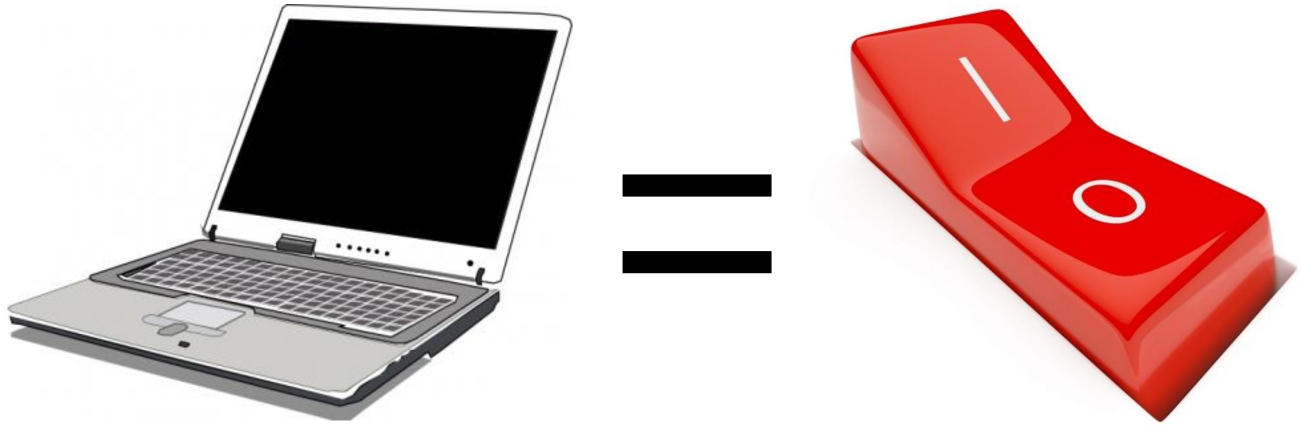
DIGITALE SCHALTUNGEN

Robert Wille (robert.wille@jku.at)
Sebastian Pointner (sebastian.pointner@jku.at)

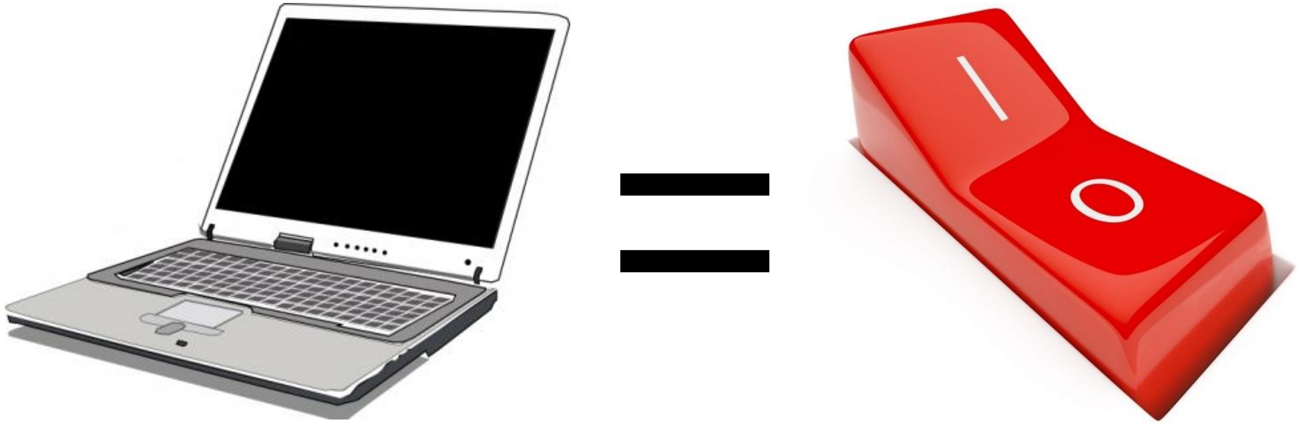
Institut für Integrierte Schaltungen
Abteilung für Schaltkreis- und Systementwurf

INHALT DER VORLESUNG



- **Grundlagen**
  - ☐ Beschreibungen über „0“ und „1“ (Boolesche Algebra)
  - ☐ Beschreibungen von Schaltungen
- **Rechnen**
  - ☐ Darstellung von Zahlen
  - ☐ Digitale Schaltungen für Addition, Subtraktion, Multiplikation
- **Speichern**
  - ☐ Sequentielle Schaltungen
  - ☐ Speicherelemente
- **Steuern**
  - ☐ Endliche Automaten
  - ☐ Synthese von Steuerwerken
- **Entwerfen**
  - ☐ Synthese von allgemeinen Schaltungen
  - ☐ Logikminimierung

INHALT DER VORLESUNG



- **Grundlagen**
  - **Beschreibungen über „0“ und „1“ (Boolesche Algebra)**
  - **Beschreibungen von Schaltungen**
- **Rechnen**
  - Darstellung von Zahlen
  - Digitale Schaltungen für Addition, Subtraktion, Multiplikation
- **Speichern**
  - Sequentielle Schaltungen
  - Speicherelemente
- **Steuern**
  - Endliche Automaten
  - Synthese von Steuerwerken
- **Entwerfen**
  - Synthese von allgemeinen Schaltungen
  - Logikminimierung

GRUNDLAGEN: BOOLESCHE ALGEBRA

Robert Wille (robert.wille@jku.at)
Sebastian Pointner (sebastian.pointner@jku.at)

Institut für Integrierte Schaltungen
Abteilung für Schaltkreis- und Systementwurf

AUSSAGEN

- Aussagen haben einen Wahrheitswert
  - „10 ist eine gerade Zahl" (wahr)
  - „9 ist eine Primzahl" (falsch)
  - Aber: „Freds Schwester" ist keine Aussage
- Aussagen können durch Platzhalter symbolisiert werden (Aussagevariablen)
  - *a* = „10 ist eine gerade Zahl"
  - *b* = „9 ist eine Primzahl"
- Bezeichnungen für Wahrheitswerte
  - Wahr, w, true, 1
  - Falsch, f, false, 0
- Aussagen können miteinander verknüpft werden
  → ***Aussagenlogik***

LOGISCHE VERKNÜPFUNGEN #1

- <u>Einstelliger Operator:</u>
  Negation
  Umkehrung des Wahrheitswertes
  - Symbol: ¬ oder Querstrich über der Aussage
  - Beispiel: $\neg a$, $\overline{b}$

- <u>Zweistellige Operatoren:</u>
  Und, Oder, Äquivalenz, Antivalenz, Implikation

LOGISCHE VERKNÜPFUNGEN #2

- **Und-Verknüpfung**
  - Gesamtaussage ist wahr, wenn beide Teilaussagen wahr sind
  - Symbol: $\wedge$
  - Beispiel: $a \wedge b$

- **Oder-Verknüpfung**
  - Gesamtaussage ist dann wahr, wenn mindestens eine der beiden Teilaussagen wahr ist
  - Symbol: $\vee$
  - Beispiel: $a \vee b$

LOGISCHE VERKNÜPFUNGEN #3

- **Äquivalenz-Verknüpfung**
  - ☐ Gesamtaussage ist dann wahr, wenn beide Aussagen den gleichen Wahrheitswert haben
  - ☐ Symbol: ≡ ⇔
  - ☐ Beispiel: $a \equiv b$

- **Antivalenz**
  - ☐ Gesamtaussage ist wahr, wenn genau eine Teilaussage wahr ist
  - ☐ Symbol: ≠
  - ☐ Beispiel: $a \neq b$

LOGISCHE VERKNÜPFUNGEN #3

- **Negiertes Exklusives Oder (XNOR)**
  - ☐ Gesamtaussage ist dann wahr, wenn beide Aussagen den gleichen Wahrheitswert haben
  - ☐ Symbol: $\equiv$ $\Leftrightarrow$
  - ☐ Beispiel: $a \equiv b$

- **Exklusives Oder (XOR)**
  - ☐ Gesamtaussage ist wahr, wenn genau eine Teilaussage wahr ist
  - ☐ Symbol: $\otimes$
  - ☐ Beispiel: $a \otimes b$

LOGISCHE VERKNÜPFUNGEN #4

- **Implikation**
  - Gesamtaussage ist nur dann falsch, wenn erste Teilaussage wahr und zweite Teilaussage falsch ist
  - Entspricht in etwa der umgangssprachlichen Wenn-dann Formulierung (ist aber exakter, z.B. wenn die erste Teilaussage falsch ist)
  - Symbol: $\Rightarrow$
  - Beispiel: $a \Rightarrow b$

    a = „ich bestehe die Prüfung“
    b = „ich bin glücklich“
    Kann ich glücklich sein ohne die Prüfung zu bestehen? ✓
    Kann ich die Prüfung bestehen ohne glücklich zu sein? ϟ

JKU

WAHRHEITSTABELEN

- Tabellarische Darstellung der Funktion
  - Alle möglichen Eingangskombinationen
  - Resultierende Funktionswerte

		Operator				
a	b	Und	Oder	Äquivalenz	Antivalenz	Implikation
0	0	0	0	1	0	1
0	1	0	1	0	1	1
1	0	0	1	0	1	0
1	1	1	1	1	0	1

REGELN #1

- **Kommutativität**
  - $a \land b \quad \equiv \quad b \land a$
  - Und, Oder, Äquivalenz und Antivalenz sind kommutativ
  - Implikation ist nicht kommutativ

- **Assoziativität**
  - $(a \land b) \land c \quad \equiv \quad a \land (b \land c)$
  - Und, Oder, Äquivalenz und Antivalenz sind assoziativ
  - Implikation ist nicht assoziativ
  - Wichtig bei der Verknüpfung mehrerer Aussagen

REGELN #2

- **Tautologie**
  - Aussage, die immer wahr ist
  - Beispiele:
    - $a \lor \neg a$
    - $(a \Rightarrow b) \lor (b \Rightarrow a)$

- **Kontradiktion**
  - Aussage, die immer falsch ist
  - Beispiel: $a \land \neg a$

BOOLESCHE ALGEBRA #1

- Spezielle Algebra (Komplementärer, distributiver Verband)
- Zahlenmenge (Körper) mit Addition und Multiplikation
  - Zahlenmenge z.B. Wahrheitswerte
  - Addition ≡ Oder-Verknüpfung
  - Multiplikation ≡ Und-Verknüpfung

BOOLESCHE ALGEBRA #2

Definition:

Eine Menge B von Elementen, über der zwei Operationen $+$ und $*$ erklärt sind, ist genau dann eine Boolesche Algebra $(B; +, *)$, wenn für beliebige Elemente $a, b, c \in B=\{0,1\}$ folgende Axiome gelten:

(1) $a + b = b + a$
$a * b = b * a$

Kommutativität

(2) $0 + a = a$
$1 * a = a$

Nullelement 0 bzw. Einselelement 1 bzgl. + bzw. * existiert (neutrale Element)

(3) $(a + b) * c = (a * c) + (b * c)$
$(a * b) + c = (a + c) * (b + c)$

Distributivität einer Operation bezüglich einer anderen

(4) $a + \neg a = 1$
$a * \neg a = 0$

Zu jedem Element $a \in B$ existiert ein komplementäres Element $\neg a \in B$

WEITERE GESETZE

- Assoziativitätsgesetz

- Idempotenz
  - $a + a = a$
  - $a * a = a$

- Absorption
  - $(a + b) *a = a$
  - $(a * b) + a = a$

- De Morgansche Regeln
  - $(\neg a * \neg b) = \neg (a + b)$
  - $(\neg a + \neg b) = \neg (a * b)$

DE MORGANSCHE REGELN

- Folgerung: Und und Oder lassen sich ineinander verwandeln
  - $a + b \;=\; \neg(\neg a * \neg b)$
    Variante (1) negieren
  - $a * b \;=\; \neg(\neg a + \neg b)$
    Variante (2) negieren

- Wird zur algebraischen Vereinfachung von Ausdrücken gebraucht

EINDEUTIGKEIT

- Verschiedene algebraische Darstellungen der gleichen Funktion möglich:
  - $y = b * a + \neg a$
  - $Y = b + \neg a * \neg b$

a	b	y
0	0	1
0	1	1
1	0	0
1	1	1

- Beide Darstellungen sind nicht minimal:
  - $y = b + \neg a$
- Normalformen
- Verfahren zur Optimierung
  - Karnaugh-Veitch
  - Quine-McCluskey
  - Binäre Entscheidungsdiagramme

  → Thema im Teil „Entwerfen“

BOOLESCHE AUSDRÜCKE

- Die Elemente 0 und 1 sind Boolesche Ausdrücke
- Die Symbole/Variablen $x_1, \ldots, x_n$ sind Boolesche Ausdrücke
- Sind g und h Boolesche Ausdrücke, so auch
  die **Disjunktion** $(g+h)$,
  die **Konjunktion** $(g \cdot h)$
  und die **Negation** $(\sim g)$.
- Nichts sonst ist ein Boolescher Ausdruck.

Vereinbarung für das Schreiben

1. Negation ~ bindet stärker als Konjunktion ·
2. Konjunktion · bindet stärker als Disjunktion +

→ Klammern können häufig weggelassen werden,
ohne dass Mehrdeutigkeiten entstehen

GRUNDLAGEN: BESCHREIBUNG VON SCHALTUNGEN

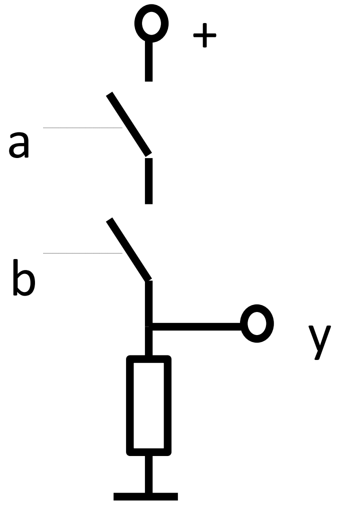
Robert Wille (robert.wille@jku.at)
Sebastian Pointner (sebastian.pointner@jku.at)

Institut für Integrierte Schaltungen
Abteilung für Schaltkreis- und Systementwurf

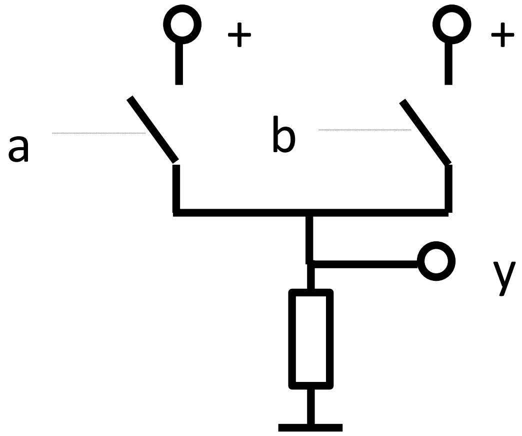
SCHALTERÄQUIVALENZ #1

- Verknüpfungen können durch spannungsgesteuerte Schalter dargestellt werden
- Wahr entspricht einer hohen Spannung

Und-Verknüpfung durch Reihenschaltung



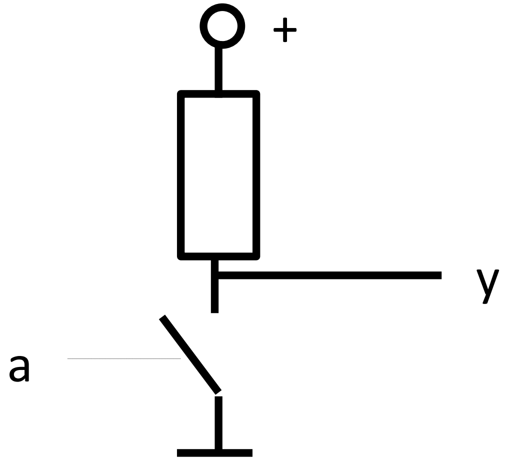
Oder-Verknüpfung durch Parallelschaltung



SCHALTERÄQUIVALENZ #2

Negation

- Realisierung der Negation
  - Bei geöffnetem Schalter muss eine Spannung anliegen
  - Schließen des Schalters muss Spannung auf 0 senken



SCHALTERÄQUIVALENZ #3

- Variante mit Widerstand nicht günstig
- Stattdessen: Schalter, der schließt, wenn keine Spannung anliegt (an Stelle des Widerstands)

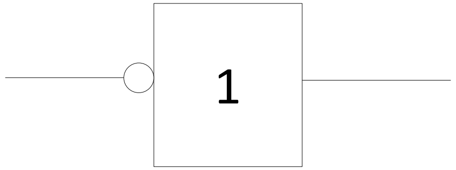
- In der Realität:
  - Feldeffekt-Transistoren als Schalter-Ersatz (funktionieren spannungsgesteuert)
  - Zwei Varianten verfügbar
    - NMOS-Transistor leitet bei hohem Potential
    - PMOS-Transistor leitet bei niedrigem Potential

DIN/IEC-SYMBOLE #1

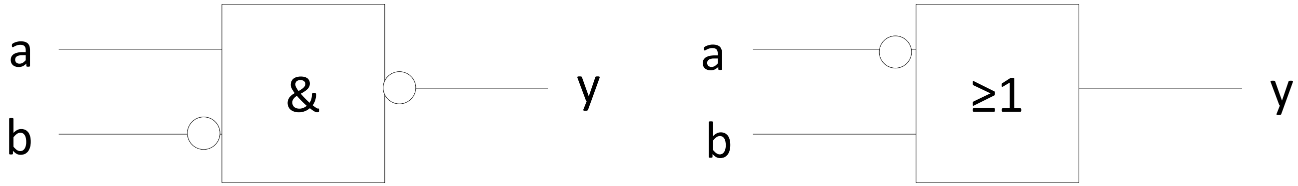
freie Symbole	Schaltsymbole nach DIN 40 700 Teil 14		amerikanische Symbole	logische Darstellung
	seit 1976	bis 1976		
UND	&			$x_1 \wedge \ldots \wedge x_n$
ODER	≥ 1			$x_1 \vee \ldots \vee x_n$
Anti-valenz	= 1			$x_1 \neq \ldots \neq x_n$
NAND	&			$\overline{x_1 \wedge x_2 \wedge \ldots \wedge x_n}$
NOR	≥ 1			$\overline{x_1 \vee x_2 \vee \ldots \vee x_n}$
Negation	1			$\overline{x_1}$

DIN/IEC-SYMBOLE #2

- Negationssymbol darf auch am Eingang stehen



- Darstellungsmöglichkeiten der Implikation



EXOR-REALISIERUNG

- EXOR nicht als Basisfunktion vorhanden

a	b	Exor
0	0	0
0	1	1
1	0	1
1	1	0

- Realisierung mit Hilfe von Und und Oder
- $y = \neg a * b \; + \; a * \neg b$
- Andere Realisierungen möglich (de Morgan)

SCHALTKREIS

- Hier: kombinatorische Schaltkreise
- Gerichteter, zyklenfreier Graph
- Knoten repräsentieren
  - Primäre Eingänge
  - Primäre Ausgänge
  - Gatter (i.d.R. basierend auf vorher festgelegter Gatterbibliothek)
- Kanten repräsentieren
  - Signale zwischen den Gattern bzw. primären Eingängen/Ausgängen
- Gängige Kostenmaße
  - Anzahl der Gatter (Größe)
  - Tiefe, d.h. Zahl der Gatter auf dem längsten Pfad von einem primären Eingang zu einem primären Ausgang (Geschwindigkeit)

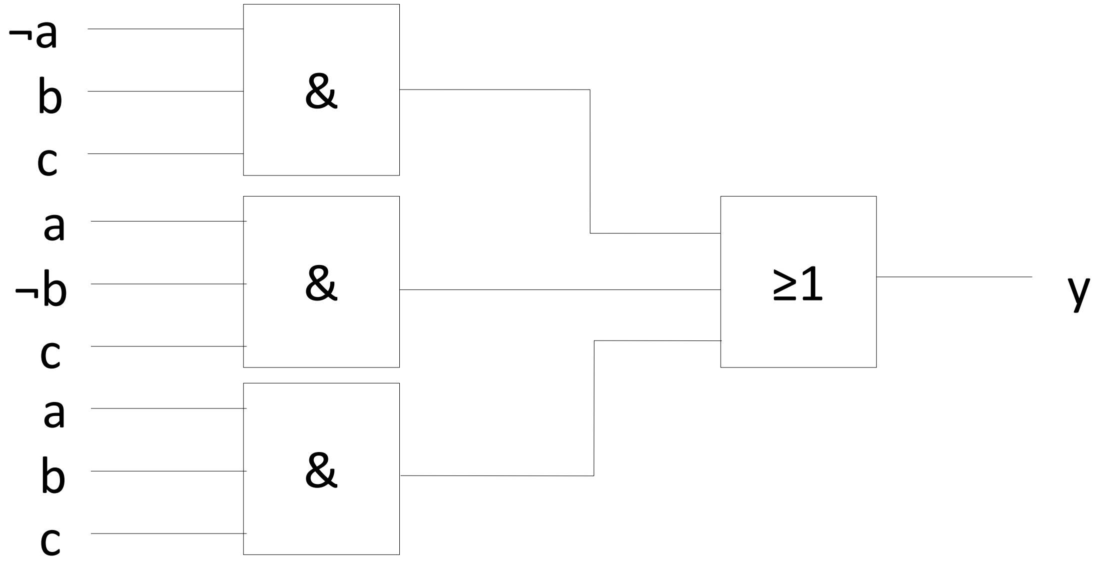
SYNTHESE (SIMPEL)

- Realisierung beliebiger Wahrheitstabellen durch Grundgatter möglich

- Vorgehen:
  - Für jede Zeile mit Ausgabewert 1:
    Und-Gatter mit passender Eingangsbeschaltung
  - Oder-Verknüpfung aller Und-Gatter

SYNTHESE (SIMPEL) – BEISPIEL

a	b	c	y
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1



SYNTHESE (SIMPEL)

- Realisierung beliebiger Wahrheitstabellen durch Grundgatter möglich

- Vorgehen:
  - Für jede Zeile mit Ausgabewert 1:
    Und-Gatter mit passender Eingangsbeschaltung
  - Oder-Verknüpfung aller Und-Gatter

- Funktioniert für alle Tabellen, aber
  - teuer und
  - nicht skalierbar