Digitaltechnik
Wintersemester 2017/2018
2. Vorlesung

TECHNISCHE UNIVERSITÄT DARMSTADT

Inhalt

Einleitung

Organisatorisches

TECHNISCHE UNIVERSITÄT DARMSTADT

- 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

Rückblick auf letzte Vorlesung

TECHNISCHE UNIVERSITÄT DARMSTADT

- Beherrschen von Komplexität
  - Abstraktion
  - Diszplin
  - Hierarchie
  - Modularität
  - Regularität

- Digitale Abstraktion
  - Bits und Bitfolgen
  - binäre Größenfaktoren (KiBi, MeBi, GiBi, TeBi)
  - Nibble, Bytes, Wort

Wiederholung: Zweierpotenzen schnell schätzen

- 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 Boolsche Funktionen
  - Wahrheitswertetabellen
  - Symbole und Schreibweisen
  - Anwendung

Harris 2013
Kap. 1.4 + 1.5
Seite 9 - 22

Darstellung von natürlichen Zahlen

TECHNISCHE
UNIVERSITÄT
DARMSTADT

Definition von Zahlenmengen

TECHNISCHE UNIVERSITÄT DARMSTADT

- natürliche Zahlen $\mathbb{N} = \{0, 1, 2, ...\}$
- ganze Zahlen $\mathbb{Z} = \{..., -2, -1, 0, 1, 2, ...\}$
- rationale Zahlen $\mathbb{Q} = \{\frac{a}{b} : a \in \mathbb{Z} \land b \in \mathbb{N} \land b \neq 0\}$
- reele 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

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

binär: 1101_2

$$\begin{aligned} 1101_2 &= 1 \cdot 2^0 + 0 \cdot 2^1 + 1 \cdot 2^2 + 1 \cdot 2^3 \\ &= 1 \cdot 1 + 0 \cdot 2 + 1 \cdot 4 + 1 \cdot 8 \\ &= 13_{10} \end{aligned}$$

hexadezimal: $1F3A_{16}$

$$\begin{aligned} 1F3A_{16} &= 10 \cdot 16^0 + 3 \cdot 16^1 + 15 \cdot 16^2 + 1 \cdot 16^3 \\ &= 10 \cdot 1 + 3 \cdot 16 + 15 \cdot 256 + 1 \cdot 4096 \\ &= 7994_{10} \end{aligned}$$

Darstellungen von natürlichen Zahlen — Verallgemeinerung (Abstraktion)

TECHNISCHE UNIVERSITÄT DARMSTADT

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 $\mathrm{u}_{b,k}$ bildet eine Ziffernfolge der Breite $k \in \mathbb{N}$ auf eine natürliche Zahl ab:

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

Vorzeichenlosen Stellenwertsystem

TECHNISCHE UNIVERSITÄT DARMSTADT

- 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

	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	$1101\ 0011_2$	323_8	211_{10}	$D3_{16}$
	0b11010011	0o323	0d211	0xD3
		0323	211	

- 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

Handwerkszeug: Zweiterpotenzen

TECHNISCHE UNIVERSITÄT DARMSTADT

$$
\begin{array}{lclclcr}
16^0 & = & 8^0 & = & 2^0 & = & 1 \\
 & & & & 2^1 & = & 2 \\
 & & & & 2^2 & = & 4 \\
 & & 8^1 & = & 2^3 & = & 8 \\
16^1 & & & = & 2^4 & = & 16 \\
 & & & & 2^5 & = & 32 \\
 & & 8^2 & = & 2^6 & = & 64 \\
 & & & & 2^7 & = & 128 \\
16^2 & & & = & 2^8 & = & 256 \\
 & & 8^3 & = & 2^9 & = & 512
\end{array}
$$

$$
\begin{array}{lclclcr}
 & & & & 2^{10} & = & 1\,\mathrm{Ki} \\
 & & & & 2^{11} & = & 2\,\mathrm{Ki} \\
16^3 & = & 8^4 & = & 2^{12} & = & 4\,\mathrm{Ki} \\
 & & & & 2^{13} & = & 8\,\mathrm{Ki} \\
 & & & & 2^{14} & = & 16\,\mathrm{Ki} \\
 & & 8^5 & = & 2^{15} & = & 32\,\mathrm{Ki} \\
16^4 & & & = & 2^{16} & = & 64\,\mathrm{Ki} \\
16^5 & & & = & 2^{20} & = & 1\,\mathrm{Mi} \\
 & & 8^{10} & = & 2^{30} & = & 1\,\mathrm{Gi} \\
16^{10} & & & = & 2^{40} & = & 1\,\mathrm{Ti}
\end{array}
$$

Handwerkszeug: Nibble-Werte

TECHNISCHE UNIVERSITÄT DARMSTADT

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

Binär/Hexadezimal $\rightarrow$ Dezimal

TECHNISCHE UNIVERSITÄT DARMSTADT

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

Binär $\leftrightarrow$ Hexadezimal

TECHNISCHE UNIVERSITÄT DARMSTADT

- 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)

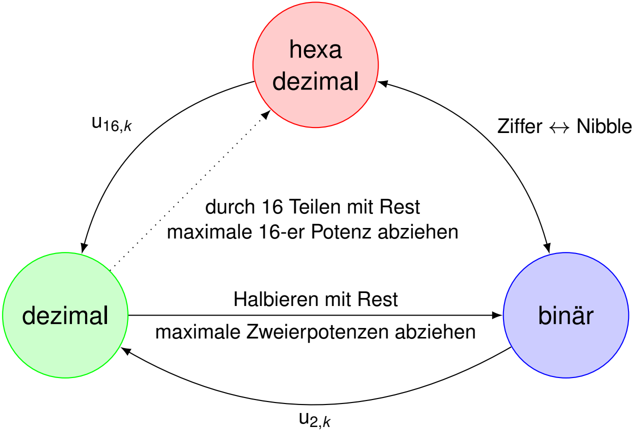
- Methode 1 (links nach rechts): maximale Zweierpotenzen abziehen

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

- 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



Zweierpotenzen verinnerlichen!

Addition von vorzeichenlosen Binärzahlen

TECHNISCHE UNIVERSITÄT DARMSTADT

Schriftliche Addition

- Dezimal:

		1	1		Übertrag
	3	7	3	4	Summand
+	5	1	6	8	Summand
=	8	9	0	2	Summe

- Binär:

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

Addition mit Überlauf

- Binär:

		1	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 Ergenis nicht mit der verfügbaren Bitbreite dargestellt werden kann
- für 4 bit Addierer gilt: $11 + 6 = 1$

Vorzeichenbehaftete Binärzahlen

TECHNISCHE
UNIVERSITÄT
DARMSTADT

Darstellungen von ganzen Zahlen — Dezimal

$$-5347_{10} = (7 \cdot 1 + 4 \cdot 10 + 3 \cdot 100 + 5 \cdot 1000) \cdot -1$$
$$= \left(7 \cdot 10^0 + 4 \cdot 10^1 + 3 \cdot 10^2 + 5 \cdot 10^3\right) \cdot (-1)^1$$

$$+5347_{10} = (7 \cdot 1 + 4 \cdot 10 + 3 \cdot 100 + 5 \cdot 1000) \cdot 1$$
$$= \left(7 \cdot 10^0 + 4 \cdot 10^1 + 3 \cdot 10^2 + 5 \cdot 10^3\right) \cdot (-1)^0$$

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

Darstellung von ganzen Zahlen — Verallgemeinerung (Abstraktion)

TECHNISCHE UNIVERSITÄT DARMSTADT

Definition: Betrag und Vorzeichen

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

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

Ganze Zahlen als Betrag und Vorzeichen

- 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

- Beispiele

$$bv_{2,4}(1110_2) = \left(0 \cdot 2^0 + 1 \cdot 2^1 + 1 \cdot 2^2\right) \cdot (-1)^1 = -6_{10}$$

$$bv_{2,4}(0110_2) = \left(0 \cdot 2^0 + 1 \cdot 2^1 + 1 \cdot 2^2\right) \cdot (-1)^0 = +6_{10}$$

- *inkompatibel* mit binärer (unsigned) Addition:

$$\begin{array}{cccccccl}
 & 1 & 1 & 1 & & & \\
 & & 1 & 1 & 1 & 0 & = -6_{10} \\
+ & & 0 & 1 & 1 & 0 & = +6_{10} \\
\hline
= & 1 & 0 & 1 & 0 & 0 & = 4_{10} \ \lightning
\end{array}$$

Darstellung von ganzen Zahlen — Digitaler „Goldstandard“

TECHNISCHE UNIVERSITÄT DARMSTADT

Definition: Zweierkomplement

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

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

- auch für Basen $b > 2$ verallgemeinerbar: $\mathrm{s}_{b,k}$
- wird aber heute kaum noch verwendet

Ganze Zahlen als Zweierkomplement

- 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 $\{-2^{k-1}, \ldots, 2^{k-1} - 1\}$ 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:

$$\begin{array}{cccccc l}
 & \color{red}{1} & 1 & 1 & & & \\
 & & 1 & 0 & 1 & 0 & = -6_{10} \\
+ & & 0 & 1 & 1 & 0 & = +6_{10} \\
\hline
= & \color{red}{1} & 0 & 0 & 0 & 0 & = 0_{10} \checkmark
\end{array}$$

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

Dezimal → 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 $\overline{a} = 1 - a$) 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

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

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

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

Bitbreitenerweiterung

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

$$\mathrm{u}_{2,k+1}(\textcolor{red}{0}a_{k-1} \ldots a_0) = 0 \cdot 2^k + \sum_{i=0}^{k-1} a_i \cdot 2^i = \mathrm{u}_{2,k}(a_{k-1} \ldots a_0)$$

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

$$\begin{aligned} \mathrm{s}_{k+1}(\textcolor{red}{a_{k-1}}a_{k-1} \ldots a_0) &= a_{k-1} \cdot \underbrace{(-2^k)}_{2\cdot(-2^{k-1})} + a_{k-1} \cdot 2^{k-1} + \sum_{i=0}^{k-2} a_i \cdot 2^i \\ &= a_{k-1} \cdot \left(-2^{k-1} - 2^{k-1} + 2^{k-1}\right) + \sum_{i=0}^{k-2} a_i \cdot 2^i \\ &= \mathrm{s}_k(a_{k-1} \ldots a_0) \end{aligned}$$

Bitbreitenerweiterung — Beispiel

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

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

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

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

Wertebereiche

- $u_{2,7}$
- $bv_{2,6}$
- $s_{2,10}$

Binär $\rightarrow$ (Hexa-)Dezimal

- $u_{2,6}(11\ 0011_2) =$

- $bv_{2,6}(11\ 0011_2) =$

- $s_{2,6}(11\ 0011_2) =$

- $hex(11\ 0011_2) =$

Dezimal $\rightarrow$ Zweierkompliment, Addition

- 8 Bit Zweierkomplement von 60_{10} =

- 6 Bit Zweierkomplement von -20_{10} =

- binär addieren:

- Überlauf?

Nabla-Katalog für Rechnerarchitektur

TECHNISCHE UNIVERSITÄT DARMSTADT

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

Binäre Addition

Addition und Subtraktion von Bits

Binary number: addition

Bereitgestellt von Wikipedia.org

W

- ◉ Leicht 0 0
- ○ Mittel 0 0
- ○ Schwer 0 0

Start

Erstellungsdatum: Freitag, 26. Mai 2017

Autor: Lukas Weber

Logikgatter

Schichtenmodell eines Computers

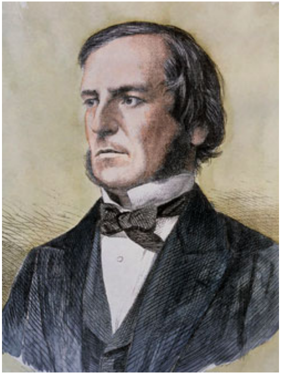
Anwendungs-software	Programme
Betriebs-systeme	Gerätetreiber
Architektur	Befehle Register
Mikro-architektur	Datenpfade Steuerung
Logik	Addierer Speicher
Digital-schaltungen	UND Gatter Inverter
Analog-schaltungen	Verstärker Filter
Bauteile	Transistoren Dioden
Physik	Elektronen

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

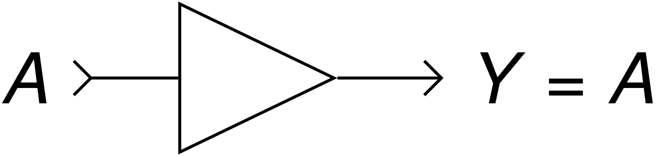


Logische Operationen

TECHNISCHE UNIVERSITÄT DARMSTADT

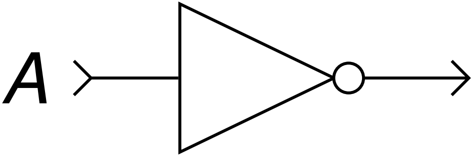
- verknüpfen binäre Werte: $\mathbb{B}^n \to \mathbb{B}^k$
- zunächst $k = 1$
- Beispiele für
  - $n = 1$: NOT
  - $n = 2$: AND, OR, XOR
  - $n = 3$: MUX
- Charakterisierung durch Wahrheitswertetabellen

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



A	Y
0	0
1	1

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

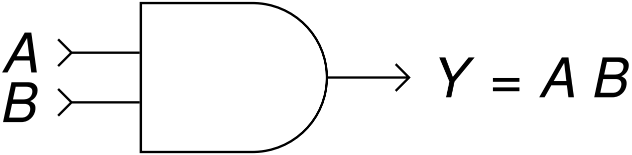


$$Y = \overline{A}$$

A	Y
0	1
1	0

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

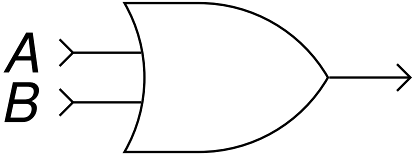
AND : $\mathbb{B}^2 \rightarrow \mathbb{B}$



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

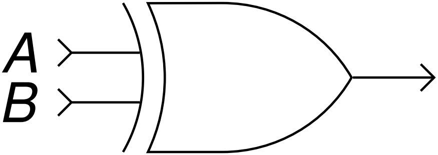


$$Y = A + B$$

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

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

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



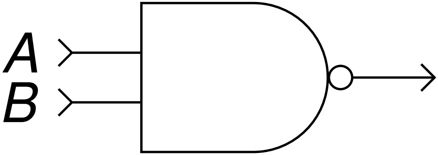
$$Y = A \oplus B$$

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

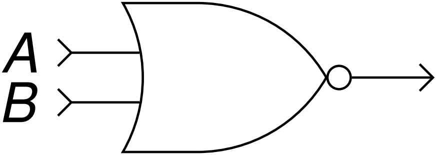


$$Y = \overline{A\,B}$$

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

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

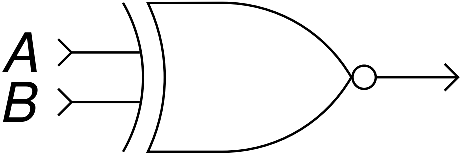
TECHNISCHE UNIVERSITÄT DARMSTADT



$$Y = \overline{A + B}$$

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

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



$$Y = \overline{A \oplus B}$$

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

entspricht Test auf Gleichheit

Zusammenfassung

1 0 1 1 1 1 0 1 1 1 0 1 0 1 0 0 1 1 0 1 0 0 0 1 1 1 1 0 1 1 1 0 1 1 1 1 0 1 1 1
0 1 1 1 1 1 1 1 1 1 1 0 1 0 1 0 0 1 1 1 1 0 0 0 1 1 0 1 0 0 1 1 0 0 1 0 0 0 0 1
0 0 1 0 0 1 1 0 1 1 0 1 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 1 0 0 0 1 0 0 1 1 1
1 1 0 0 0 1 1 1 0 1 0 0 1 1 1 0 1 0 1 1 1 1 0 1 1 1 1 1 0 1 0 1 0 0 1 0 0 1 1 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 1 0 1 1 1 0 1 0 0 1 0 1 0 0 0 1 0 1 1 1
0 0 0 0 1 0 0 1 0 0 1 0 0 0 1 1 1 0 0 1 1 1 1 0 1 1 1 1 0 1 1 1 1 1 1 1 0 1 1 1
0 0 0 0 1 0 1 1 0 0 1 1 0 0 1 0 1 1 1 1 1 0 0 0 1 1 1 1 1 0 1 1 0 1 1 0 1 0 1 1
0 0 0 1 1 1 0 1 1 1 0 0 0 0 1 1 1 1 1 0 0 0 0 0 1 0 0 0 0 1 1 0 1 1 1 1 0 1 0 0
1 0 1 1 0 1 1 1 0 0 1 1 1 0 1 1 0 1 1 0 1 0 1 0 1 0 0 1 0 0 1 0 0 1 0 0 1 0 1 0
1 1 1 0 0 0 0 1 1 0 1 0 0 1 0 1 0 1 1 1 0 0 1 0 1 0 1 0 0 1 0 0 0 1 1 0 1 1 1 0
0 1 1 0 1 1 1 0 0 0 0 1 1 1 1 0 0 0 0 1 1 0 1 0 1 1 0 1 0 0 1 0 0 0 1 1 1 1 0 1
1 1 1 0 1 1 0 0 0 1 0 0 1 1 1 1 1 0 1 0 0 1 1 1 0 0 0 1 1 0 1 1 1 0 1 0 0 0 1 1
0 0 1 0 1 0 1 0 0 1 0 1 1 1 1 0 1 1 0 1 0 0 0 0 0 1 0 1 0 1 0 1 0 1 0 1 1 1 0 0
1 1 1 1 1 0 1 0 0 0 1 0 0 0 0 1 1 0 1 0 1 0 0 1 1 1 1 0 0 1 1 1 1 0 0 1 0 1 1 1
0 0 0 0 0 1 0 1 0 1 0 1 0 1 1 1 1 1 0 0 1 0 0 1 0 1 0 1 0 1 1 1 1 0 0 1 1 1 1 0
1 0 0 1 1 1 0 1 1 1 0 1 1 1 1 0 0 0 1 0 0 0 0 1 0 1 1 0 0 1 0 0 1 1 1 1 1 0 0 0

Zusammenfassung und Ausblick

TECHNISCHE UNIVERSITÄT DARMSTADT

- Zahlensysteme: Bitfolgen $\leftrightarrow$ (ganze) Zahlen
  - Dezimal-, Binär-, Hexadezimalzahlen
  - Darstellung
  - Umrechnung
  - Addition von Binärzahlen
  - Vorzeichenbehaftete Binärzahlen
- Logikgatter
  - Darstellung
  - Wahrheitswertetabellen

- nächste Vorlesung behandelt
  - physikalische Realisierung von Logikgattern