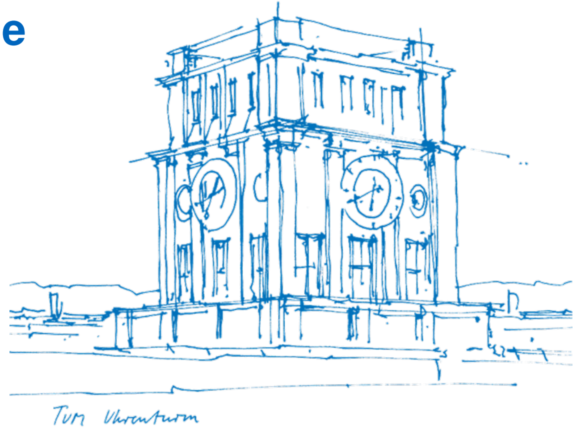
Übung 06: Kombinatorische Schaltungen

Einführung in die Rechnerarchitektur

Niklas Ladurner

School of Computation, Information and Technology
Technische Universität München

21. November 2025

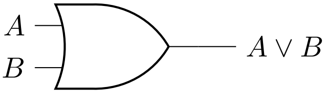


Keine Garantie für die Richtigkeit der Tutorfolien.

Bei Unklarheiten/Unstimmigkeiten haben VL/ZÜ-Folien recht!

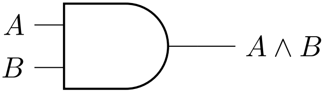
Boolesche Funktionen

OR-Gatter



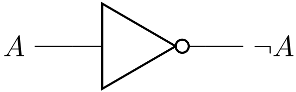
A	B	$A \lor B$
0	0	0
0	1	1
1	0	1
1	1	1

AND-Gatter



A	B	$A \land B$
0	0	0
0	1	0
1	0	0
1	1	1

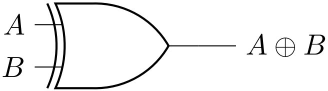
NOT-Gatter



A	$\neg A$
0	1
1	0

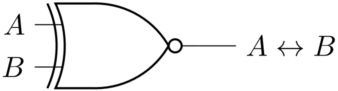
Boolesche Funktionen

XOR-Gatter



A	B	$A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

XNOR-Gatter



A	B	$A \leftrightarrow B$
0	0	1
0	1	0
1	0	0
1	1	1

Definitionen (1)

Funktionale Vollständigkeit

Eine Menge M boolescher Operatoren heißt funktional vollständig, falls alle booleschen Funktionen $f_i \in \mathcal{F}$ als Kombination von Operatoren aus M darstellbar sind. Die Menge $\{\wedge, \neg\}$ ist funktional vollständig.

- Struktur zum Beweis, dass eine Menge M funktional vollständig ist:
  - Zeige, dass $\neg x$ durch Operatoren aus M dargestellt werden kann
  - Zeige, dass $x \wedge y$ durch Operatoren aus M dargestellt werden kann
- Gegenbeweis deutlich schwieriger: Finden einer booleschen Funktion mit $n \geq 1$ Variablen, die nicht dargestellt werden kann (bspw. $\overline{x}$)

Definitionen (2)

Dualität

Zu einer gegebenen boolesche Formel f in Negationsnormalform erhält man den dazugehörigen dualen Ausdruck f^D durch Ersetzung: $\{0 \mapsto 1; 1 \mapsto 0; \wedge \mapsto \vee; \vee \mapsto \wedge\}$. Negationen bleiben davon unberührt!

Beispiele

1. $f = (x \vee y) \wedge 1 \rightsquigarrow f^D = (x \wedge y) \vee 0$
2. $g = (a \wedge b \wedge c) \vee (\overline{a} \wedge (0 \vee \overline{b})) \rightsquigarrow g^D = (a \vee b \vee c) \wedge (\overline{a} \vee (1 \wedge \overline{b}))$

Definitionen (3)

Negationstheorem

Es gilt, dass $\overline{f(x_1, x_2, \ldots, x_n, 0, 1, \wedge, \vee)} \equiv f(\overline{x_1}, \overline{x_2}, \ldots, \overline{x_n}, 1, 0, \vee, \wedge)$.
Achtung: Anders als bei der Dualität werden hier auch die Variablen negiert!

Beispiele

1. $\overline{x \vee y} \equiv (\overline{x} \wedge \overline{y})$ (klassischer De Morgan)
2. $\overline{a \vee (b \wedge 1) \vee (c \vee 0)} \equiv \overline{a} \wedge (\overline{b} \vee 0) \wedge (\overline{c} \wedge 1)$

Gesetze der booleschen Algebra[1]

Bezeichnung	Gesetz	duales Gesetz
Involution	$\overline{\overline{x}} \equiv x$	–
Idempotenz	$x + x \equiv x$	$x \cdot x \equiv x$
Neutralität	$x + 0 \equiv x$	$x \cdot 1 \equiv x$
Extremalgesetz	$x + 1 \equiv 1$	$x \cdot 0 \equiv 0$
Komplementärgesetz	$x + \overline{x} \equiv 1$	$x \cdot \overline{x} \equiv 0$

[1] In ERA werden sowohl die Schreibweisen $\wedge/\vee$ als auch $\cdot/+$ akzeptiert, solange sie einheitlich verwendet werden.

Gesetze der booleschen Algebra[1]

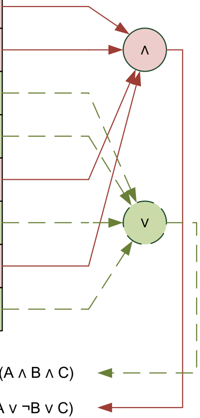
Bezeichnung	Gesetz	duales Gesetz
Involution	$\overline{\overline{x}} \equiv x$	–
Idempotenz	$x + x \equiv x$	$x \cdot x \equiv x$
Neutralität	$x + 0 \equiv x$	$x \cdot 1 \equiv x$
Extremalgesetz	$x + 1 \equiv 1$	$x \cdot 0 \equiv 0$
Komplementärgesetz	$x + \overline{x} \equiv 1$	$x \cdot \overline{x} \equiv 0$
Assoziativität	$x + (y + z) \equiv (x + y) + z$	$x \cdot (y \cdot z) \equiv (x \cdot y) \cdot z$
Kommutativität	$x + y \equiv y + x$	$x \cdot y \equiv y \cdot x$
Distributivität	$x + (y \cdot z) \equiv (x + y) \cdot (x + z)$	$x \cdot (y + z) \equiv (x \cdot y) + (x \cdot z)$
De Morgan	$\overline{x + y} \equiv (\overline{x} \cdot \overline{y})$	$\overline{x \cdot y} \equiv (\overline{x} + \overline{y})$
Absorption	$x + (x \cdot y) \equiv x$	$x \cdot (x + y) \equiv x$

[1] In ERA werden sowohl die Schreibweisen $\wedge/\vee$ als auch $\cdot/+$ akzeptiert, solange sie einheitlich verwendet werden.

Normalformen

- Konjunktive Normalform (OR in den Klammern, AND dazwischen): $(x + y) \cdot (x + \overline{y})$
- Disjunktive Normalform (AND in den Klammern, OR dazwischen): $(x \cdot y) + (x \cdot \overline{y})$
- KNF bei wenigen Nullen im Ergebnis, DNF bei wenigen Einsen im Ergebnis

A	B	C	Ergebnis	Klausel
0	0	0	0	A ∨ B ∨ C
0	0	1	0	A ∨ B ∨ ¬C
0	1	0	1	¬A ∧ B ∧ ¬C
0	1	1	1	¬A ∧ B ∧ C
1	0	0	0	¬A ∨ B ∨ C
1	0	1	1	A ∧ ¬B ∧ C
1	1	0	0	¬A ∨ ¬B ∨ C
1	1	1	1	A ∧ B ∧ C



DNF: (¬A ∧ B ∧ ¬C) ∨ (¬A ∧ B ∧ C) ∨ (A ∧ ¬B ∧ C) ∨ (A ∧ B ∧ C)

KNF: (A ∨ B ∨ C) ∧ (A ∨ B ∨ ¬C) ∧ (¬A ∨ B ∨ C) ∧ (¬A ∨ ¬B ∨ C)

Quelle: Wikipedia

Fragen?

Links

- Zulip: „ERA Tutorium – Mi-1600-3“ bzw. „ERA Tutorium – Fr-1500-1“
- ERA-Moodle-Kurs
- ERA-Artemis-Kurs
- Elektronik-Kompendium zu logischen Grundschaltungen
- GitHub: Digital

Übung 06: Kombinatorische Schaltungen

Einführung in die Rechnerarchitektur

Niklas Ladurner

School of Computation, Information and Technology
Technische Universität München

21. November 2025