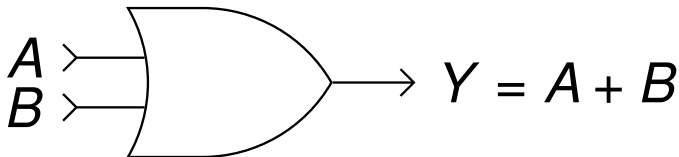
A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

alternativ: $Y = A \cdot B = A \& B = A \wedge B$

OR : $\mathbb{B}^2 \rightarrow \mathbb{B}$



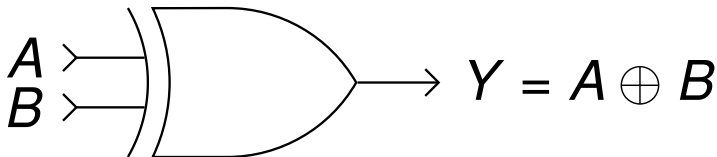
TECHNISCHE
UNIVERSITÄT
DARMSTADT



A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

alternativ: $Y = A|B = A \vee B$

$$\text{XOR} : \mathbb{B}^2 \rightarrow \mathbb{B}$$



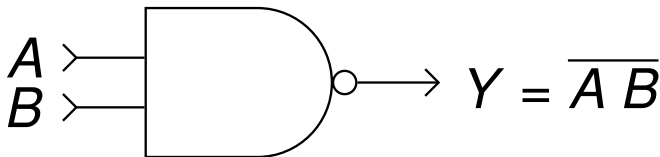
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

alternativ: $Y = A \hat{=} B$

NAND : $\mathbb{B}^2 \rightarrow \mathbb{B}$



TECHNISCHE
UNIVERSITÄT
DARMSTADT

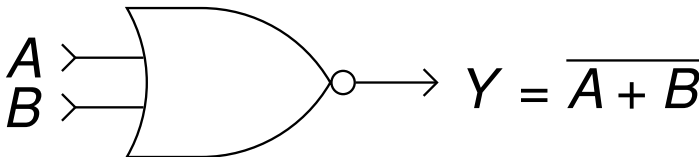


A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

NOR : $\mathbb{B}^2 \rightarrow \mathbb{B}$



TECHNISCHE
UNIVERSITÄT
DARMSTADT

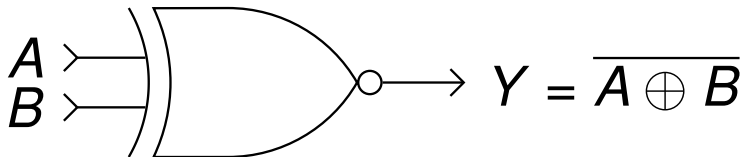


A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

XNOR : $\mathbb{B}^2 \rightarrow \mathbb{B}$



TECHNISCHE
UNIVERSITÄT
DARMSTADT



A	B	Y
0	0	1
0	1	0
1	0	0
1	1	1

entspricht Test auf Gleichheit

Zusammenfassung



TECHNISCHE
UNIVERSITÄT
DARMSTADT

1011110111010100110100011110111011110111
0111111111101010011110001101001100100001
0010011011010000000001110000001000100111
1100011101001110101111011111010100100111
00000000000000001001010111010010100010111
0000100100100011100111101111011111110111
0000101100110010111110001111101101101011
0001110111000011111000001000011011110100
1011011100111011011010101001001001001010
1110000110100101011100101010010001101110
0110111000011110000110101101001000111101
1110110001001111101001110001101110100011
0010101001011110110100000101010101011100
1111101000100001101010011110011110010111
0000010101010111110010010101011110011110
1001110111011110001000010110010011111000



- ▶ Zahlensysteme: Bitfolgen \leftrightarrow (ganze) Zahlen
 - ▶ Dezimal-, Binär-, Hexadezimalzahlen
 - ▶ Darstellung
 - ▶ Umrechnung
 - ▶ Addition von Binärzahlen
 - ▶ Vorzeichenbehaftete Binärzahlen
- ▶ Logikgatter
 - ▶ Darstellung
 - ▶ Wahrheitstabelle
- ▶ nächste Vorlesung behandelt
 - ▶ physikalische Realisierung von Logikgattern