TECHNISCHE UNIVERSITÄT DRESDEN

VLSI-EDA

Fakultät Informatik, Institut für Technische Informatik, Professur VLSI-Entwurfssysteme, Diagnostik und Architektur

Rechnerstrukturen und -organisation (RSO)

Schaltnetze

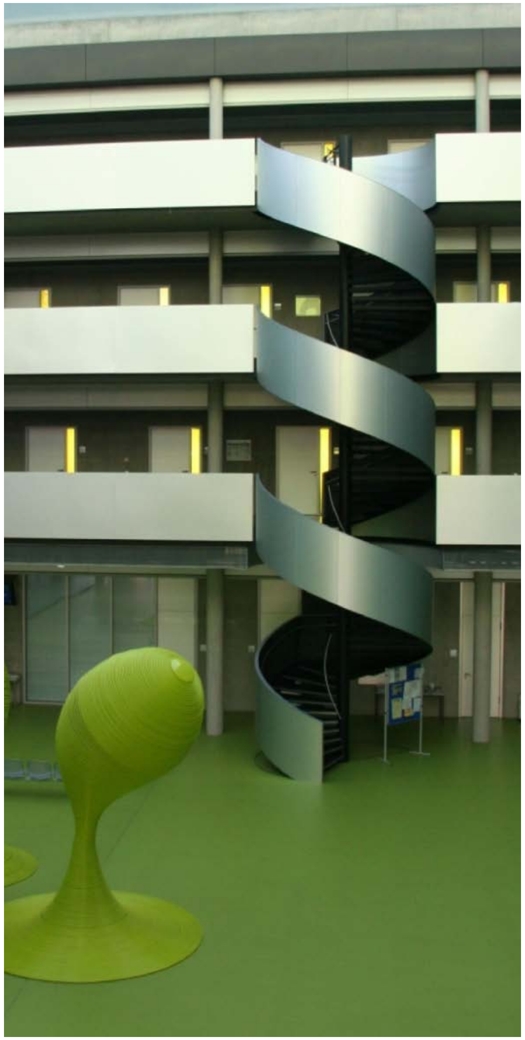
Rainer G. Spallek

TU Dresden, 03.12.2020

Rainer.Spallek@tu-dresden.de

http://vlsi-eda.inf.tu-dresden.de

Gliederung



1 Zielstellung

- Erlangung eines Grundverständnisses für elektrische und elektronische Schalter.
- Schalter und deren Zusammenschaltung zu Schaltnetzen.
- Schaltalgebra und Boolesche Algebra als theoretische Grundlage für Schaltnetze.
- Aufbau und Funktion elektronischer Verknüpfungsglieder.
- Analyse und Synthese von komplexen Schaltnetzen. Vereinfachung und Umwandlung von Schaltnetzen.
- Übersicht über die technische Realisierung von Schaltnetzen.
- Kennenlernen von Standardschaltnetzen und deren Anwendungen in der Computertechnik.

2 Schalter, Netzwerke von Schaltern

Schalter als binäre Elemente

Schalter realisieren eine eindeutige Trennung zwischen genau zwei definierten Zuständen: Schalter **entweder** offen **oder** geschlossen.

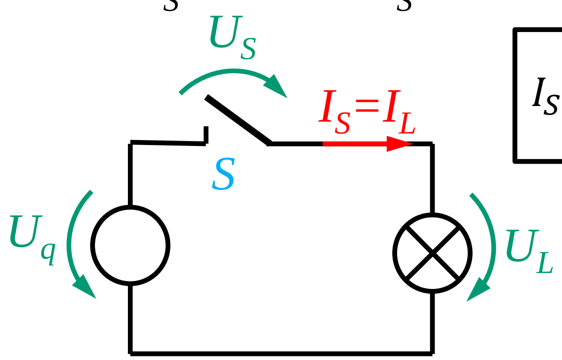
Schalter realisieren eine binäre Entscheidung:
→ Grundlage für logische Verknüpfungen.
→ Zentrale Bedeutung der Schalter in der Computertechnik.

Schalter sind für die verschiedensten physischen Größen realisierbar: elektrisch, elektronisch, pneumatisch, hydraulisch, mechanisch, thermisch, optisch, . . .

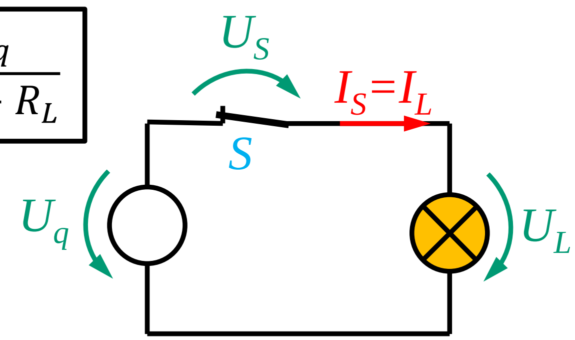
In der Computertechnik dominieren derzeit eindeutig elektronische Schalter in Form von elektronischen Verknüpfungsgliedern.

Schalter in elektrischer Schaltung

Für folgende elektrischer Schaltung mit $U_q > 0$, idealer Schalter mit Schalter-widerstand $R_S = 0$ bzw. $R_S = \infty$ und einem Lampenwiderstand $0 < R_L < \infty$ gilt:



$$I_S = I_L = \frac{U_q}{R_S + R_L}$$



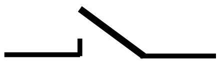
S – offen → Lampe aus
$U_L = 0,\ I_L = 0,\ R_S = \infty$
Schalter S: $U_S > 0,\ I_S = 0$

S – geschlossen → Lampe an
$U_L = U_q,\ I_L > 0\ ,\ R_S = 0$
Schalter S: $U_S = 0,\ I_S > 0$

Definition Schalterzustand (binär):

Schalter offen:	$S = 0$	keine (offene) Verbindung $R_S = \infty$
Schalter geschlossen:	$S = 1$	leitende (geschlossene) Verbindung $R_S = 0$

Schalterzustand, Schaltvariable, Schaltfunktion



elektrisch

S_x S_x

— 0 — **Schalterzustand** — 1 —

keine Verbindung ← $S_x=0$ **S_x** $S_x=1$ → leitende Verbindung

high-aktiv (Schließer) low-aktiv (Öffner)

Schaltvariable

(Schalterbetätigung)

x

$S_x(x) = 1$ für $x = 1$ $x = 1$ → Schalter wird betätigt $S_x(x) = 1$ für $x = 0$

$S_x(x) = 0$ für $x = 0$ $x = 0$ → Schalter nicht betätigt $S_x(x) = 0$ für $x = 1$

Schaltfunktion

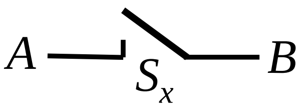
— x — (nach Betätigung mit x) — $\overline{x}$ —

$S_x(x) = x$ **$S_x(x)$** $S_x(x) = \overline{x}$

$S_x(x), x, \overline{x} \in \{0,1\}$ x – Schaltvariable für Schalter S_x bzw. Schaltfunktion $S_x(x)$

Schaltfunktion: Einfache Schalter

$S_{AB} = 1$ → leitende Verbindung zwischen A und B → $S_{AB} = 0$ keine Verbindung



A —— x —— B

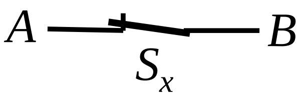
$S_{AB}(x = 1) = 1$
$S_{AB}(x = 0) = 0$

S_{AB}	x
0	0
1	1

→ $S_{AB}(x) = x$

→ **Identität**

$S_{AB}(x) = 1$ gilt dann und nur dann, wenn $x = 1$, sonst $S_{AB}(x) = 0$



A —— $\overline{x}$ —— B

$S_{AB}(x = 1) = 0$
$S_{AB}(x = 0) = 1$

S_{AB}	x
0	1
1	0

→ $S_{AB}(x) = \overline{x}$

→ **Negation (NICHT)**

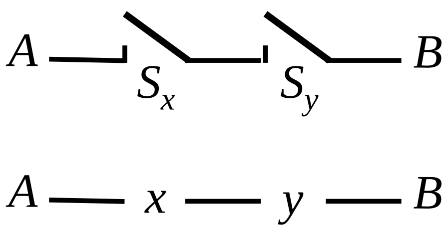
$S_{AB}(x) = 1$ gilt dann und nur dann, wenn $x = 0$, sonst $S_{AB}(x) = 0$

x – Schaltvariable zu Schalter S_x y – Schaltvariable zu Schalter S_y
S_{AB} – Schalterfunktion zwischen den Punkten A und B

Schaltfunktion: Reihenschaltung von Schaltern

$S_{AB} = 1$ → leitende Verbindung zwischen A und B

$S_{AB} = 0$ → keine Verbindung zwischen A und B



$S_{AB}(x=0, y=0) = 0$
$S_{AB}(x=0, y=1) = 0$
$S_{AB}(x=1, y=0) = 0$
$S_{AB}(x=1, y=1) = 1$

S_{AB}	x	y
0	0	0
0	0	1
0	1	0
1	1	1

→ $S_{AB}(x,y) = x \wedge y$

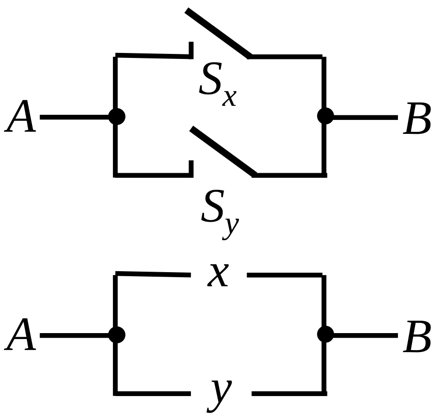
→ **Konjunktion (UND)**

$S_{AB}(x,y) = 1$ gilt dann und nur dann, wenn $x = 1$ **und** $y = 1$ sonst $S_{AB}(x,y) = 0$

Schaltfunktion: Parallelschaltung von Schaltern

$S_{AB} = 1$ → leitende Verbindung zwischen A und B

$S_{AB} = 0$ → keine Verbindung zwischen A und B



$S_{AB}(x{=}0, y{=}0) = 0$
$S_{AB}(x{=}0, y{=}1) = 1$
$S_{AB}(x{=}1, y{=}0) = 1$
$S_{AB}(x{=}1, y{=}1) = 1$

S_{AB}	x	y
0	0	0
1	0	1
1	1	0
1	1	1

→ $S_{AB}(x,y) = x \vee y$

→ **Disjunktion (ODER)**

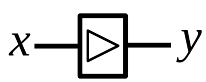
$S_{AB}(x,y) = 1$ gilt dann und nur dann, wenn $x = 1$ **oder und** $y = 1$ sonst $S_{AB}(x,y) = 0$

Boolesche Verknüpfungen mit Schaltern

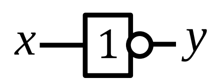
Bezeichnung: Identität | NOT | AND | OR

Schaltsymbol

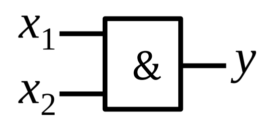
Identität:



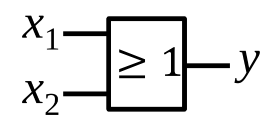
NOT:



AND:



OR:



Wertetabelle

Identität:

x	y
0	0
1	1

NOT:

x	y
0	1
1	0

AND:

x_1	x_2	y
0	0	0
0	1	0
1	0	0
1	1	1

OR:

x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	1

Schaltfunktion

Identität: $y(x) = x$

NOT: $y(x) = \overline{x}$

AND: $y(x_1,x_2) = x_1 \wedge x_2$

OR: $y(x_1,x_2) = x_1 \vee x_2$

3 Elektronische Verknüpfungsglieder

Realisierung gesteuerter Schalter durch elektronische Bauelemente.

Als Bauelemente werden überwiegend Halbleiterbauelemente verwendet:

- Halbleiter-Dioden (pn-Dioden),
- BJT (**B**ipolar **J**unktion **T**ransistor),
- MOS-FET (**M**etall **O**xide **S**emiconductor - **F**ield **E**ffect **T**ransistor).

Transistor: Trans - Res**istor** (steuerbarer Widerstand).
Der Schalter kann ebenfalls als steuerbarer Widerstand verstanden werden $R_s=0 \leftrightarrow R_s=\infty$

Vorteile elektronischer Schalter

- Keine mechanisch beweglichen Teile → hohe Schaltgeschwindigkeiten, hohe Zuverlässigkeit (keine Kontakte) und geringe Ansteuerleistung.
- Geringe Abmaße (*nm* Bereich), Höchstintegration im Festkörper.
- Gleichzeitige Integration extrem vieler Schalter → hohe Ausfallsicherheit

CMOS Schaltungstechnik (Complementary MOS)

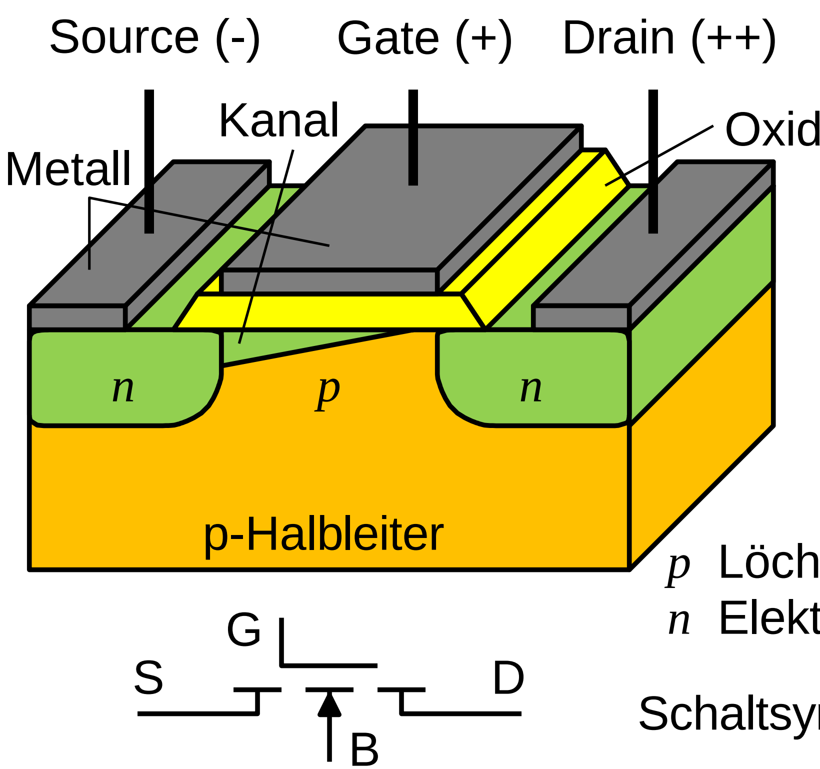
Die CMOS Schaltungstechnik wird in einem Halbleitertechnologieprozess, typischerweise auf Siliziumbasis, hergestellt. Im CMOS Halbleiterprozess werden im Gegensatz zu anderen Halbleiterprozessen zwei komplementäre Schalter (Öffner und Schließer) realisiert → CMOS Schaltungstechnik.

Vorteile der CMOS-Technik (Comlementary MOS):

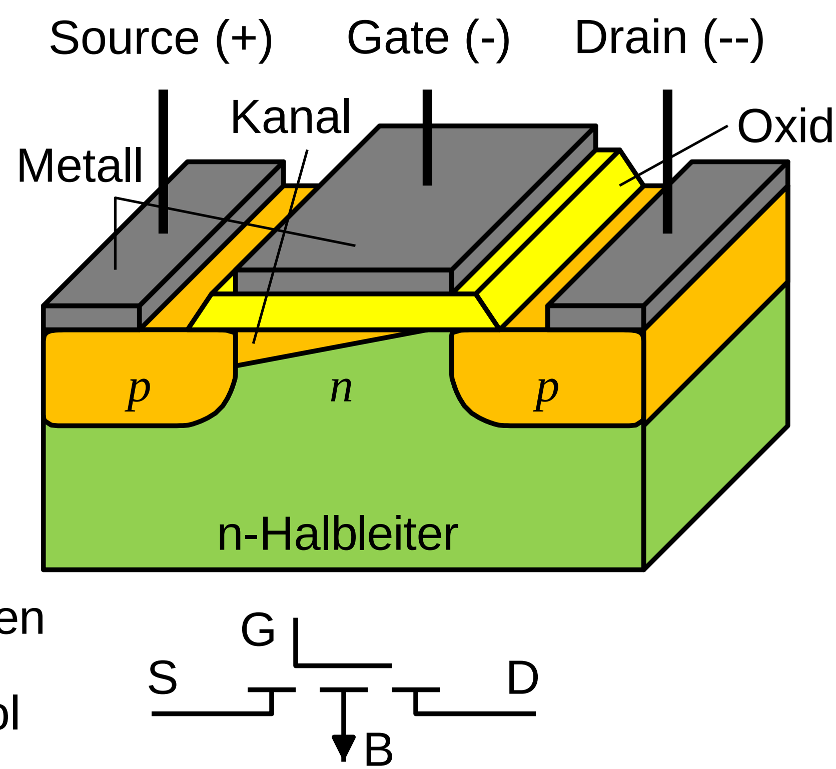
- Kein statischer Querstrom und kein statischer Ansteuerstrom und somit keine statische Verlustleistung.
- Symmetrisches Schaltverhalten, kurze Schaltzeiten, ideale statische Pegel.
- Realisierbarkeit hoher Zuverlässigkeit, hoher Ausfallsicherheit und geringer Strahlungsempfindlichkeit.
- Geringe Abmaße (*nm* Bereich) und geringe Verlustleistung und damit sehr gut geeignet für die Höchstintegration in Silizium (>10^9 Bauelemente/Chip).
- Für analoge, digitale und Mixed Signal Schaltungstechnik geeignet.

Aufbau und Wirkungsweise von MOS-FET

Schnitt durch einen n- und p-Kanal MOS-FET (Enhancement Type) Source (-)



n-Kanal MOS-FET (NMOS)



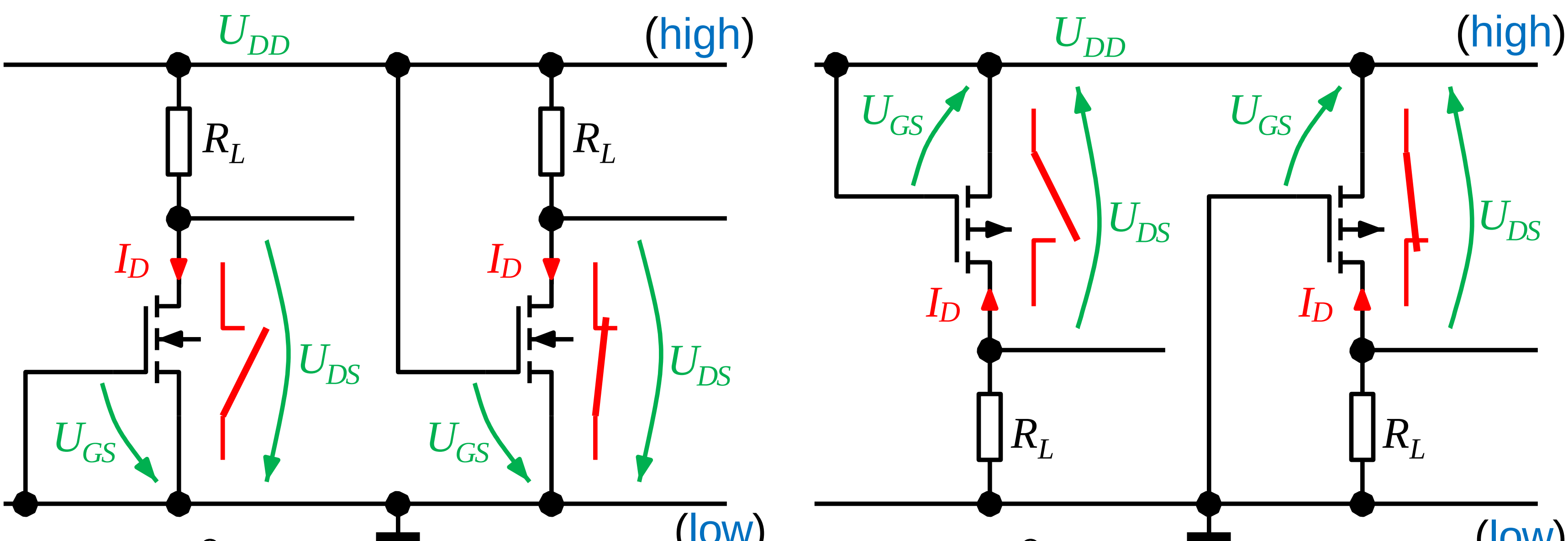
p-Kanal MOS-FET (PMOS)

p Löcher
n Elektronen

Schaltsymbol

Der MOS-FET als elektronischer Schalter

Sperrbereich : $|U_{GS}| < |U_{th}|$ $|I_D| = 0$ (U_{th} - Schwellspannung)
Durchlaßbereich : $|U_{GS}| \geq |U_{th}|$ $|I_D| > 0$ (aktiver Bereich)



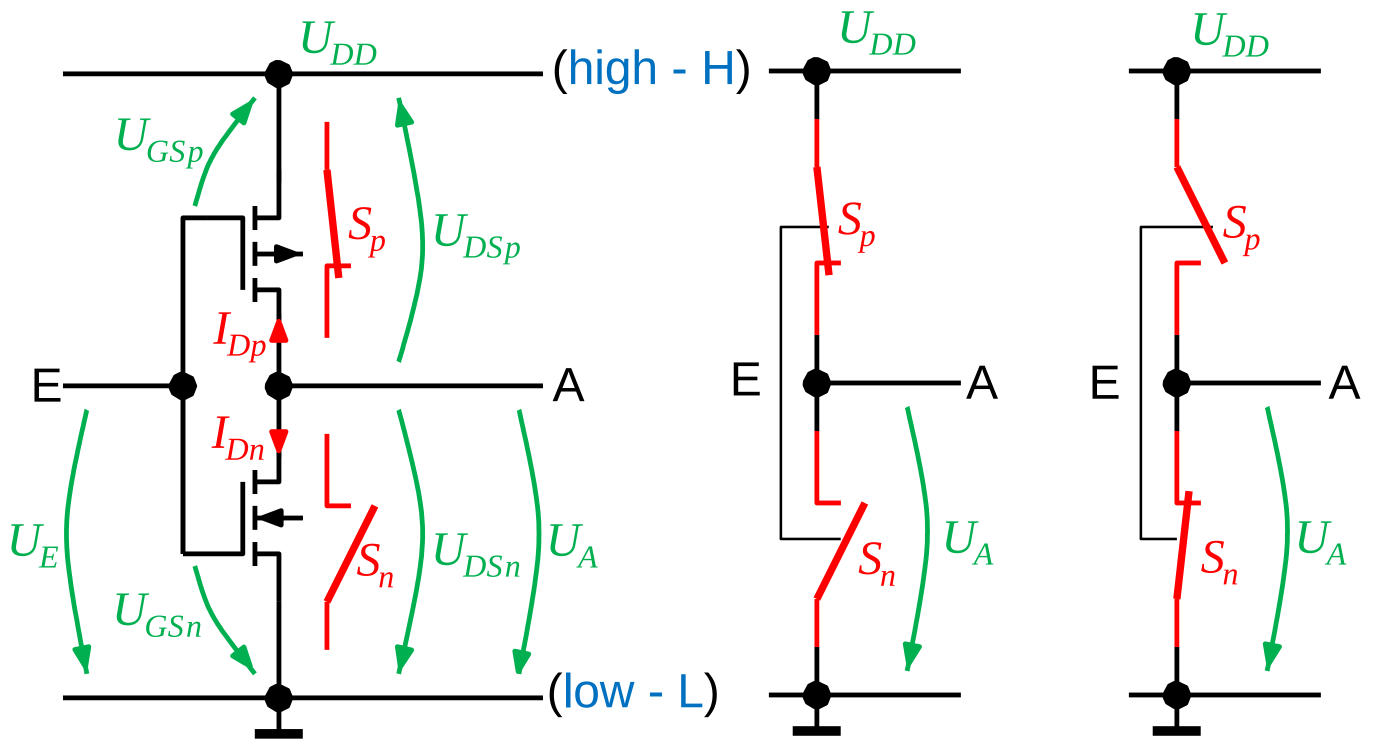
$U_{GS} = 0$
$U_{DS} = U_{DD}$
$I_D = 0$

$U_{GS} = U_{DD}$
$U_{DS} = 0$
$I_D > 0$

$U_{GS} = 0$
$U_{DS} = -U_{DD}$
$I_D = 0$

$U_{GS} = -U_{DD}$
$U_{DS} = 0$
$I_D < 0$

CMOS-NOT mit komplementären Schaltern



$U_{GSn} = U_E$, $U_{DSp} = U_E - U_{DD}$

$U_E = U_{DSn} = U_{DD} + U_{DSp}$

$I_{dn} = -I_{Dp}$

$U_E=0 \rightarrow U_A=U_{DD}$ $U_E=U_{DD} \rightarrow U_A=0$

Spannungen

U_E	U_A
0	U_{DD}
U_{DD}	0

Pegel

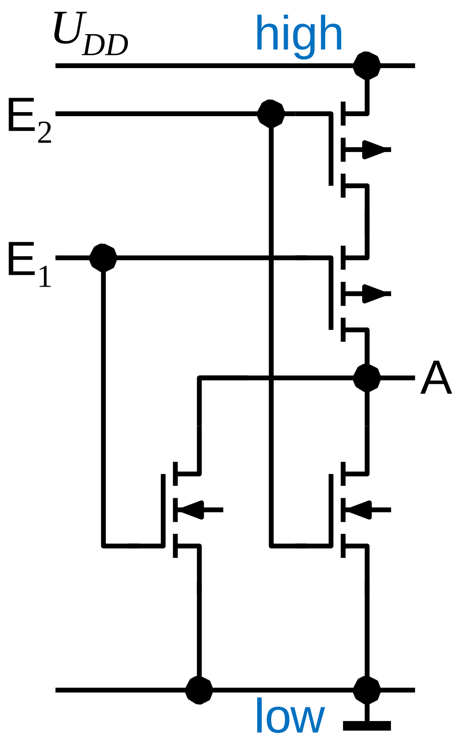
E	A
L	H
H	L

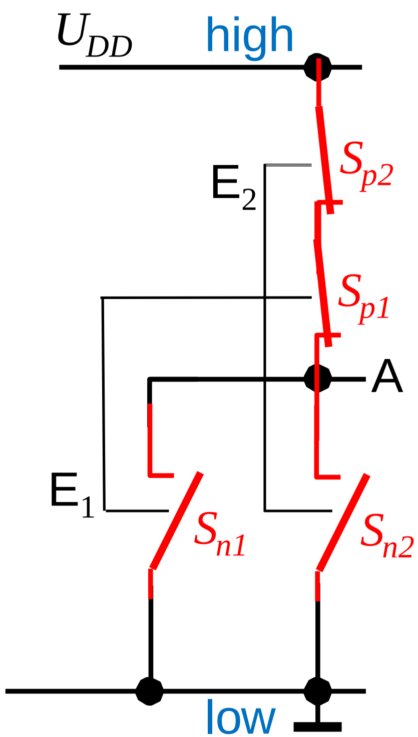
Logik

E	A
0	1
1	0

Schaltsymbol E—[1]o— $A=\overline{E}$ **NOT**

CMOS-NOR mit komplementären Schaltern





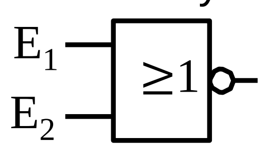
Pegel

E_1	E_2	A
L	L	H
L	H	L
H	L	L
H	H	L

Logik

E_1	E_2	A
0	0	1
0	1	0
1	0	0
1	1	0

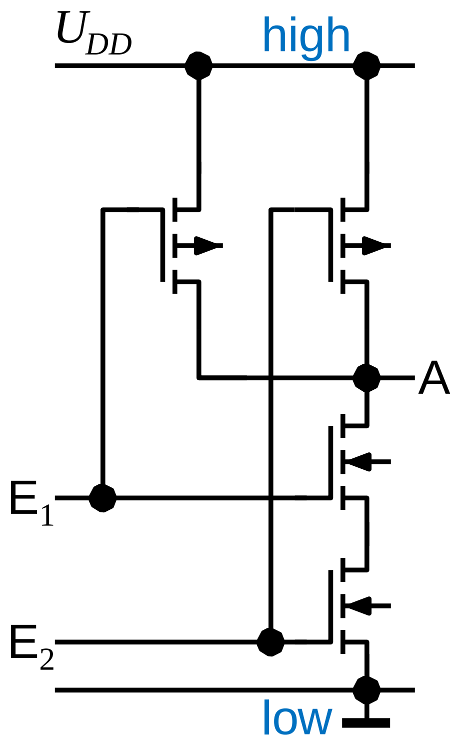
Schaltsymbol

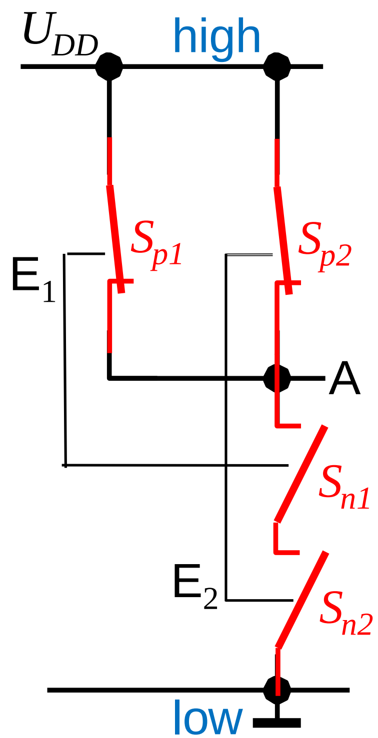


NOR

$A=\overline{E_1 \vee E_2}$

CMOS –NAND mit komplementären Schaltern





Pegel

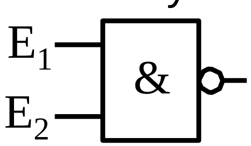
E_1	E_2	A
L	L	H
L	H	H
H	L	H
H	H	L

Logik

E_1	E_2	A
0	0	1
0	1	1
1	0	1
1	1	0

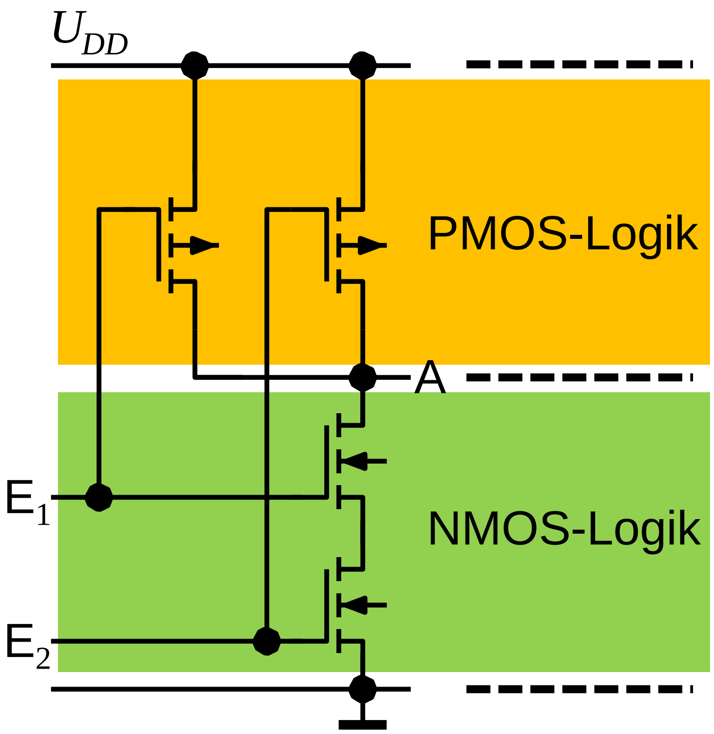
Schaltsymbol

NAND



$A = \overline{E_1 \wedge E_2}$

CMOS Schaltungen Analyse, Entwurf



Die Trennlinie zwischen PMOS- und NMOS-Logik bildet der Ausgang A.

PMOS-Logik		NMOS-Logik
Parallelschaltung	←→	Reihenschaltung
Reihenschaltung	←→	Parallelschaltung

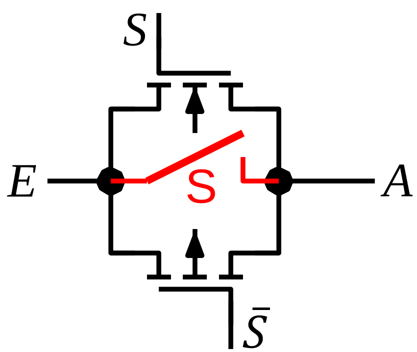
der zusammengehörigen Transistoren, verbundene Gates NMOS - PMOS

Beachte: Es darf zu keinem Zeitpunkt, bei beliebigen Schalterzuständen, eine Verbindung zwischen Masse und Betriebsspannung (U_{DD}) geben !

PMOS- und NMOS-Logik sind komplementär. Aus der PMOS-Logik kann die NMOS-Logik und umgekehrt generiert werden. Doppelte Logik in CMOS.

CMOS-Transmission Gate (Transferschalter)

Parallelschaltung von NMOS und PMOS mit komplementärer Ansteuerung.

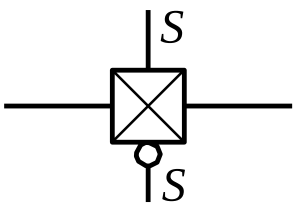


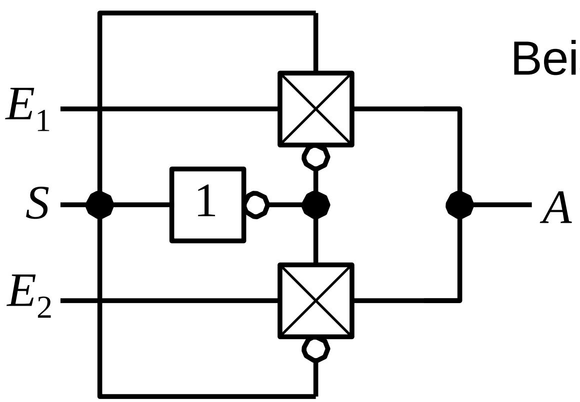
Pegel

E	S	A
L	0	Z
L	1	L
H	0	Z
H	1	H

Z - hochohmig
offener Ausgang

Schaltsymbol





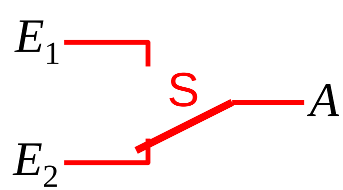
Beispiel 2:1 Multiplexer

Logik

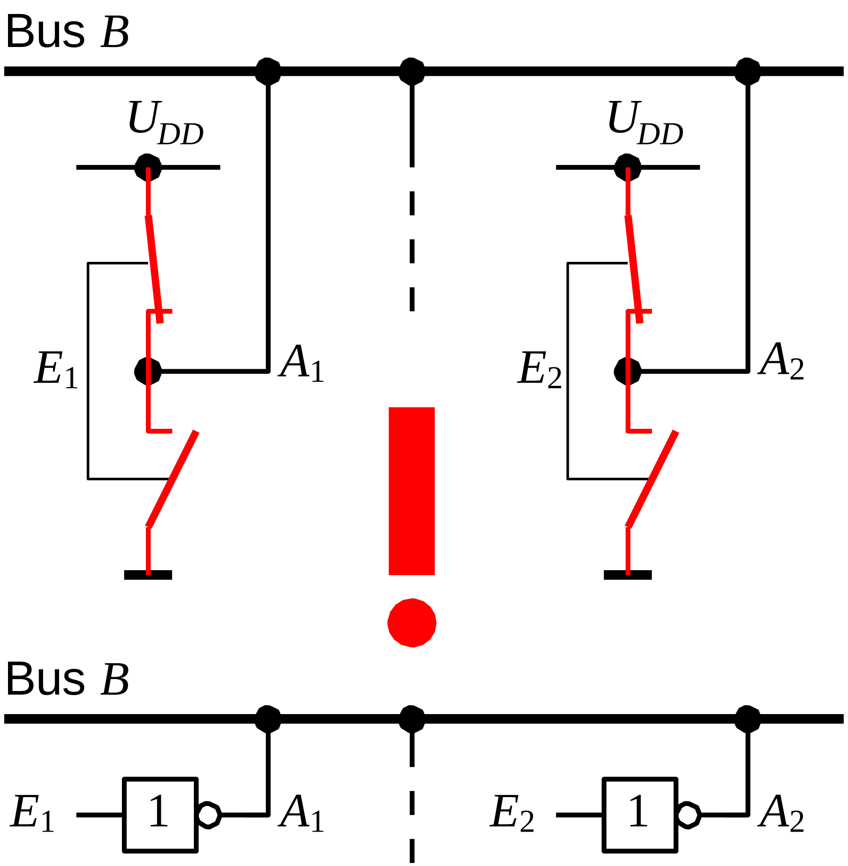
S	A
0	E_2
1	E_1

$$A = (E_1 \wedge S) \vee (E_2 \wedge \bar{S})$$

Umschalter



CMOS Ausgangsseitige Verknüpfung



- Kommunikation über eine Leitung
- Anschluss vieler Komponenten
- Alle dürfen lesen
- nur einer darf schreiben
- direkte Ausgangsverknüpfung von CMOS nicht zulässig → Kurzschluss

Pegel

A_1	A_2	B
H	H	H
H	L	-
L	H	-
L	L	L

$B=A_1=A_2$

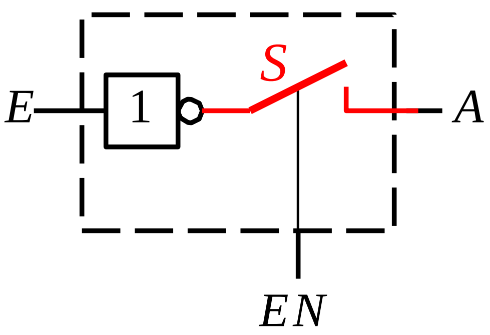
Bei unterschiedlicher Ausgangsbelegung

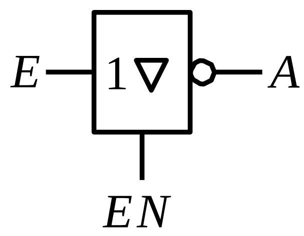
→ Kurzschluss

→ Ausweg:
Tristate-Ausgänge
Open Drain Ausgänge

CMOS Tristate

Tristate-Logik



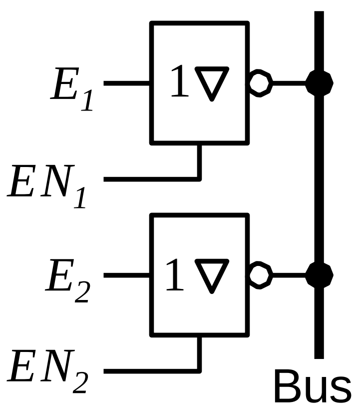


Pegel

E	EN	A
L	L	Z
L	H	H
H	L	Z
H	H	L

Z - Tristate (hochohmig, offen)

Busanbindung mit Tristate

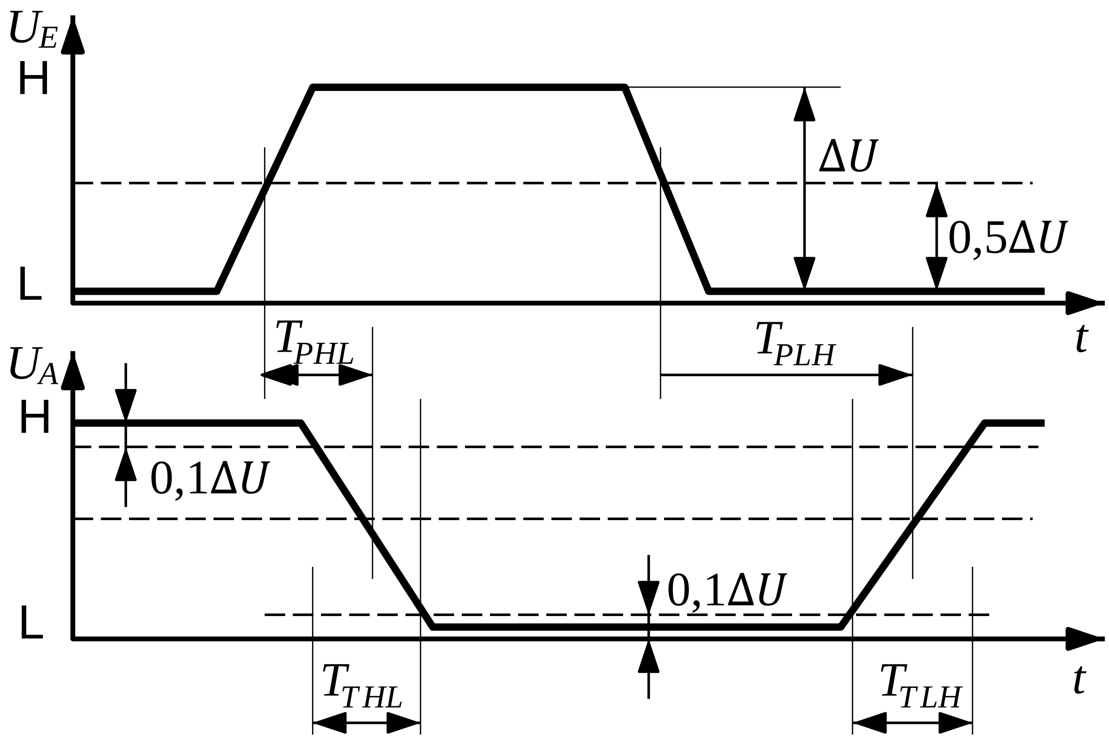


Auswahl der Busanbindung mit dem Enable-Signal EN, nur ein Enable-Signal darf aktiv sein.

$EN_1 \wedge EN_2 = 1$ Nicht erlaubt → Kurzschluss

CMOS Zeitverhalten von Verknüpfungsgliedern

Lineare Näherung des Übertragungsverhaltens eines NOT (Trapezsignal)

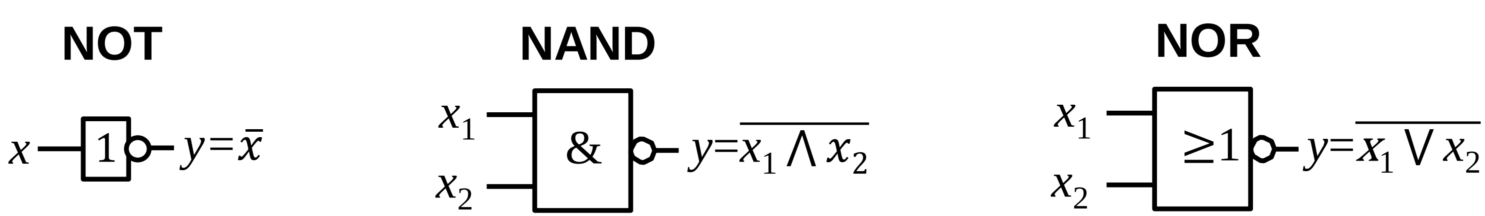


ΔU	Spannungshub
T_{PLH}	Einschaltverzögerung
T_{PHL}	Ausschaltverzögerung
T_{TLH}	Anstiegszeit
T_{THL}	Abfallzeit

H	High - Pegel
L	Low - Pegel
T_P	Propagation Delay Time
T_T	Transition Time

CMOS-Verknüpfungsglieder

- Aufgrund der Transistorschaltungstechnik sind CMOS-Verknüpfungsglieder zumeist invertierend: NOT, NAND, NOR (Standard CMOS).
- Durch Zusammenschaltungen von NOT, NAND und NOR kann jede beliebige Schaltfunktion realisiert werden.
- Unabhängig davon können in CMOS auch hochkomplexe Schaltnetze direkt realisiert werden.
- CMOS-Verknüpfungsglieder haben ein Zeitverhalten. Für statische Betrachtungen kann es jedoch oft vernachlässigt werden.



4 Schaltalgebra, Boolesche Algebra

Definitionen der Schaltalgebra:

Die Schaltalgebra ist ein Modell und eine technische Anwendung der Booleschen Algebra.

1. Es existiert eine Menge $B \in \{0,1\}$ mit folgender Zuordnung:

Zustand	positive Logik 0	positive Logik 1	negative Logik 1	negative Logik 0
Schalter	offen	geschlossen	offen	geschlossen
Pegel	low (L)	high (H)	low (L)	high (H)

Allgemein wird nur positive Logik verwendet (→ einheitlich verwenden).

Schaltvariablen sind Symbole für Elemente der Schaltalgebra aus der Menge B:

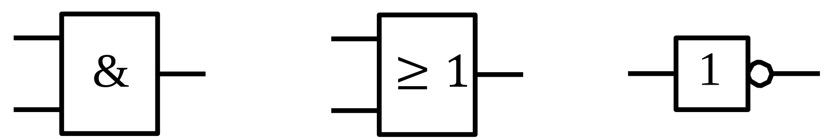
$x \in B$ mit $B=\{0, 1\}$ → x kann nur die Werte 0 oder 1 annehmen.

Schaltalgebra, Definitionen

2. Es existieren Operatoren mit folgender Zuordnung:

Operator	$\wedge$	$\vee$	$\neg$
Boolesche Verknüpfung	AND	OR	NOT
Schreibweise in Funktionen	$\cdot$	$+$	$\overline{}$

3. Einführung von Schaltsymbolen für die Darstellung der Operatoren:



4. Es gelten die Gesetze der Booleschen Algebra.

Die Boolesche Algebra ist eine algebraische Struktur
(George Boole, englischer Mathematiker, 1815 - 1864).

Boolesch: Eigenschaft von Variablen oder Funktionen binär, zweiwertig zu sein.

Gesetze der Booleschen Algebra

Die boolesche Algebra ist ein Oberbegriff aller zweiwertigen (binären) Algebren. In diesem Sinne spricht man von „booleschen Schaltvariablen“, „booleschen Schaltfunktionen“, „booleschen Verknüpfungen“. Der Ausdruck *logisch* anstelle von **boolesch** ist als missverständlich zu vermeiden.

Es existiert eine Menge $\mathrm{B} = \{a,\ b,\ c, ...,\ e,\ n\}$ und die eindeutigen Operatoren: $\wedge$, $\vee$, $\neg$.

Operator	Verknüpfung	Abbildung	Bezeichnung	
$\wedge$:	$\mathrm{B} \times \mathrm{B}$	$\rightarrow \mathrm{B}$	Konjunktion	(UND, AND)
$\vee$:	$\mathrm{B} \times \mathrm{B}$	$\rightarrow \mathrm{B}$	Disjunktion	(ODER, OR)
$\neg$:	$\times \mathrm{B}$	$\rightarrow \mathrm{B}$	Negation	(NICHT, NOT)

Bindung im Ausdruck nimmt von $\neg$ nach $\wedge$ und nach $\vee$ ab.

e - neutrales Element bzgl. der Operation $\wedge$
n - neutrales Element bzgl. der Operation $\vee$

$\Rightarrow (\mathrm{B},\vee,\wedge)$ bildet einen Körper

$n \neq e,\ \neg n = e,\ \neg e = n$

Rechengesetze der Booleschen Algebra

$\wedge$	**Gesetz**	$\vee$
$a \wedge b = b \wedge a$	Kommutativgesetz	$a \vee b = b \vee a$
$(a \wedge b) \wedge c = a \wedge (b \wedge c)$	Assoziativgesetz	$(a \vee b) \vee c = a \vee (b \vee c)$
$a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c)$	Distributivgesetz	$a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c)$
$a \wedge (a \vee b) = a$	Absorptionsgesetz	$a \vee (a \wedge b) = a$
$a \wedge 0 = 0$	Aufhebung	$a \vee 1 = 1$
$a \wedge 1 = a$	Identität	$a \vee 0 = a$
$a \wedge a = a$	Idempotent	$a \vee a = a$
$a \wedge \bar{a} = 0$	Komplement	$a \vee \bar{a} = 1$
$\overline{a \wedge b} = \bar{a} \vee \bar{b}$	De Morgan Theorem	$\overline{a \vee b} = \bar{a} \wedge \bar{b}$
$\bar{\bar{a}} = a$	Doppelte Negation	$\neg a\,(\neg a) = a$
$a \wedge \bar{a} = n$	Neutrales Element	$a \vee \bar{a} = e$

Dualizäzssatz

Ist A eine Aussage der Booleschen Algebra (Boolesche Funktion), so auch $\neg A$, die man durch Vertauschen von $\wedge$ gegen $\vee$ und n gegen e erhält.

5 Schaltfunktion

Schaltvariable

Die Schaltvariable ist ein Symbol für ein Element der Schaltalgebra mit der Menge $\mathrm{B} = \{0,1\}$. Ist $\mathrm{B}=\{0,1\}$, dann bedeutet $x\in\mathrm{B}$: x hat entweder den Wert 0 oder den Wert 1. Eine andere Belegung für x ist nicht möglich.

Menge der Schaltvariablen: $\mathrm{U} = \{x_1,\ldots, x_\nu,\ldots, x_n\}$

Schaltvariable: $x_\nu\in\mathrm{U}$

Schaltvariablen $\rightarrow$ Wert (positive Logik)

Wert	Schalter	$P(x)$
x=1, $\bar{x}$=0	geschlossen	x
x=0, $\bar{x}$=1	offen	$\bar{x}$

Formalisierung:

Aussagenlogik $\rightarrow$ Prädikatenlogik:

Das (einstellige) Prädikat $P(x) = x$ beschreibt die Eigenschaft x=1 und $\bar{x}$ die Eigenschaft x=1, bzw. $\bar{x}$=0.

Wertekombinationen von Schaltvariablen

Für die Menge der Schaltvariablen $U = \{x_1,\ldots, x_\nu,\ldots, x_n\}$ mit $x_\nu \in B$ $(\nu=1,\ldots,n)$ enthält die Produktmenge (kartesisches Kreuzprodukt) B^n genau 2^n Elemente.

Diese Elemente stellen alle möglichen Wertekombinationen der Schaltvariablen $x_1,\ldots, x_n$ dar.

Beispiel

$n=2$ mit $x_1, x_2 \in B$ und es gilt $B \times B = B^2 = \{(0,0),(0,1),(1,0),(1,1)\}$

x_1	x_2	Eigenschaft	$K(x_1, x_2)$
0	0	$x_1 = 0$ und $x_2 = 0$	$\overline{x_1}\,\overline{x_2}$
0	1	$x_1 = 0$ und $x_2 = 1$	$\overline{x_1}\,x_2$
1	0	$x_1 = 1$ und $x_2 = 0$	$x_1\,\overline{x_2}$
1	1	$x_1 = 1$ und $x_2 = 1$	$x_1\,x_2$

→ $B \times B = \{\overline{x_1}\,\overline{x_2}, \overline{x_1}\,x_2, x_1\,\overline{x_2}, x_1\,x_2\}$

→ $2^2 = 4$ Kombinationen von x_1, x_2

Schaltfunktion

Eine Schaltfunktion ordnet den Wertekombinationen $K(x_1,\ldots, x_\nu,\ldots, x_n) \in \mathrm{D} \subset \mathrm{B}^n$ der Schaltvariablen $\mathrm{U} = \{x_1,\ldots, x_\nu,\ldots, x_n\}$ mit $x_\nu \in \mathrm{B}$ $(\nu=1,\ldots,n)$ eindeutig einen Funktionswert entsprechend der Zuordnungsvorschrift f zu.

Mit $\mathrm{B}=\{0,1\}$ und $\mathrm{D} \subset \mathrm{B}^n$ ist $f : \mathrm{D} \rightarrow \mathrm{B}$ eine Schaltfunktion (Abbildung) mit dem Definitionsbereich D. Dabei gilt für alle Wertekombinationen $\mathrm{K}(x_1,\ldots, x_\nu,\ldots, x_n)$ mit $x_\nu \in \mathrm{B}$ $(\nu=1,\ldots,n)$ auch $f(x_1,\ldots, x_\nu,\ldots, x_n) \in \mathrm{B}$.

$f : \mathrm{D} \rightarrow \mathrm{B}$ — f ist eine Abbildung von D in B. Die Abbildung stellt eine rechtseindeutige Relation zwischen D und B dar.

$y=f(x)$ — das Element x wird dem Element y durch f zugeordnet oder das Element x wird durch f auf das Element y abgebildet.

Funktionsgleichung (auch Term): $f\colon K(x_1,\ldots, x_\nu,\ldots, x_n) \rightarrow y$

Zuordnungsvorschrift f als Boolesche Gleichung $y = f(x_1,\ldots, xv,\ldots, x_n)$

Darstellung der Schaltfunktion, Wertetabelle

Darstellung der Schaltfunktion als Wertetabelle (Funktionstabelle):
→ tabellarische Zuordnung: Wertekombinationen → Funktionswerte

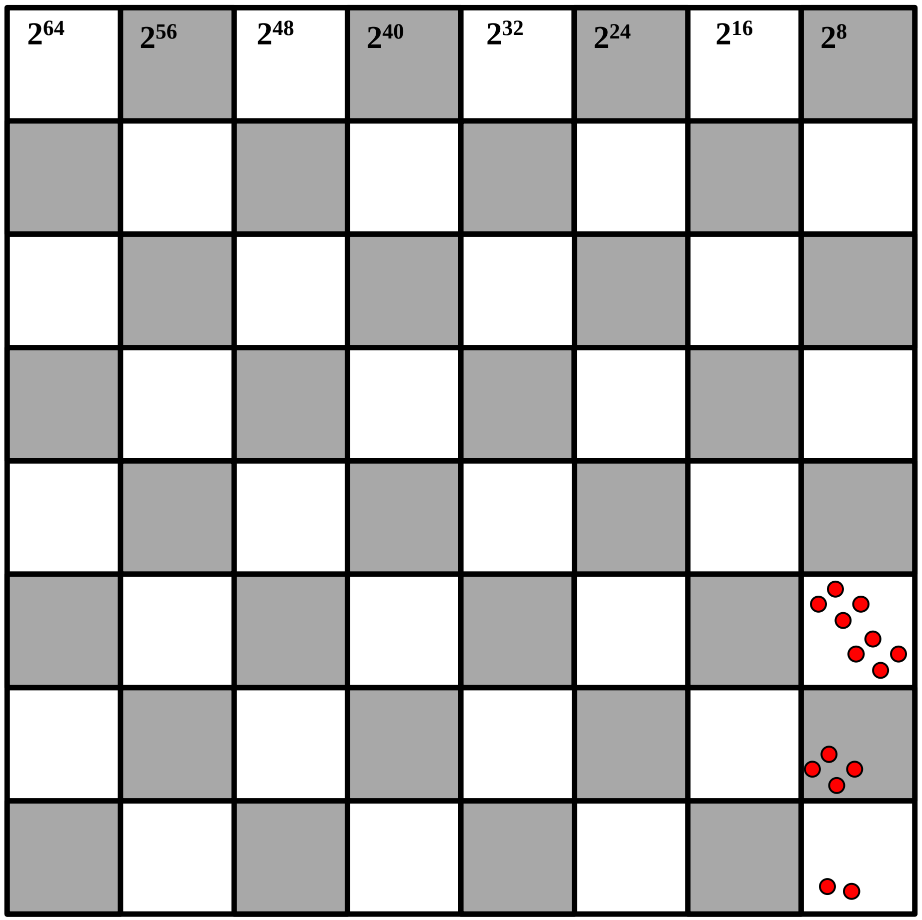
x_1	…	x_ν	…	x_n	$K(x_1,\ldots,x_\nu,\ldots,x_n)$	$y = f(x_1,\ldots,x_\nu,\ldots,x_n)$
0	…	0	…	0	$\overline{x_1} \wedge\ldots\wedge \overline{x_\nu} \wedge\ldots\wedge \overline{x_n}$	0
0	…	0	…	1	$\overline{x_1} \wedge\ldots\wedge \overline{x_\nu} \wedge\ldots\wedge x_n$	0
⋮	…	⋮	…	⋮	$\vdots \wedge\ldots\wedge \vdots \wedge\ldots\wedge \vdots$	⋮
1	…	1	…	1	$x_1 \wedge\ldots\wedge x_\nu \wedge\ldots\wedge x_n$	1

Jeder Wertekombination $K(x_1,\ldots x_n)$ können genau zwei verschiedene Funktionswerte (0 oder 1) zugeordnet werden. Bei n vorgegebenen Schaltvariablen existieren genau 2^n Wertekombinationen und damit $2^{(2^n)}$ mögliche Schaltfunktionen dieser Schaltvariablen.

n Schaltvariable → 2^n Wertekombinationen → $2^{(2^n)}$ Schaltfunktionen

Die Anzahl der Wertekombinationen wächst exponentiell mit der Variablen-anzahl, die Anzahl der möglichen Schaltfunktionen doppelt exponentiell.

Exponentielle Komplexität auf dem Schachbrett



Ausgangspunkt: ein Schachbrett und Reiskörner die auf dem Schachbrett in folgender Weise zu platzieren sind:

- Lege auf das 1.Feld 2 Körner
- Lege auf das 2. Feld 4 Körner
- Lege auf das 3. Feld 8 Körner
- ...
- Lege immer auf das nächst folgende Feld $n+1$ die doppelten Menge Körner des letzten Feldes n.

Ist das möglich?

$2^8 = 256$

$2^{16} = 65.536$

$2^{32} = 4.294.967.296$

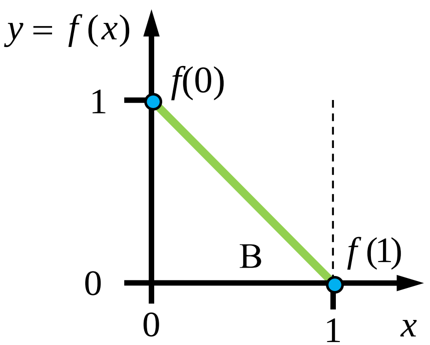
$2^{64} = 18.446.744.073.709.551.616$

$2^{128} = 340282366920938463463374607431768211456$

$2^{256} =$
$115792089237316195423570985008687907853269984665640564039457584007913129639936$

Grafische Darstellung der Schaltfunktion

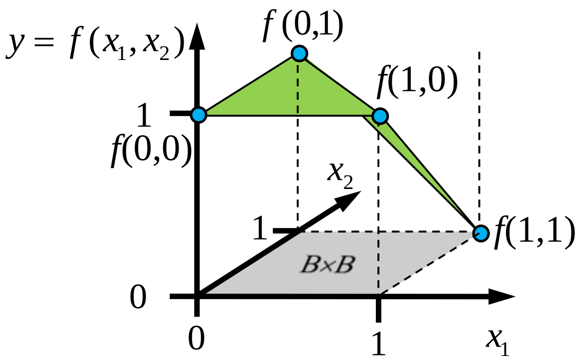
Funktion mit einer Variablen



x	$y = f(x)$
0	1
1	0

→ Diskrete Funktionswerte, Kurven nur zur Veranschaulichung

Funktion mit zwei Variablen



x_1	x_2	$y = f(x_1, x_2)$
0	0	1
0	1	1
1	0	1
1	1	0

Schaltfunktionen von zwei Schaltvariablen

Beispiel: Mögliche Wertekombinationen und Schaltfunktionen bei zwei Schaltvariablen:

Für zwei Schaltvariablen $x_1, x_2 \in \mathrm{B}$ ist $\mathrm{B} \times \mathrm{B} = \{(0,0), (0,1), (1,0), (1,1)\}$. Damit existieren genau vier Wertekombinationen dieser Schaltvariablen x_1 und x_2.

Da für jede dieser Wertekombinationen der Funktionswert f gleich 0 oder 1 sein kann, existieren damit genau $2^{(2^2)}=16$ mögliche Schaltfunktionen dieser Wertekombinationen $f_\mu \in \mathrm{B}$, $\mu = 0, \ldots, 15$, die wie folgt in lexikographischer Ordnung dargestellt werden können:

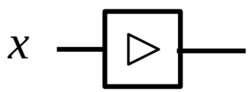
x_1	x_2	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}	f_{16}
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
		y_1	y_2	y_3	y_4	y_5	y_6	y_7	y_8	y_9	y_{10}	y_{11}	y_{12}	y_{13}	y_{14}	y_{15}	y_{16}

Schaltfunktionen von zwei Schaltvariablen

$f(x_1, x_2)$	Funktion	Bezeichnung	Kürzel
f_1	$y_1 = 0$	Konstanz	0
f_2	$y_2 = x_1 \wedge x_2$	Konjunktion	AND
f_3	$y_3 = x_1 \wedge \overline{x_2}$	Inhibition	
f_4	$y_4 = x_1$	Identität	
f_5	$y_5 = \overline{x_1} \wedge x_2$	Inhibition	
f_6	$y_6 = x_2$	Identität	
f_7	$y_7 = x_1 \wedge \overline{x_2} \vee \overline{x_1} \wedge x_2$	Antivalenz	XOR
f_8	$y_8 = x_1 \vee x_2$	Disjunktion	OR
f_9	$y_9 = \overline{x_1 \vee x_2}$	Antidisjunktion	NOR
f_{10}	$y_{10} = (x_1 \vee \overline{x_2}) \wedge (\overline{x_1} \vee x_2)$	Äquivalenz	XNOR
f_{11}	$y_{11} = \overline{x_2}$	Negation	NOT
f_{12}	$y_{12} = x_1 \vee \overline{x_2}$	Implikation	
f_{13}	$y_{13} = \overline{x_1}$	Negation	NOT
f_{14}	$y_{14} = \overline{x_1} \vee x_2$	Implikation	
f_{15}	$y_{15} = \overline{x_1 \wedge x_2}$	Antikonjunktion	NAND
f_{16}	$y_{16} = 1$	Konstanz	1

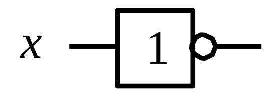
Verknüpfungsglieder für n=1, 2 Schaltvariablen

Identität (Buffer)



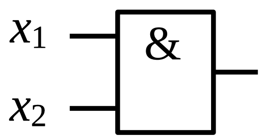
$y = x$

Negation (NOT)



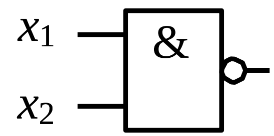
$y = \bar{x}$

Konjunktion (AND)



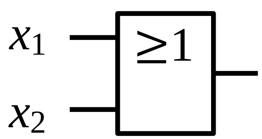
$y = x_1 \wedge x_2$

Antikonjunktion (NAND)



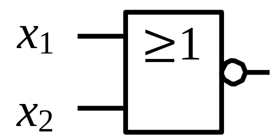
$y = \overline{x_1 \wedge x_2}$

Disjunktion (OR)



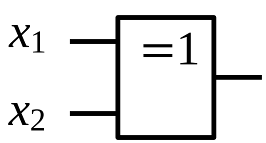
$y = x_1 \vee x_2$

Antidisjunktion (NOR)



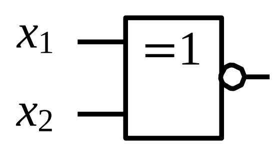
$y = \overline{x_1 \vee x_2}$

Antivalenz (XOR)

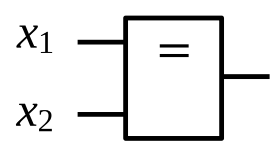


$y = x_1 \not\leftrightarrow x_2$

Äquivalenz (XNOR)



$y = \overline{x_1 \not\leftrightarrow x_2}$



$y = x_1 \leftrightarrow x_2$

Grundoperationssysteme (Basissysteme)

Ein Grundoperationssystem ist ein System von Grundschaltfunktionen minimaler Anzahl, aus denen jede beliebige Schaltfunktion durch Superposition (Überlagerung) realisiert werden kann. Das Prinzip der funktionellen Vollständigkeit ist dabei unbedingt zu beachten.

Jedes Grundoperationssystem muss in der Lage sein die Grundfunktionen AND, OR und NOT zu realisieren (→ Boolesche Algebra). Damit ist automatisch jede beliebige Schaltfunktion realisierbar.

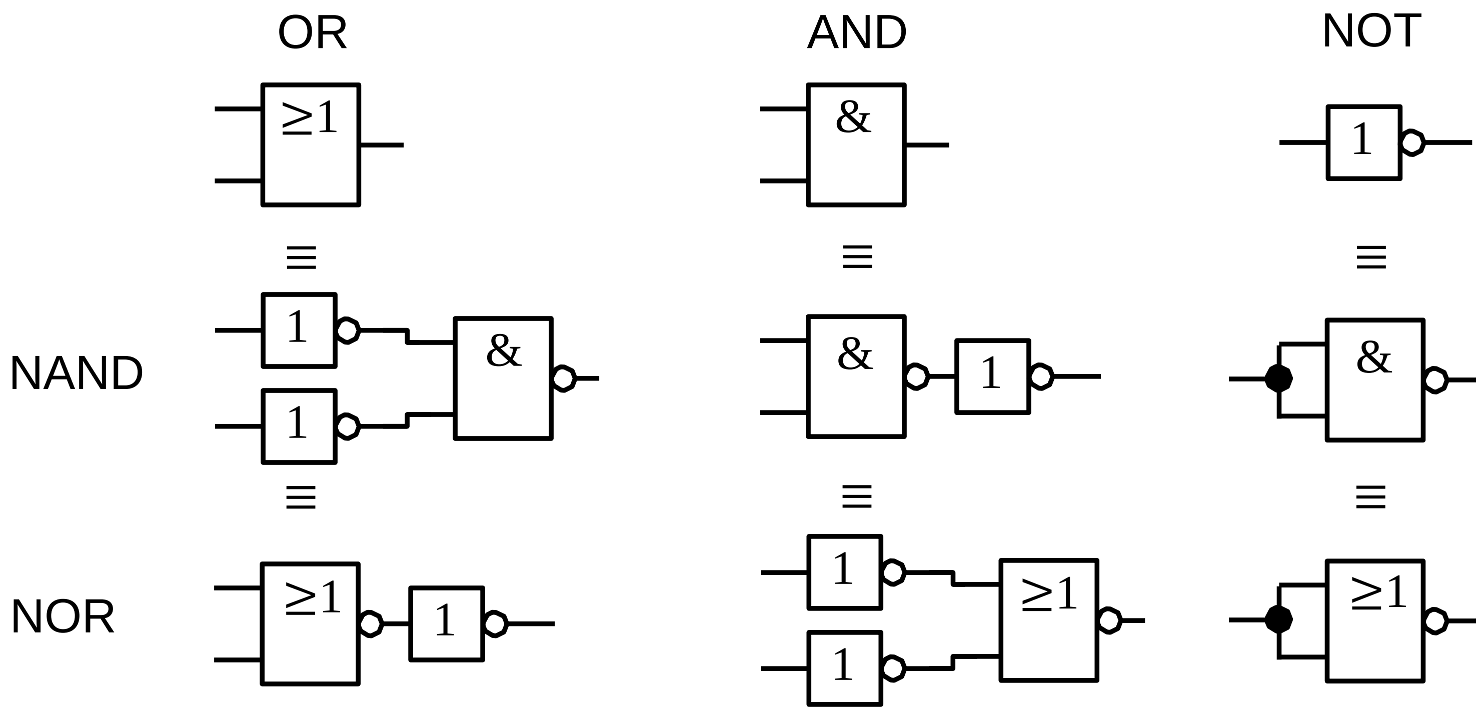
44 Grundoperationssysteme in binärer Logik :

- 2 mit einer Funktion
- 16 mit zwei Funktionen
- 23 mit drei Funktionen
- 3 mit vier Funktionen.

System 1	System 2	System 3	System 4	System 5	System 6	. . .
AND	AND	OR	NAND	NOR	AND	. . .
OR		NOT			XOR	. . .
NOT	NOT				1	. . .

Grundoperationssysteme NAND, NOR

Mit nur NAND oder nur NOR Verknüpfungen ist jede beliebige Schaltfunktion realisierbar.

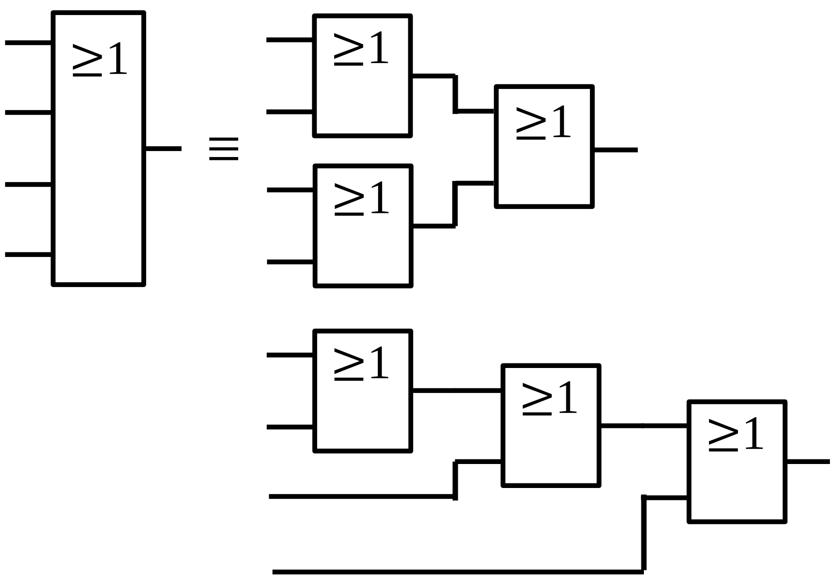


$x_1 \vee x_2 = \overline{\overline{x_1 \vee x_2}} = \overline{\overline{x_1} \wedge \overline{x_2}}$ $x_1 \wedge x_2 = \overline{\overline{x_1 \wedge x_2}} = \overline{\overline{x_1} \vee \overline{x_2}}$ $\bar{x} = \overline{x \vee x} = \overline{x \wedge x}$

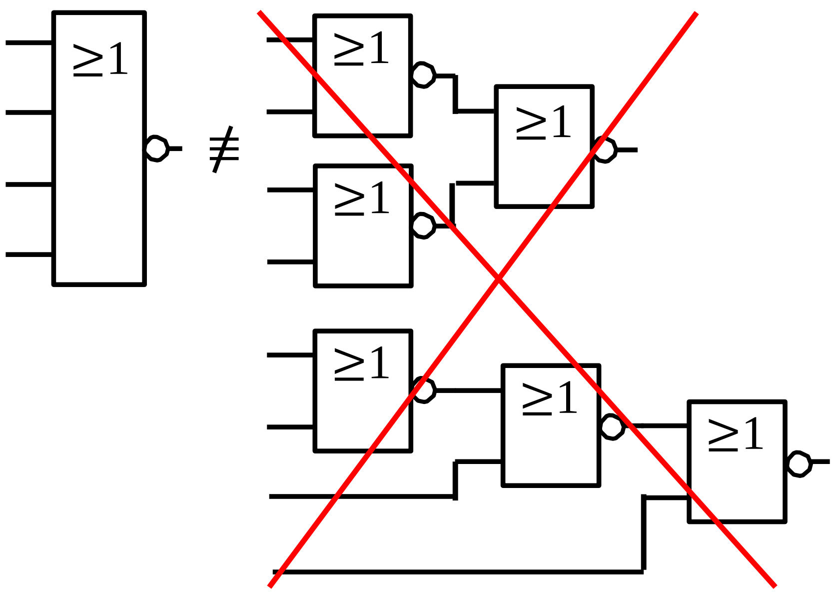
Verknüpfungen mit mehr als 2 Schaltvariablen

Verknüpfungen mit mehr als zwei Schaltvariablen können auf Verknüpfungen mit zwei und einer Schaltvariablen zurückgeführt werden.

Beachte: NAND und NOR sind nicht assoziativ !



z.B.: AND, OR, XOR



z.B.: NAND, NOR

Normalformen der Schaltfunktionen

Normalformen beschreiben eine Schaltfunktion z.B. ausgehend von einer Wertetabelle in Gleichungsform als Boolesche Gleichung.

SHANNON'sche Zerlegung:

Entwicklung einer Schaltfunktion $f(x_1,\dots,x_\nu,\dots,x_n)$ nach der Schaltvariablen x_ν

$$f(x_1,...,x_\nu,...,x_n) = x_\nu \wedge f(x_1,...,x_\nu = 1,...,x_n) \vee \overline{x_\nu} \wedge f(x_1,...,x_\nu = 0,...,x_n)$$

Bei n Schaltvariablen ergeben sich bei vollständiger Zerlegung nach allen Schaltvariablen 2^n Terme.

Die Zerlegung (Entwicklung) kann sowohl bzgl. $\vee$ oder auch $\wedge$ durchgeführt werden (Dualitätsprinzip).

Beispiel: Shannon'sche Zerlegung

Vollständige Zerlegung einer Schaltfunktion $f(x_1, x_2)$ mit zwei Schaltvariablen x_1, x_2 mit Hilfe der Wertetabelle.

Wertetabelle der Schaltfunktion

x_1	x_2	$y = f(x_1, x_2)$
0	0	0
1	0	1
0	1	1
1	1	0

Vollständige Zerlegung der Schaltfunktion

$$y = f(x_1, x_2)$$

$$x_1 \rightarrow \quad = \overline{x_1} \wedge f(0, x_2) \vee x_1 \wedge f(1, x_2)$$

$$x_2 \rightarrow \quad = \overline{x_1} \wedge \overline{x_2} \wedge f(0,0) \vee \overline{x_1} \wedge x_2 \wedge f(0,1) \vee x_1 \wedge \overline{x_2} \wedge f(1,0) \vee x_1 \wedge x_2 \wedge f(1,1)$$

$$y = \overline{x_1} \wedge \overline{x_2} \wedge 0 \vee \overline{x_1} \wedge x_2 \wedge 1 \vee x_1 \wedge \overline{x_2} \wedge 1 \vee x_1 \wedge x_2 \wedge 0$$

$$y = \overline{x_1} \wedge x_2 \vee x_1 \wedge \overline{x_2}$$

Minterme und Maxterme

Minterme sind Konjunktionen von Schaltvariablen einer Schaltfunktion, die alle Schaltvariablen einmal (negiert oder nichtnegiert) enthalten.

$$K_n^d = \bigwedge_{v=1}^{n} \tilde{x}_v$$

n – Anzahl der Schaltvariablen
d – duale Wertigkeit des Minterms
$\sim$ – negiert beim Wert 0 der Schaltvariablen $(0 \rightarrow \bar{x})$
nichtnegiert beim Wert 1 der Schaltvariablen $(1 \rightarrow x)$

Jeder Minterm hat nur bei einer Kombination der Schaltvariablen (in der Wertetabelle) den Wert 1, bei allen anderen den Wert 0.

Maxterme sind Disjunktionen von Schaltvariablen einer Schaltfunktion, die alle Schaltvariablen einmal (negiert oder nichtnegiert) enthalten.

$$D_n^d = \bigvee_{v=1}^{n} \tilde{x}_v$$

n – Anzahl der Schaltvariablen
d – duale Wertigkeit des Minterms
$\sim$ – negiert beim Wert 1 der Schaltvariablen $(1 \rightarrow \bar{x})$
nichtnegiert beim Wert 0 der Schaltvariablen $(0 \rightarrow x)$

Jeder Maxterm hat nur bei einer Kombination der Schaltvariablen (in der Wertetabelle) den Wert 0, bei allen anderen den Wert 1.

Bei n Schaltvariablen gibt es genau 2^n verschiedene Minterme bzw. Maxterme.

Normalformen der Schaltfunktion

Kanonisch Disjunktive Normalform (KDNF)

Disjunktion (Oder-Verknüpfung) aller Minterme der Schaltfunktion (Minterme mit dem Funktionswert 1 in der Wertetabelle).

$$f(x_1, \ldots, x_v, \ldots, x_n) = \bigvee_d K_d^n$$

Kanonisch Konjunktive Normalform (KKNF)

Konjunktion (Und-Verknüpfung) aller Maxterme der Schaltfunktion (Maxterme mit dem Funktionswert 0 in der Wertetabelle).

$$f(x_1, \ldots, x_v, \ldots, x_n) = \bigwedge_d D_d^n$$

Jede Schaltfunktion ist als KDNF oder KKNF darstellbar.
Beide Darstellungen sind äquivalent und ineinander überführbar.

Schaltfunktionen in KDNF bzw. KKNF sind direkt vergleichbar. Sie können sofort aus einer Wertetabelle herausgelesen bzw. in eine eingetragen werden.

Disjunktive Normalformen werden allgemein bevorzugt.

Beispiel Wertetabelle → KDNF und KKNF

x_1	x_2	x_3	y	Minterme	Maxterme
0	0	0	0		$x_1 \vee x_2 \vee x_3$
0	0	1	1	$\overline{x_1} \wedge \overline{x_2} \wedge x_3$	
0	1	0	1	$\overline{x_1} \wedge x_2 \wedge \overline{x_3}$	
0	1	1	0		$x_1 \vee \overline{x_2} \vee \overline{x_3}$
1	0	0	1	$x_1 \wedge \overline{x_2} \wedge \overline{x_3}$	
1	0	1	0		$\overline{x_1} \vee x_2 \vee \overline{x_3}$
1	1	0	0		$\overline{x_1} \vee \overline{x_2} \vee x_3$
1	1	1	1	$x_1 \wedge x_2 \wedge x_3$	

KDNF:

$$y = \overline{x_1} \wedge \overline{x_2} \wedge x_3 \vee \overline{x_1} \wedge x_2 \wedge \overline{x_3} \vee x_1 \wedge \overline{x_2} \wedge \overline{x_3} \vee x_1 \wedge x_2 \wedge x_3$$

KKNF:

$$y = (x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \overline{x_2} \vee \overline{x_3}) \wedge (\overline{x_1} \vee x_2 \vee \overline{x_3}) \wedge (\overline{x_1} \vee \overline{x_2} \vee x_3)$$

Verkürzte Normalform , don‘t care Terme

Schaltfunktionen in kanonischer Normalform sind oft redundant, sie lassen sich weiter vereinfachen, verkürzen (Rechenregeln).

Bezeichnung der verkürzten Normalformen

- Disjunktive Normalform (DNF statt KDNF)
- Konjunktive Normalform (KNF statt KKNF)

Don‘t car-Terme

Bei technischen Anwendungen von Schaltfunktionen kann oft nicht für jede mögliche Kombination der Schaltvariablen der Wert der Schaltfunktion fest vorgegeben bzw. definiert werden. Diese Kombinationen kommen entweder nicht vor, oder aber sie haben keinen Einfluss auf die gewünschte Funktion (don‘t care, Funktionswert beliebig, aber 0 oder 1).

Kennzeichnung: Funktionswert: X (steht für 0 oder 1)

Don‘t care Terme sind von besonderer Bedeutung für die Vereinfachung von Schaltfunktionen.

Darstellung von Schaltfunktionen

Jede Schaltfunktion kann durch eine Wertetabelle, eine Boolesche Funktion oder durch einen Schaltplan eindeutig beschrieben werden (Struktur und Verhalten).

Schaltfunktion → Schaltnetze, Netzwerke von Verknüpfungsgliedern

Signalrückführungen sind nicht zulässig → Schaltfunktionen haben kein Zeit-, Speicherverhalten.

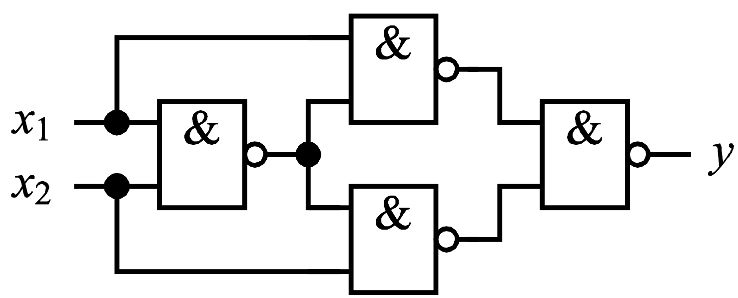
Wertetabelle
(Wahrheitstabelle)

x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	0

Boolesche Gleichung
(Funktionsgleichung)

$$y = (x_1 \wedge \bar{x}_2) \vee (\bar{x}_1 \wedge x_2)$$

Schaltplan
(Logikplan)



Vereinfachung von Schaltfunktionen

Durch Vereinfachung einer Schaltfunktion in KDNF erhält man eine verkürzte Schaltfunktion in DNF (Disjunktive Normalform, nicht kanonisch).

Analog für die KKNF → KNF.

Der Realisierungsaufwand technischer Schaltnetze kann auf diese Art und Weise oft wesentlich reduziert werden.

Methoden zur Vereinfachung von Schaltfunktionen

- Rechenregeln der Schaltalgebra
- Verfahren von Quine-McCluskey
- Karnaugh-Veitch-Diagramm (KV-Diagramm)
- …

Vereinfachung: Rechenregeln der Schaltalgebra

1. Auflösung von Negationen (De Morgan Theorem)

$\overline{a \wedge b} = \bar{a} \vee \bar{b}$ *oder* $\overline{a \vee b} = \bar{a} \wedge \bar{b}$

2. Erweitern von Termen (Komplement, Identität)

$a = a \wedge a \wedge a \wedge a \ldots$ *oder* $a = a \vee a \vee a \vee a \ldots$

$a = a \wedge (b \vee \bar{b}) = (a \wedge \mathrm{b}) \vee (a \wedge \bar{b})$ *oder* $a = a \vee (b \wedge \bar{b}) = (a \vee b) \wedge (a \vee \bar{b})$

3. Kürzen von Termen (Komplement, Identität)

$a \wedge a \wedge a \wedge a \ldots = a$ *oder* $a \vee a \vee a \vee a \ldots = a$

$(a \wedge \mathrm{b}) \vee (a \wedge \bar{b}) = a \wedge (b \vee \bar{b}) = a$ *oder* $(a \vee b) \wedge (a \vee \bar{b}) = a \vee (b \wedge \bar{b}) = a$

Als Ausgangspunkt für die Vereinfachung sollte die KDNF verwendet werden.
Eine DNF kann durch Erweitern in eine KDNF überführt werden.

→ Nur Schaltfunktionen in KDNF bzw. KKNF sind direkt vergleichbar.

Beispiele: Vereinfachung mit Rechenregeln

$$
\begin{aligned}
y &= \overline{a}\overline{b}\overline{c}\overline{d}_1 + \overline{a}b\overline{c}\overline{d}_2 + \overline{a}\overline{b}cd_3 + \overline{a}bcd_4 + abcd_5 + a\overline{b}cd_6 + abc\overline{d}_7 \\
&= \overline{a}\,\overline{c}\overline{d}(\overline{b}+b)_{1,2} + \overline{a}cd(\overline{b}+b)_{3,4} + acd(b+\overline{b})_{5,6} + abc(\overline{d}+d)_{5,7} \\
&= \overline{a}\,\overline{c}\overline{d}_{1,2} + \overline{a}cd_{3,4} + acd_{5,6} + abc_{5,7} \\
&= \overline{a}\,\overline{c}\overline{d}_{1,2} + cd(\overline{a}+a)_{3,4,5,6} + abc_{5,7} \\
&= \overline{a}\,\overline{c}\overline{d}_{1,2} + cd_{3,4,5,6} + abc_{5,7}
\end{aligned}
$$

$$
\begin{aligned}
y &= \overline{a}\overline{b}c_1 + \overline{a}b\overline{c}_2 + \overline{a}bc_3 + ab\overline{c}_4 + a\overline{b}\overline{c}_5 + a\overline{b}c_6 \\
&= \overline{a}c(\overline{b}+b)_{1,3} + b\overline{c}(\overline{a}+a)_{2,4} + a\overline{b}(\overline{c}+c)_{5,6} \\
&= \overline{a}c_{1,3} + b\overline{c}_{2,4} + a\overline{b}_{5,6}
\end{aligned}
$$

Weitere Vereinfachungsmöglichkeit:

$$
\begin{aligned}
&= \overline{b}c(\overline{a}+a)_{1,6} + \overline{a}b(\overline{c}+c)_{2,3} + a\overline{c}(b+\overline{b})_{4,5} \\
&= \overline{b}c_{1,6} + \overline{a}b_{2,3} + a\overline{c}_{4,5}
\end{aligned}
$$

Identische Schaltfunktion in DNF

Minimierung nicht immer eindeutig möglich → mehrere Schaltfunktionsminima

Verfahren von Quine-McCluskey

Unterscheiden sich zwei Terme einer Schaltfunktion in KDNF oder DNF nur in einer einzigen Schaltvariablen (in dem einen Term negiert in dem anderen jedoch nichtnegiert), so können beide Terme zu einem zusammengefasst werden, wobei die sich unterscheidende Variable gekürzt wird (entfällt).

d unterschiedlich → kürzen

Beispiel: $abcdef + abc\bar{d}ef = abcef$

Das Verfahren von Quine-McCluskey stellt einen Algorithmus zur Automatisierung dieser Vorgehensweise dar und liefert nach einer festen Anzahl von Vereinfachungsschritten eine minimale Schaltfunktion. Dieses Verfahren versagt jedoch bei Schaltfunktionen mit einer großen Anzahl von Schaltvariablen (NP-Problem).

Für große Variablenanzahlen wurden modifizierte Verfahren entwickelt.

Vereinfachung: Karnaugh-Veitch - Diagramm

Merkmale

- Grafische Darstellung der Funktionswerte,
  ebenes Diagramm für KDNF oder KKNF.
- Direkt Übernahme der Funktionswerte aus der Wertetabelle,
  Wertetabelle → KV – Diagramm.
- Einfache übersichtliche Funktionswertbestimmung,
  Zuordnung Kombination (Term) → Funktionswert
- Grafisch bearbeitbar, wenig flexibel (einfache grafische Algorithmen),
  Vereinfachungsmöglichkeiten sind sofort optisch sichtbar.
- Nur für wenige Variablen praktisch geeignet,
  unübersichtlich bei mehr als 4 Variablen.
- Nicht trivial auf dem Computer realisierbar
  → Quine-McCluskey funktioniert ähnlich.

Konstruktion des KV-Diagramms (KDNF)

Bei n Schaltvariablen enthält die Matrix 2^n Felder. Jedes Feld entspricht genau einem Minterm, gekennzeichnet durch Zeile und Spalte (analog für KKNF).

Die Aufteilung der Schaltvariablen in den Köpfen der Zeilen und Spalten der Matrix erfolgt so, dass sich benachbarte Zeilen oder Spalten nur in der Belegung einer einzigen Schaltvariablen unterscheiden (nichtnegiert, negiert). Erste und letzte Zeile bzw. Spalte der Matrix sind ebenfalls benachbart.

Eintragen aller Minterme der Schaltfunktion in KDNF entsprechend der Variablenbelegung der Zeilen und Spalten der Matrix (mit 1 belegt). Alle anderen Felder werden analog zur Wertetabelle mit 0 belegt.

Wertetabelle → KV–Diagramm

KV – Diagramm
4 Schaltvariable

y	$\bar{a}\bar{b}$	$\bar{a}b$	ab	$a\bar{b}$
$\bar{c}\bar{d}$				0
$\bar{c}d$				
cd		1		
$c\bar{d}$				

$y = f(abcd)$

Minterm: $\bar{a}bcd$

3 Schaltvariable

y	$\bar{a}\bar{b}$	$\bar{a}b$	ab	$a\bar{b}$
$\bar{c}$				1
c				

$y = f(abcd)$

Minterm: $a\bar{b}\bar{c}$

2 Schaltvariable

y	$\bar{a}$	a
$\bar{b}$		
b		1

$y = f(ab)$

Minterm: ab

Wertetabelle

a b c d	y
…	
0 1 1 1	1
1 0 0 0	0
…	

Schaltfunktion → KV-Diagramm

Ausgangsfunktion: $y = f(x_4, x_3, x_2, x_1)$

$$= \overline{x_4}\,\overline{x_3}\,\overline{x_2}\,\overline{x_1} + \overline{x_4}\,\overline{x_3}\,x_2\,\overline{x_1} + \overline{x_4}\,x_3\,\overline{x_2}\,\overline{x_1} + \overline{x_4}\,x_3\,\overline{x_2}\,x_1$$

$$\overline{x_4}\,x_3\,x_2\,\overline{x_1} + \overline{x_4}\,x_3\,x_2\,x_1 + x_4\,\overline{x_3}\,x_2\,\overline{x_1} + x_4\,\overline{x_3}\,x_2\,x_1$$

KV-Diagramm in KDNF für 4 Variablen

y	$\overline{x_2}\,\overline{x_1}$	$\overline{x_2}\,x_1$	$x_2\,x_1$	$x_2\,\overline{x_1}$
$\overline{x_4}\,\overline{x_3}$	K_0	K_1	K_3	K_2
$\overline{x_4}\,x_3$	K_4	K_5	K_7	K_6
$x_4\,x_3$	K_{12}	K_{13}	K_{15}	K_{14}
$x_4\,\overline{x_3}$	K_8	K_9	K_{11}	K_{10}

KV-Diagramm mit Funktionswerten

y	$\overline{x_2}\,\overline{x_1}$	$\overline{x_2}\,x_1$	$x_2\,x_1$	$x_2\,\overline{x_1}$
$\overline{x_4}\,\overline{x_3}$	1	0	0	1
$\overline{x_4}\,x_3$	1	1	1	1
$x_4\,x_3$	0	0	0	0
$x_4\,\overline{x_3}$	0	0	1	1

Vereinfachungsregeln mit dem KV-Diagramm

1. Im KV-Diagramm für KDNF sind alle Minterme (1) zu kennzeichnen.
2. Benachbarte Felder einer Zeile bzw. Spalte die mit 1 belegt sind können zu Blöcken zusammengefasst werden. Das erste und das letzte Feld einer Zeile bzw. Spalte sind ebenfalls benachbart (siehe Beispiele).
3. Es dürfen nur einer, zweier, vierer, achter, … Blöcke gebildet werden.
4. Die einzelnen Blöcke, dürfen sich gegenseitig überschneiden.
5. Durch die Blöcke müssen alle Minterme (1) abgedeckt werden.
6. Es können mehrere gleichberechtigte Varianten von Blöcken auftreten.
7. Der Term eines Blockes enthält nur noch die Schaltvariablen, die sich innerhalb des Blockes nicht ändern (Kürzung der Terme).
8. Die vereinfachte Schaltfunktion wird aus den Termen der Blöcke gebildet.
9. Je größer ein Block, umso kürzer der zu diesem Block gehörende Term.

Maximale Vereinfachung im KV-Diagramm

- Für die maximale Vereinfachung einer Schaltfunktion im KV-Diagramm sind möglichst wenige und möglichst große Blöcke zu bilden.

- Don‘t care können so belegt werden, dass weniger Blöcke oder und größere Blöcke entstehen.

Beispiele zur Vereinfachung mit dem KV-Diagramm

y	$\bar a\bar b$	$\bar a b$	ab	$a\bar b$
$\bar c\bar d$	1^{1}	1	0	0
$\bar c d$	0	0	0	0
cd	1^{2}	1	1	1
$c\bar d$	0	0	$^{3}1$	0

$$y = \bar a\,\bar c\bar d_1 + cd_2 + abc_3$$

y	$\bar a\bar b$	$\bar a b$	ab	$a\bar b$
$\bar c\bar d$	0	1	0	0
$\bar c d$	0	$^{1}1$	0	0
cd	$^{2}1$	0	1	1
$c\bar d$	0	0	$^{3}1$	1

$$y = \bar a b\bar c_1 + \bar b cd_2 + ac_3$$

y	$\bar a\bar b$	$\bar a b$	ab	$a\bar b$
$\bar c\bar d$	1^{1}	1	0	0
$\bar c d$	0	0	1^{2}	0
cd	1	$^{3}1$	1	1
$c\bar d$	1	1	1	1

$$y = \bar a\bar d_1 + abd_2 + c_3$$

y	$\bar a\bar b$	$\bar a b$	ab	$a\bar b$
$\bar c$	0	1	$^{2}1$	1
c	1^{1}	1	0	$^{3}1$

$$y = \bar a c_1 + b\bar c_2 + a\bar b_3$$

y	$\bar a\bar b$	$\bar a b$	ab	$a\bar b$
$\bar c$	0	1	1^{3}	1
c	$^{1}1$	$^{2}1$	0	1

$$y = \bar b c_1 + \bar a b_2 + a\bar c_3$$

y	$\bar a$	a
$\bar b$	0	$^{1}1$
b	1^{2}	1

$$y = a_1 + b_2$$

Don‘t care Terme Alternativ Beschriftung

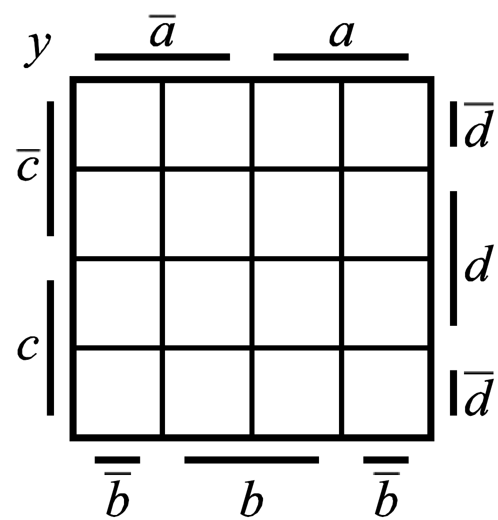
Don’t care Terme X werden entweder wie 0 oder wie 1 behandelt. Die Wahl ob als 0 oder als 1 erfolgt entsprechend der Vereinfachung. Ist eine 1 für die Blockbildung günstiger so wird X wie eine 1 verwendet, ansonsten wie eine 0.

y	$\overline{a}\overline{b}$	$\overline{a}b$	ab	$a\overline{b}$
$\overline{c}\overline{d}$	1^1	X	X	0
$\overline{c}d$	0	X	0	0
cd	X^2	X	1	1
$c\overline{d}$	0	0	$^3 1$	0

$$y = \overline{a}\overline{c}\overline{d}_1 + cd_2 + abc_3$$

don’t care Terme

cd \ ab (y)	00	01	11	10
00				
01				
11				
10				



alternative Beschriftungsarten des KV-Diagramms

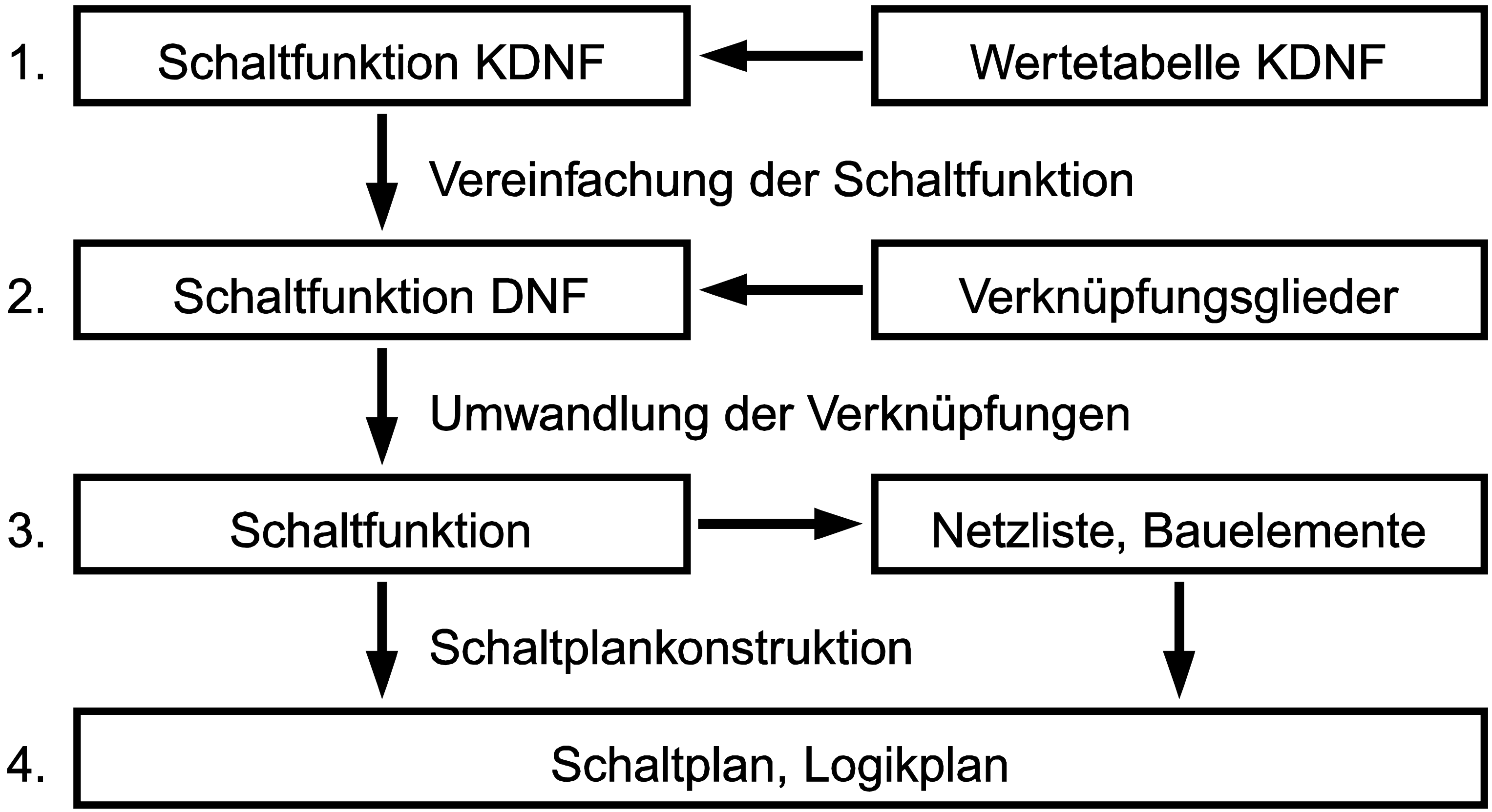
5 Schaltnetze

Schaltfunktionen → Schaltnetz → Netzwerk von Verknüpfungsgliedern

- Die Schaltfunktion wird nach der Vereinfachung so umgeformt, dass nur noch die Verknüpfungen Verwendung finden, die den vorgegebenen Verknüpfungsgliedern entsprechen.
- Entsprechend der so umgeformten Schaltfunktion kann ein Schaltnetz (Schaltplan) konstruiert werden.
- Die Beschreibung von Schaltnetzen kann auch durch eine Netzliste (Verdrahtungsplan) oder durch Hardware-Beschreibungssprachen (HDL - Hardware Description Language) erfolgen.

Schaltnetze können auch auf höherem Abstraktionsniveau durch Speicher (ROM - Read Only Memory) oder Hardware-programmierbare Schaltungen (PLD - Programmable Logic Devices) realisiert werden.

Schaltfunktion → Schaltnetz



Beispiel zur Schaltnetzsynthese

Ein Schaltnetz für die folgende Schaltfunktion ist nur unter Verwendung von NAND-Verknüpfungsglieder mit zwei Eingängen als Schaltplan zu realisieren:

1. Schaltfunktion in KDNF:

$$y = x_1\overline{x_2}\,\overline{x_3} + \overline{x_1}x_2x_3 + x_1\overline{x_2}x_3 + \overline{x_1}x_2\overline{x_3}$$

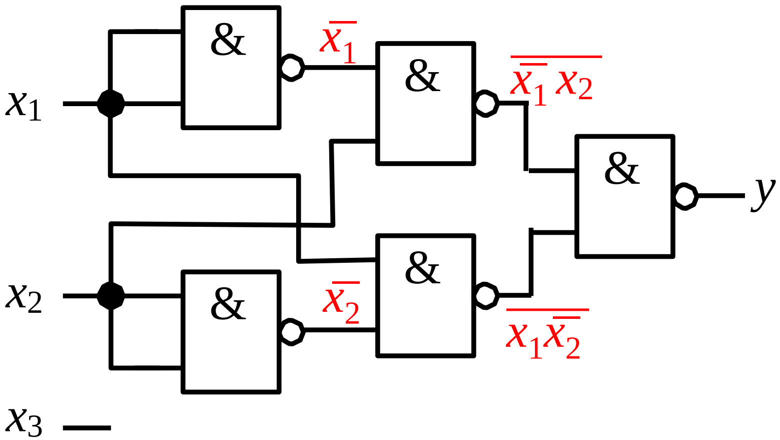
2. Vereinfachte Schaltfunktion in DNF:

$$y = x_1\overline{x_2} + \overline{x_1}x_2$$

3. Schaltfunktion unter Einbeziehung der gegebenen Verknüpfungsglieder:

$$y = \overline{\overline{x_1\overline{x_2x_2}} \cdot \overline{\overline{x_1x_1}x_2}}$$

4. Schaltnetz mit 5 NAND2 :



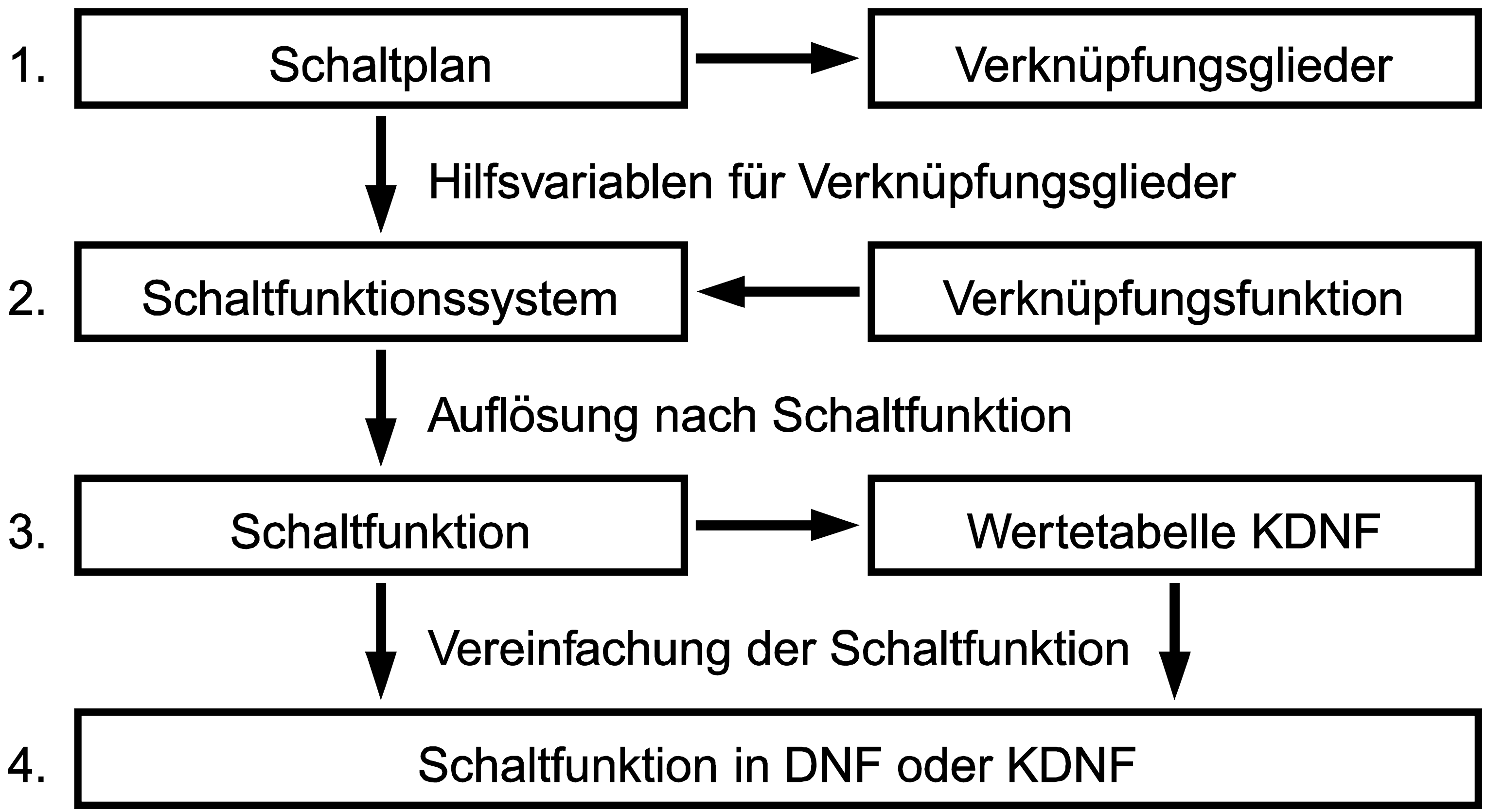
Analyse von Schaltnetzen

Die Analyse von Schaltnetzen erfolgt nach Festlegung aller Eingänge (Schaltvariablen) und Ausgänge (Schaltfunktionen) durch die schrittweise Analyse der Vernetzung und der verwendeten Verknüpfungsglieder.

- Nach der Analyse aller verwendeten Verknüpfungsglieder wird für jeden Ausgang eines Verknüpfungsgliedes eine Hilfsvariable eingeführt und die Verknüpfungsfunktion (Teilschaltfunktion) mit den jeweiligen Eingänge aufgestellt.
- Das System aller so aufgestellten Verknüpfungsfunktionen (Teilschaltfunktionen) wird dann nach der gesuchten Schaltfunktion aufgelöst.
- Bei der Auflösung geht man im ersten Schritt von den Ausgängen Richtung Eingang vor und im zweiten Schritt von den Eingängen Richtung Ausgang.

Eine Analyse über alle möglichen Wertebelegungen an den jeweiligen Ein- bzw. Ausgängen der Verknüpfungsglieder mit den Wertetabellen der Verknüpfungsglieder ist ebenfalls möglich (n Eingängen → 2^n Kombinationen).

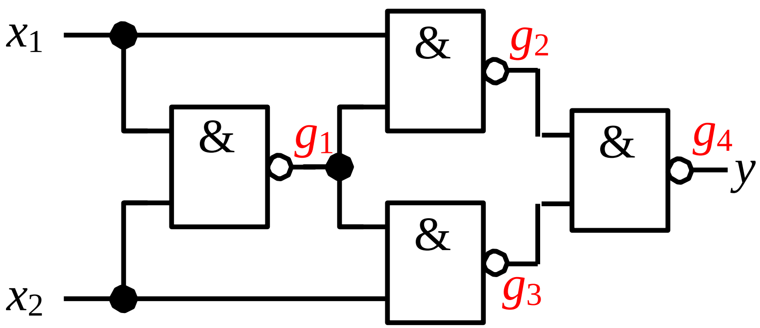
Schaltnetz → Schaltfunktion



Beispiel zur Schaltnetzanalyse

Das nachfolgende Schaltnetz ist zu analysieren, die Schaltfunktion des Schaltnetzes ist in DNF anzugeben:

1. Schaltnetz mit 4 NAND2:



Eingänge, Schaltvariable: x_1 , x_2

Ausgang, Schaltfunktion: y

$$y = x_1\overline{x_2} + \overline{x_1}x_2$$

2. Verknüpfungsfunktionen

Ausgang y

$y = g_4$

$g_4 = \overline{g_2 g_3}$

$g_3 = \overline{x_2 g_1}$

$g_2 = \overline{x_1 g_1}$

$g_1 = \overline{x_1 x_2}$

Eingänge x_1 , x_2

3.

4.

5. Vereinfachung

$y = x_1\overline{x_2} + \overline{x_1}x_2$

$g_4 = \overline{\overline{x_1\overline{x_1 x_2}}\;\overline{x_2\overline{x_1 x_2}}}$

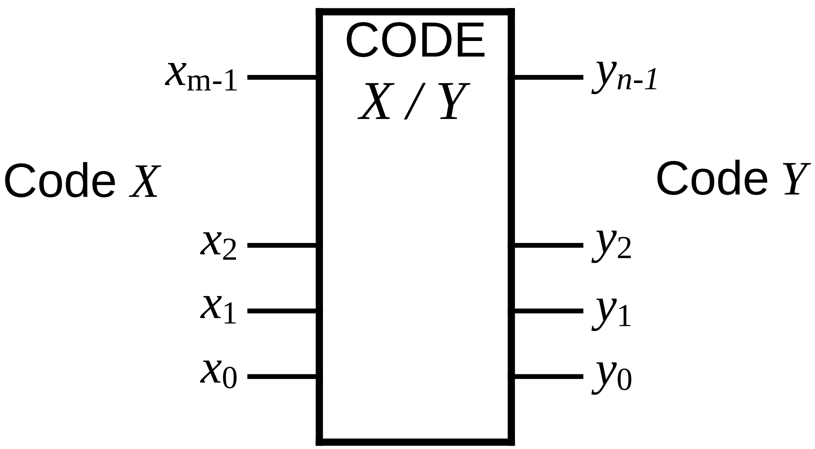
$g_3 = \overline{x_2\overline{x_1 x_2}}$

$g_2 = \overline{x_1\overline{x_1 x_2}}$

Beispiel Codierer, Decodierer

Codierer sind Schaltnetze, die Schaltvariablen eines Codes (Menge von Eingangsvariablen X) in Variablen eines anderen Codes (Menge von Ausgangsvariablen Y) eindeutig umsetzen, Decodierer arbeiten analog.

→ Codierer sind zuordnende Schaltnetze.



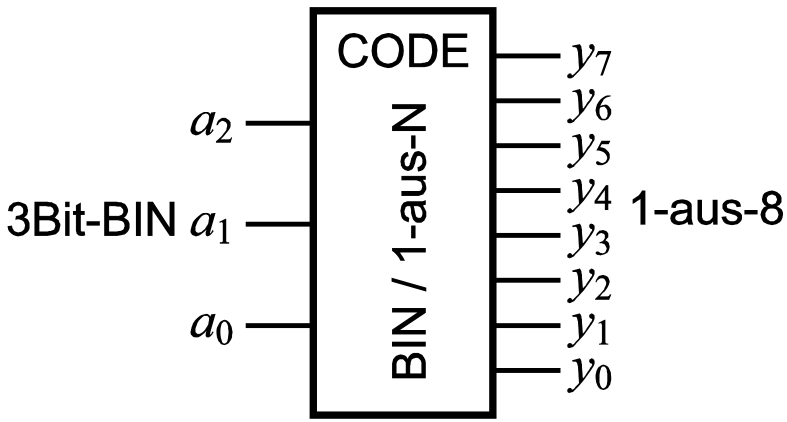
BIN	Binärcode
DEC	Dezimalcode
BCD	Binärcodierte Dezimalzahle
nGRAY	Gray-Code
1-aus-N	1-aus-N Code

BIN	←→	BCD, DEC
BIN	←→	GRAY
BIN	→	1-aus-N

3-bit Adressdecoder

Adressdecoder codieren binäre Adressen in eine 1-aus-N Codierung. Bei dieser Codierung ist nur ein Ausgang entsprechend dem Eingangsbinäräquivalent mit 1 belegt, alle anderen mit 0.

Bei m-bit Adressen gibt es 2^m Ausgänge.



$a_2 a_1 a_0$	$y_7 y_6 y_5 y_4 y_3 y_2 y_1 y_0$
0 0 0	0 0 0 0 0 0 0 1
0 0 1	0 0 0 0 0 0 1 0
0 1 0	0 0 0 0 0 1 0 0
0 1 1	0 0 0 0 1 0 0 0
1 0 0	0 0 0 1 0 0 0 0
1 0 1	0 0 1 0 0 0 0 0
1 1 0	0 1 0 0 0 0 0 0
1 1 1	1 0 0 0 0 0 0 0

$y_0 = \overline{a}_2\overline{a}_1\overline{a}_0$ $\quad y_1 = \overline{a}_2\overline{a}_1 a_0$ $\quad y_2 = \overline{a}_2 a_1\overline{a}_0$ $\quad y_3 = \overline{a}_2 a_1 a_0$

$y_4 = a_2\overline{a}_1\overline{a}_0$ $\quad y_5 = a_2\overline{a}_1 a_0$ $\quad y_6 = a_2 a_1\overline{a}_0$ $\quad y_7 = a_2 a_1 a_0$

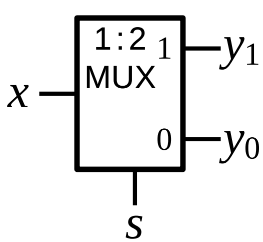
Beispiel Multiplexer, Demultiplexer

Ein Multiplexer schaltet entsprechend einer binär codierten Steueradresse s den Eingang x auf den binäräquivalenten Ausgang $y_0 \ldots y_{n-1}$ (1:N MUX) durch oder umgekehrt, den binäräquivalenten Eingang auf den Ausgang (Demultiplexer) (N:1 DMUX).

→ Multiplexer sind auswählende Schaltnetze.

Beispiel: 1:2 Multiplexer bzw. 2:1 Multiplexer (Demultiplexer) dienen auch zur Umschaltung von Datenleitungen.

Multiplexer

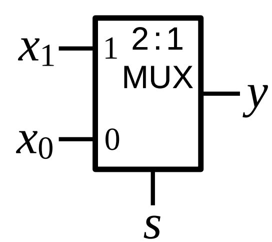


s	y_1	y_0
0	0	x
1	x	0

$$y_0 = \bar{s}x$$

$$y_0 = sx$$

Demultiplexer



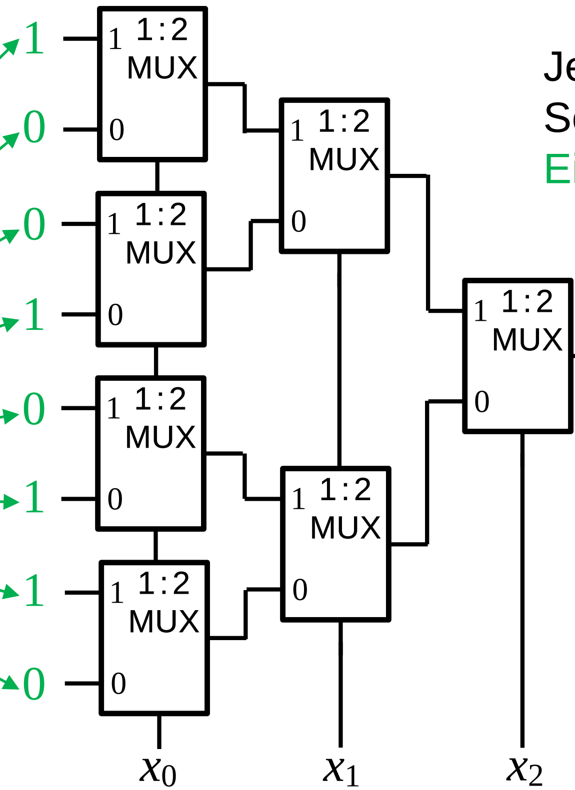
s	y
0	x_0
1	x_1

$$y = \bar{s}x_0 + sx_1$$

Beispiel: Schaltnetz mit Multiplexer (LUT)

Zusammengefasst als Multiplexerbäume können mit Multiplexern leicht komplexe Schaltnetze realisiert werden (LUT – Lookup Table).

x_2	x_1	x_0	y
1	1	1	1
1	1	0	0
1	0	1	0
1	0	0	1
0	1	1	0
0	1	0	1
0	0	1	1
0	0	0	0



Jede beliebige Schaltfunktion mit 3 Schaltvariablen ist nur durch die Eingangsbelegung realisierbar (256).

$$y = x_2 x_1 x_0 + x_2 \overline{x}_1 \overline{x}_0 + \overline{x}_2 x_1 \overline{x}_0 + \overline{x}_2 \overline{x}_1 x_0$$

n Schaltvariable

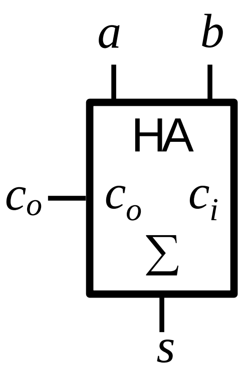
→ 2^n Kombinationen

→ $2^{(2n)}$ Schaltfunktionen

Beispiel Addierer

Addierer bilden die Grundlage für alle Rechenschaltungen (Subtrahierer, Multiplizierer, . . .). Die Addition erfolgt Modulo 2 mit Übertrag (a, b - Summanden, c_o - auslaufender-, c_i - einlaufender Übertrag, s - Summe). Addierer sind berechnende, vergleichende Schaltnetze.

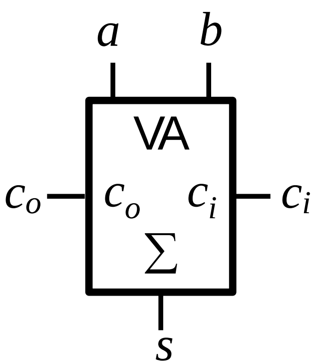
Halbaddierer



a	b	s	c_o
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

$$s = a\overline{b} + \overline{a}b$$
$$c_o = ab$$

Volladdierer

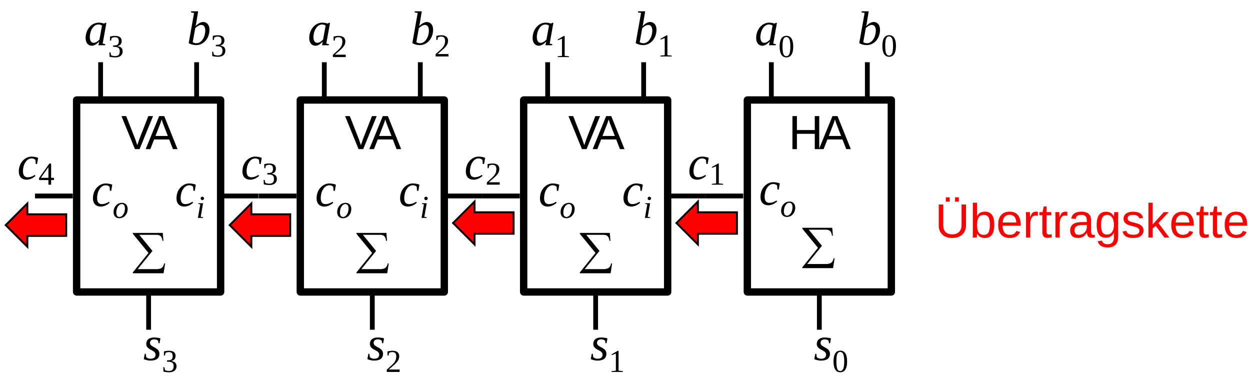


a	b	c_i	s	c_o
0	0	0	0	0
0	1	0	1	0
1	0	0	1	0
1	1	0	0	1
0	0	1	1	0
0	1	1	0	1
1	0	1	0	1
1	1	1	1	1

$$s = a\overline{b}\overline{c_i} + \overline{a}b\overline{c_i} + \overline{a}\overline{b}c_i + abc_i$$
$$c_o = ab + ac_i + bc_i$$

4-bit Ripple-Carry Adder (RCA)

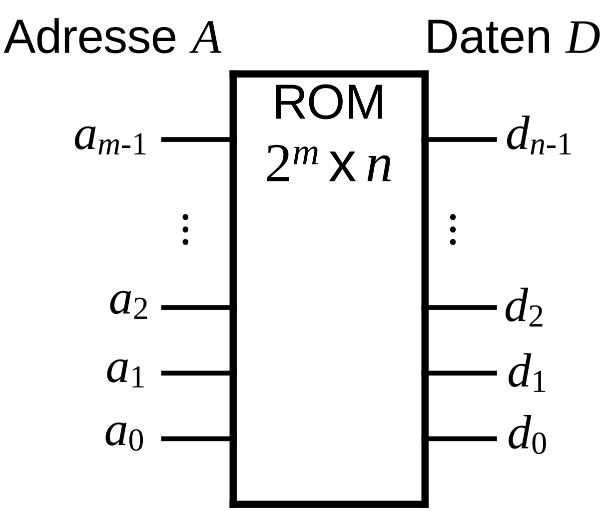
Paralleladdierer werden vorwiegend in der Computertechnik eingesetzt. Der Ripple-Carry Adder ist der einfachste, aber auch langsamste Paralleladdierer. Der Übertrag wird von Stelle zu Stelle weitergereicht, im ungünstigsten Fall durch alle Stellen.



Summanden
$$A = a_3\,2^3 + a_2\,2^2 + a_1\,2^1 + a_0\,2^0$$
$$B = b_3\,2^3 + b_2\,2^2 + b_1\,2^1 + b_0\,2^0$$
Summe
$$S = A + B = c_4\,2^4 + s_3\,2^3 + s_2\,2^2 + s_1\,2^1 + s_0\,2^0$$

Beispiel Festwertspeicher (ROM)

ROM (Read Only Memory) sind adressierbare Festwertspeicher, die vorwiegend in der Computertechnik eingesetzt werden. Einer binär codierten Adresse werden fest eingespeicherte (in Hardware) Daten zugewiesen.
ROM sind zuweisende Schaltnetze.



Speicherkapazität: 2^m x n bit

Wertetabelle des ROM (Speichertabelle)

m-bit Adresse A — n-bit Daten D (parallel)

a_{m-1}	…	a_2	a_1	a_0	d_{n-1}	…	d_2	d_1	d_0
0	…	0	0	0	$d_{n-1,0}$	…	$d_{2,0}$	$d_{1,0}$	$d_{0,0}$
0	…	0	0	1	$d_{n-1,1}$	…	$d_{2,1}$	$d_{1,1}$	$d_{0,1}$
⋮	…	⋮	⋮	⋮	⋮	…	⋮	⋮	⋮
1	…	1	1	1	$d_{n-1,2^m}$	…	$d_{2,2^m}$	$d_{1,2^m}$	$d_{0,2^m}$

$d_{\mu,\nu} \in \{0,1\}$ gespeicherte Daten

7 Zusammenfassung

- Schaltnetze sind kombinatorische Schaltungen. Verknüpfungsglieder sind Schaltnetze für einfache Booleschen Verknüpfungen → nur ein Ausgang.
- Die Werte der Ausgangsvariablen sind zu jedem Zeitpunkt nur von der Belegung der Eingangsvariablen zu genau diesem Zeitpunkt abhängig.
→ kein Zeitverhalten, → kein Speicherverhalten.
- Schaltnetze realisieren eine Schaltfunktion, die als Boolesche Funktion dargestellt werden kann.
- Die Schaltalgebra ist ein Modell und eine Anwendung der Booleschen Algebra und dient der Beschreibung, dem Entwurf, der Analyse und der Synthese von Schaltnetzen.
- Schaltnetze können durch Vernetzung von Verknüpfungsgliedern realisiert werden. Rückführungen von Ausgängen auf in Signalflussrichtung vorgelagerte Eingänge sind nicht erlaubt → Rückwirkungsfreiheit.