Rechner Architektur I (RAI)

Schaltnetze

Prof. Dr. Akash Kumar

Chair for Processor Design

Inhalt

- Schalter
- Netzwerke von Schaltern
- Elektronische Verknüpfungsglieder
- Schaltalgebra
- Boolesche Algebra
- Schaltfunktion
- Darstellung von Schaltfunktionen
- Vereinfachung von Schaltfunktionen
- Realisierung als Schaltnetz
- Analyse von Schaltnetzen
- Beispielschaltnetze

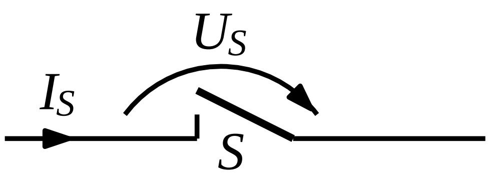
Schaltnetze

Schalter realisieren eine eindeutige Trennung zwischen zwei definierten Zuständen (binäre Entscheidung): Schalter **entweder** offen **oder** geschlossen.

Zentrale Bedeutung der Schalter bei logischen Verknüpfungen.

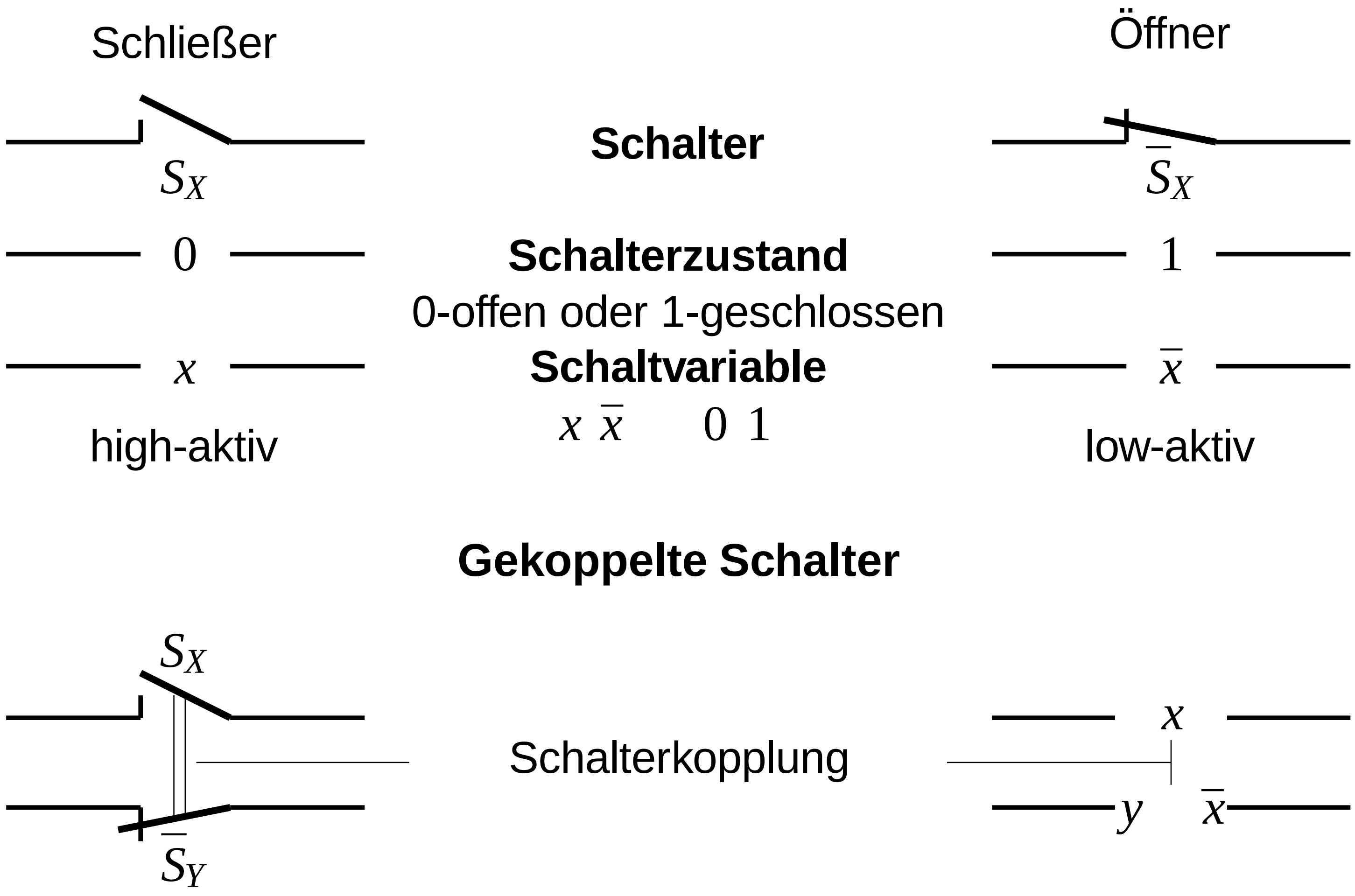
Schalter sind für die verschiedensten physikalischen Größen realisierbar: elektrisch/elektronisch, pneumatisch, hydraulisch, mechanisch, thermisch, optisch, . . .

In der Computertechnik dominieren eindeutig elektrisch-elektronische Schalter.

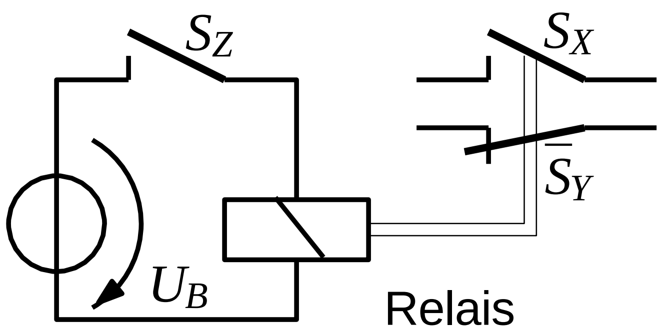


Schalter offen	: I_S	0 bei U_S	0	S 0
Schalter geschlossen	: U_S	0 bei I_S	0	S 1

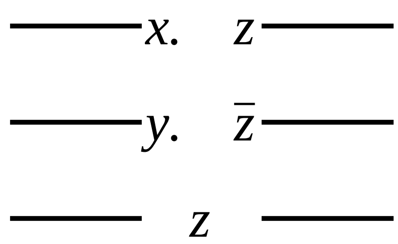
Schalterzustände - Schaltvariable



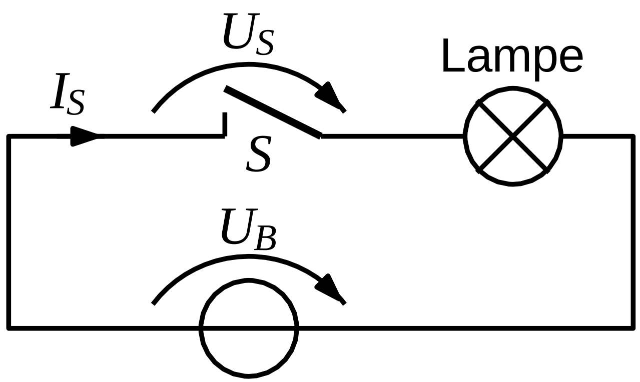
Gesteuerte Schalter



Steuerung durch Schalter S_Z



Schalter in elektrischer Schaltung

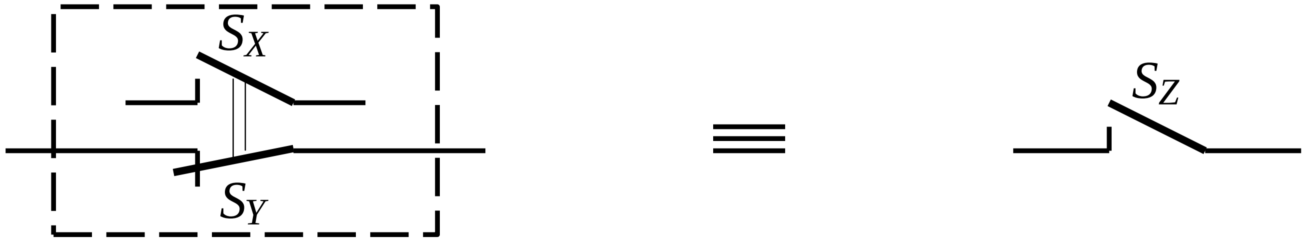


Schalter	Lampe	I_S	U_S
offen	aus	0	U_B
geschlossen	an	0	0

Netzwerke von Schaltern

Schalter sind (semantisch) identisch, wenn sie in der Schaltung gleiche Wirkungen zeigen. Der Schalter S_Z stellt hier die Ersatzschaltung für die Zusammenschaltung der Schalter S_X und S_Y dar.

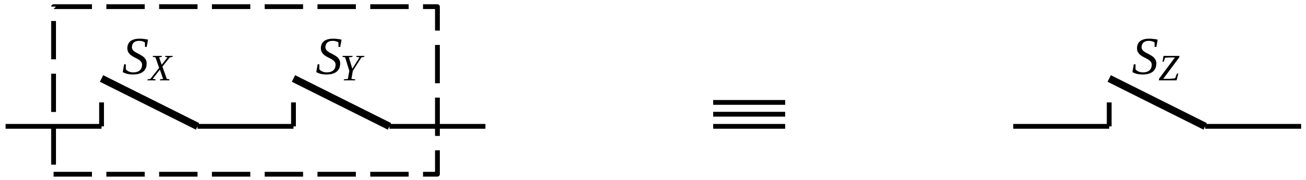
Verkopplung von Schaltern



S_Z ist dann und nur dann geschlossen, wenn S_X **nicht** geschlossen ist.

x	$z = \overline{x}$
0	1
1	0

Reihenschaltung von Schaltern



S_Z ist dann und nur dann geschlossen, wenn S_X **und** ( ) S_Y geschlossen sind.

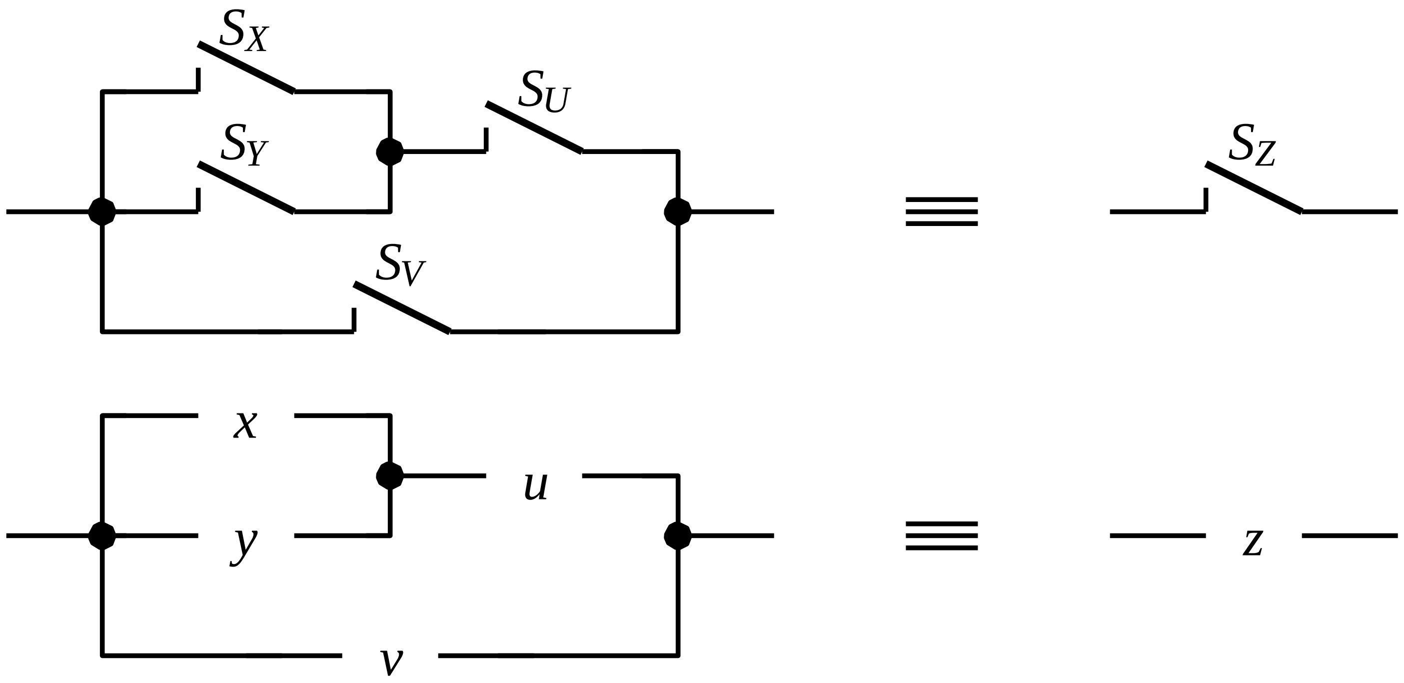
x	y	z x y
0	0	0
0	1	0
1	0	0
1	1	1

Parallelschaltung von Schaltern

S_Z ist dann und nur dann geschlossen, wenn S_X **oder / und** ( ) S_Y geschlossen sind.

x	y	z x y
0	0	0
0	1	1
1	0	1
1	1	1

Komplexe Netzwerke von Schaltern



x	y	v	u		z	v	u	x	y
0	0	0	0				0		
⋮	⋮	⋮	⋮	16 Zeilen			⋮		
1	1	1	1				1		

Verknüpfungen mit Schaltern

Reihenschaltung von S_X und S_Y	UND-Verknüpfung (AND)	$z = x \wedge y$
Parallelschaltung von S_X und S_Y	ODER-Verknüpfung (OR)	$z = x \vee y$
Verkopplung von S_X und S_Y	Negation, NICHT (NOT)	$z = \bar{x}$

Wertetabelle („Wahrheitstabelle “)

NICHT

x	$z \quad \bar{x}$
0	1
1	0

ODER

x	y	$z \quad x \quad y$
0	0	0
0	1	1
1	0	1
1	1	1

UND

x	y	$z \quad x \quad y$
0	0	0
0	1	0
1	0	0
1	1	1

Elektronische Verknüpfungsglieder

Realisierung der gesteuerten Schalter durch elektronische Bauelemente.
Als Bauelemente werden überwiegend Halbleiterbauelemente verwendet:

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**fer - Res**istor** (steuerbarer Widerstand)

Als Schaltungstechnik und Halbleitertechnologie dominiert für höchstintegrierte Digitalschaltungen (VLSI - **V**ery **L**arge **S**cale **I**ntegration) die CMOS-Technik.

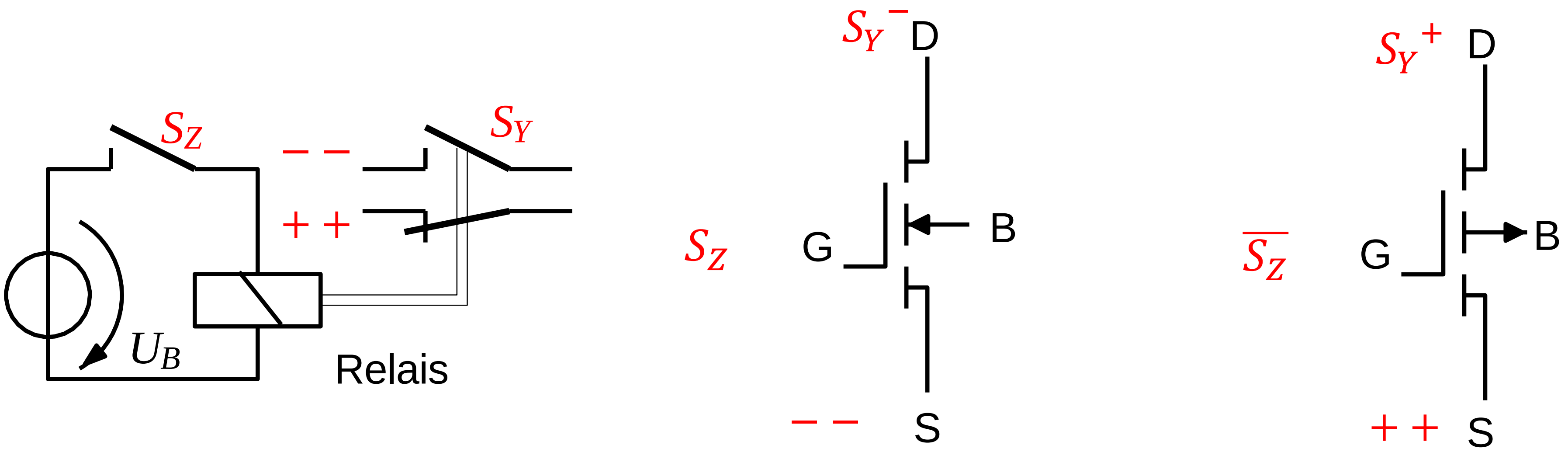
Vorteile der CMOS-Technik (Complementary MOS):

kein statischer Querstrom und Ansteuerstrom, keine statische Verlustleistung, symmetrisches Schaltverhalten, ideale statische Pegel,

geringe Abmaße, sehr gut geeignet für die Höchstintegration in Silizium.

Wirkungsweise von MOSFETs

MOSFETs ersetzen den regelbaren elektro-mechanischen Schalter (Relais) durch zwei regelbare elektronische (solid-state) "Halb"-Schalter.



n-Kanal MOS-FET (NMOS) p-Kanal MOS-FET (PMOS)

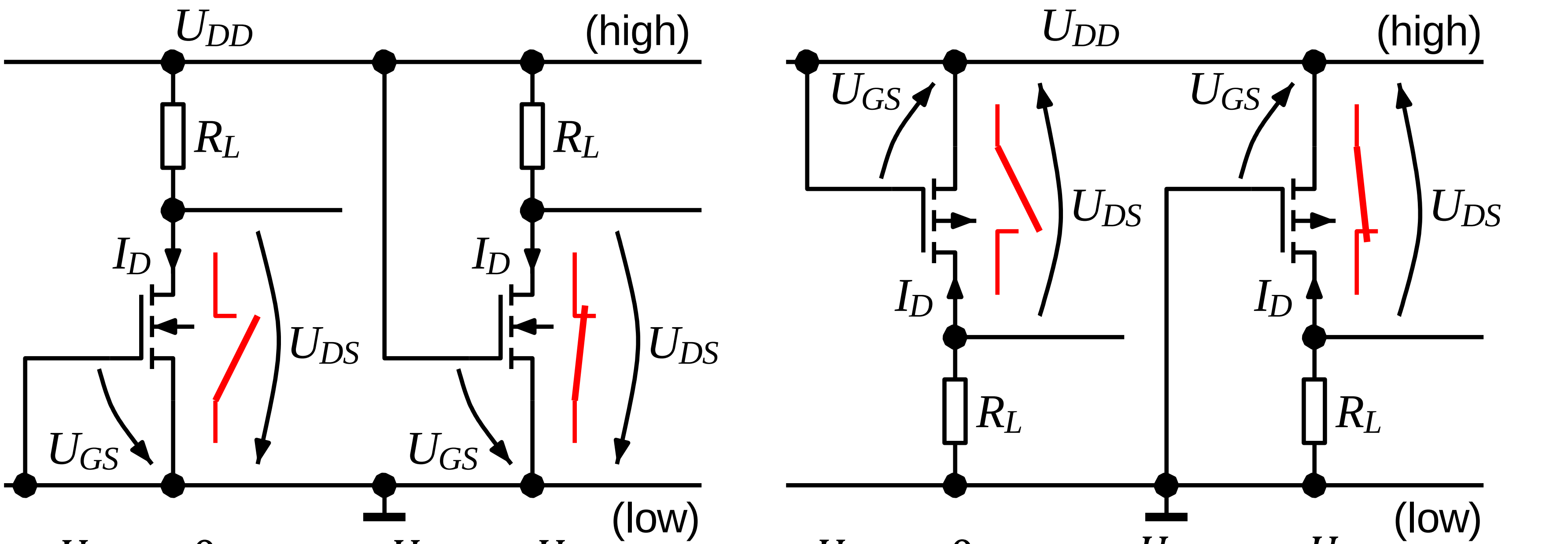
Stromgesteuerter **(nicht-)invertierender** Schalter

Spannungsgesteuerte **invertierende** Schalter

Der MOSFET als Schalter

Sperrbereich : $|U_{GS}| < |U_{Th}|$ $|I_D| = 0$ (U_{Th} - Schwellspannung)

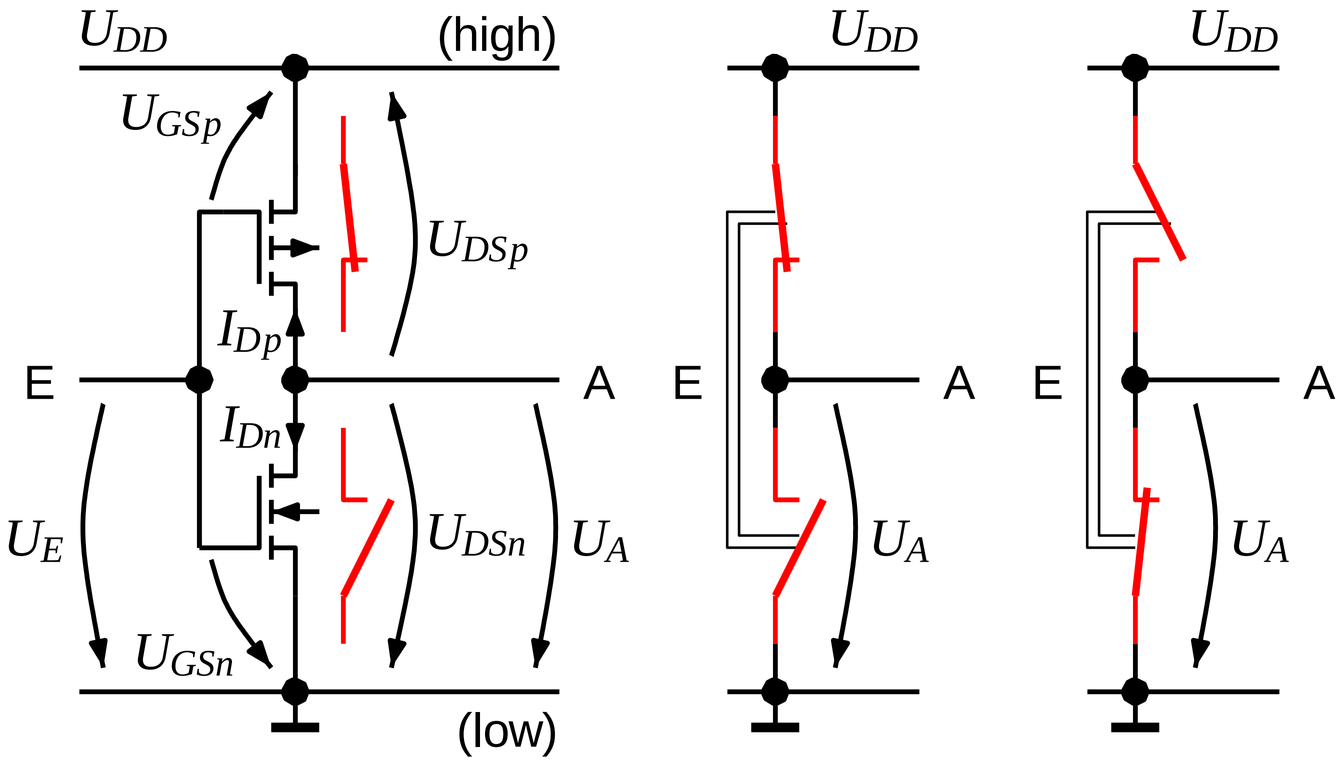
Durchlassbereich : $|U_{GS}| \geq |U_{Th}|$ $|I_D| > 0$ (aktiver Bereich)



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

CMOS-Inverter (NOT)

Reihenschaltung von NMOS und PMOS mit gleicher Ansteuerung.



$$U_{GSn} = U_E, U_{GSp} = U_E - U_{DD}$$
$$U_A = U_{DSn} = U_{DD} + U_{DSp}, I_{Dp} = -I_{Dn}$$

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

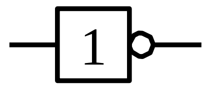
$U_E = U_{DD}$
$U_A = 0$

H - high U_{DD}
L - low 0

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

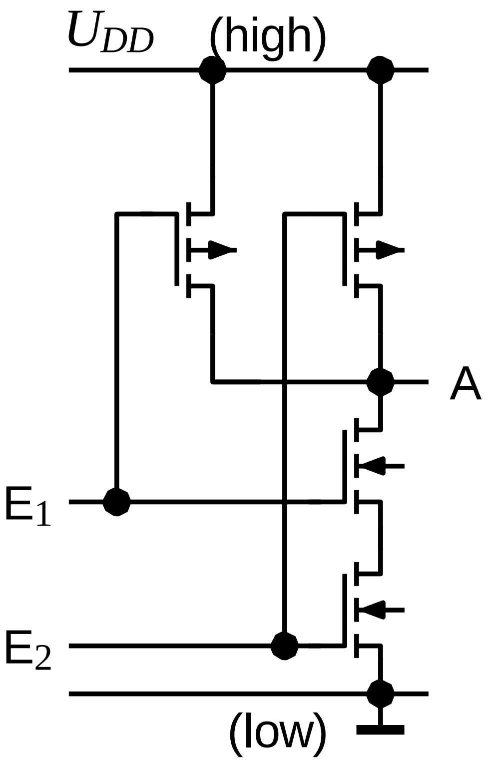
E	A
L	H
H	L

Schaltsymbol



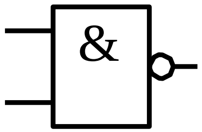
NOT

CMOS-NAND, CMOS-NOR Verknüpfungsglieder

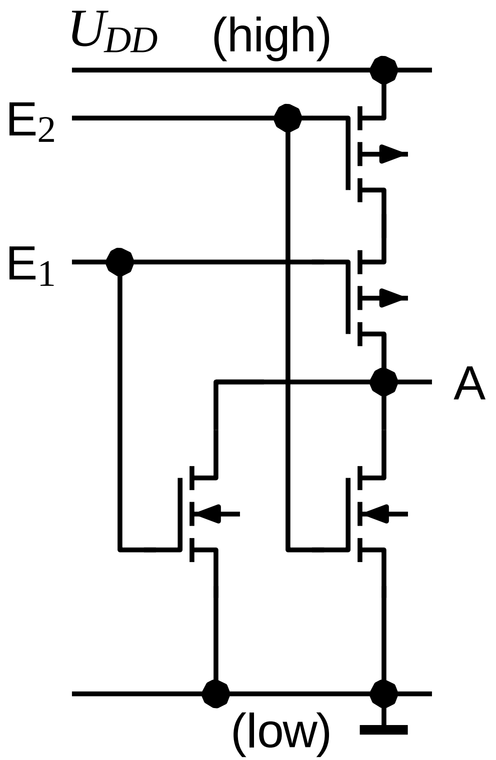


E_1	E_2	A
L	L	H
L	H	H
H	L	H
H	H	L

Schaltsymbol

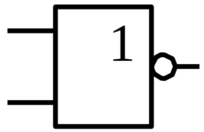


NAND



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

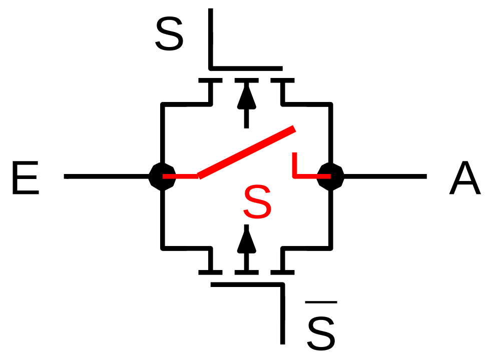
Schaltsymbol



NOR

CMOS-Transmission Gate (Transferschalter)

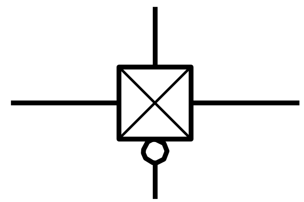
Parallelschaltung von NMOS und PMOS mit komplementärer Ansteuerung.

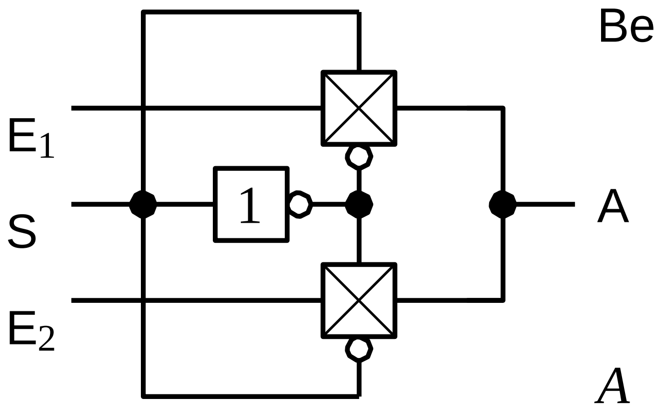


E	S	A
L	L	Z
L	H	L
H	L	Z
H	H	H

H - high U_{DD}
L - low 0
Z - hochohmig
offener Ausgang

Schaltsymbol

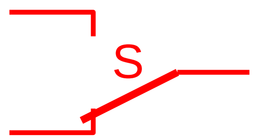




Beispiel 2:1 Multiplexer

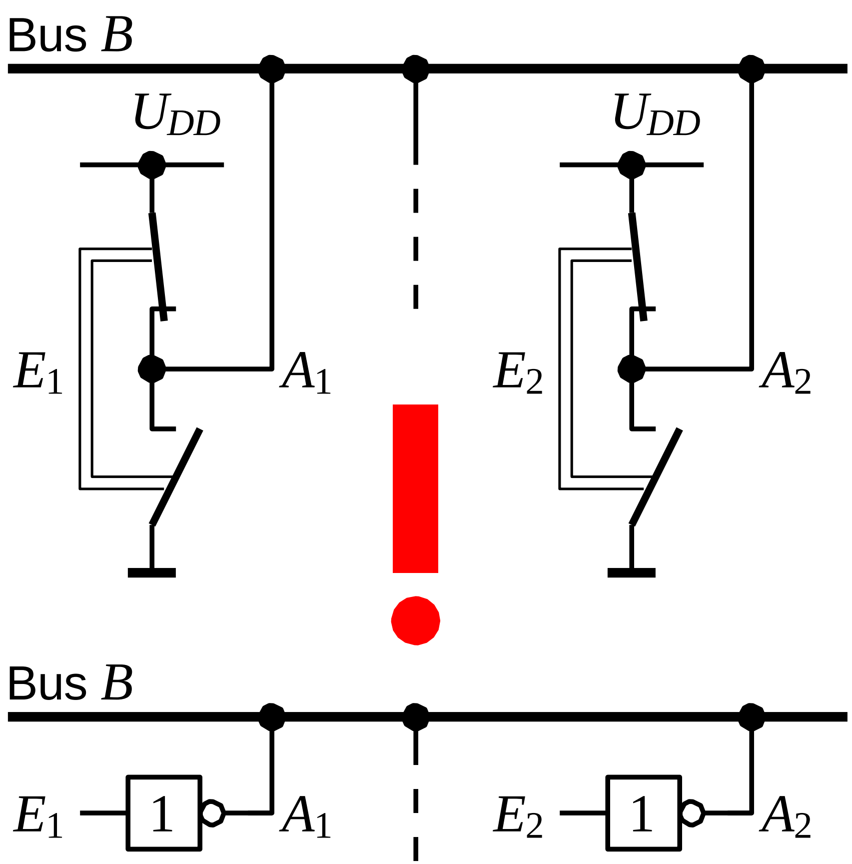
S	A
L	E_2
H	E_1

$A \qquad E_1 \quad S \qquad E_2 \quad \overline{S}$



Umschalter

Ausgangsseitige Verknüpfung



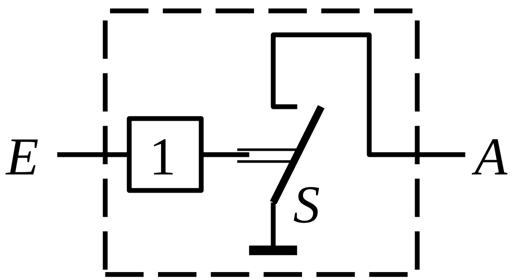
- Kommunikation über eine Leitung
- Anschluß vieler Komponenten
- alle dürfen lesen
- nur einer darf schreiben

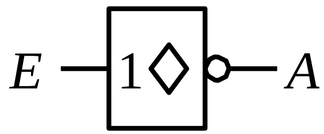
- direkte Ausgangsverknüpfung von CMOS nicht zulässig (Kurzschluß !)

E_1	E_2	B
L	L	H
L	H	Kurzschluß !!!
H	L	
H	H	L
B	A_1	A_2

Open Collector, Open Drain, Tristate-Logik

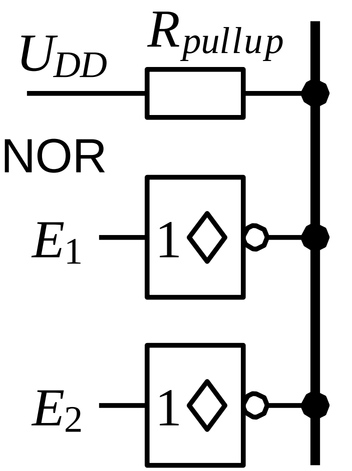
Open Collector / Open Drain





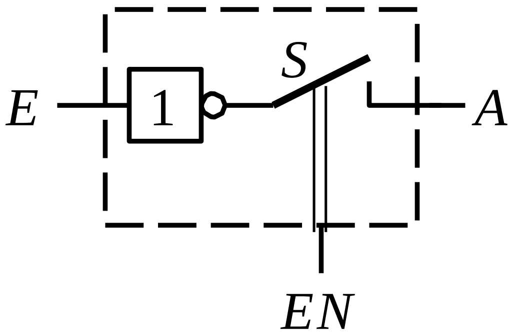
E	A
L	Z
H	L

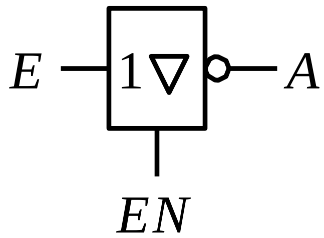
Wired NOR



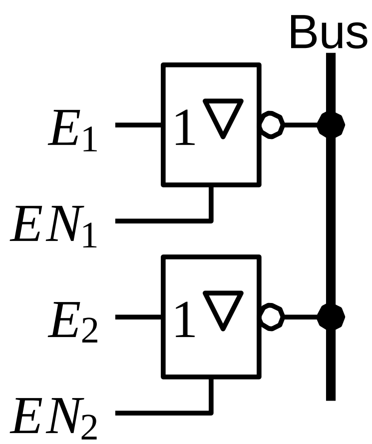
Z - Tristate (hochohmig)

Tristate-Logik





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



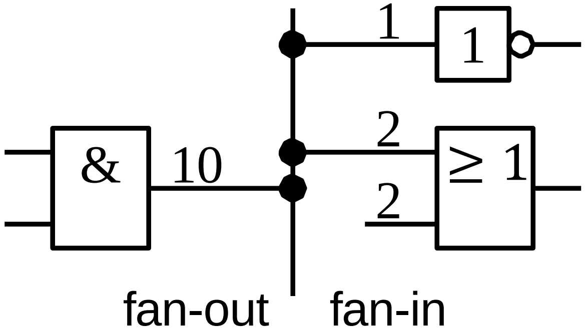
EN_1 EN_2 H nicht erlaubt !

Ausgangsbelastbarkeit

Eingangsäquivalent (Bezugsgröße)
Lastverhalten eines Standardeinganges einer Schaltkreisfamilie, normiert auf das NOT-Gate der Familie.

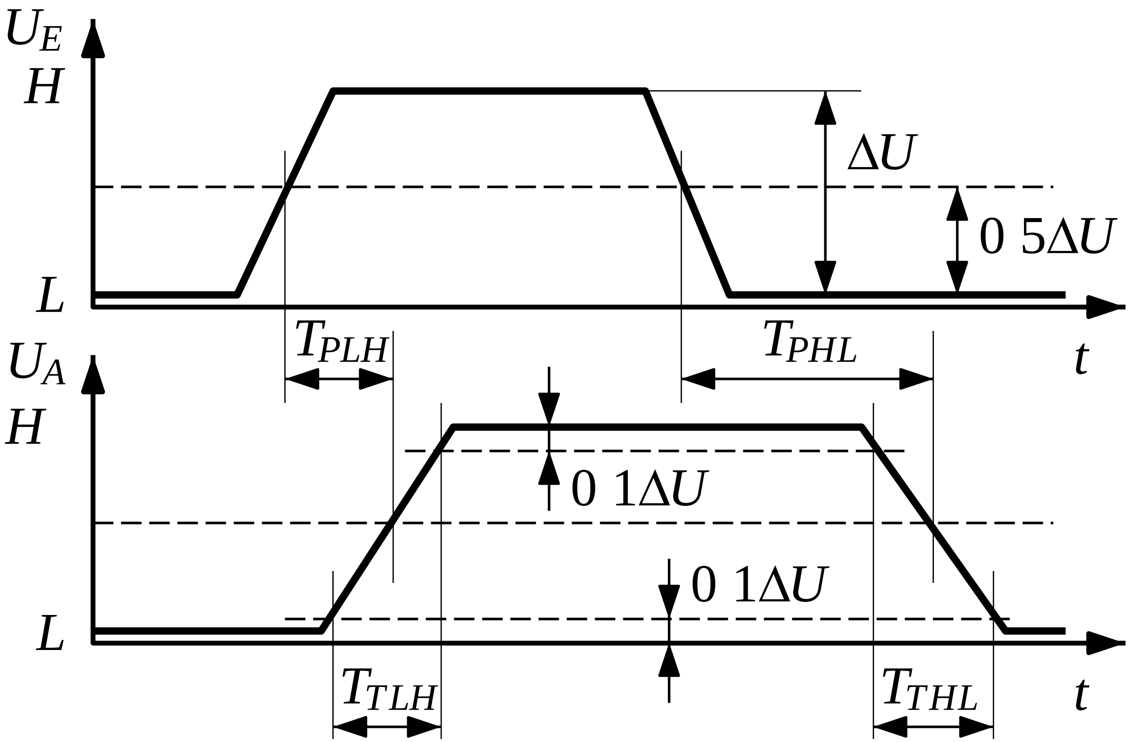
Eingangslastfaktor, normiert (Einfächerung, fan-in)
Lastverhalten eines Einganges bezogen auf das Eingangsäquivalent (ganz).

Ausgangslastfaktor, normiert (Ausfächerung, fan-out)
Mit wievielen Eingangsäquivalenten (Summe aller Einfächerungen) darf ein Ausgang maximal belastet werden, ohne die Pegel zu gefährden.



Zeitverhalten von Verknüpfungsgliedern

Lineare Näherung des Übertragungsverhaltens (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

Schaltalgebra

Definitionen der Schaltalgebra

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

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

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

Meistens wird positive Logik verwendet.

Schaltvariablen sind Symbole für Elemente der Schaltalgebra aus der Menge B :
$x \in B$ mit $B = \{0,1\} \Rightarrow x$ kann nur die Werte 0 oder 1 annehmen.

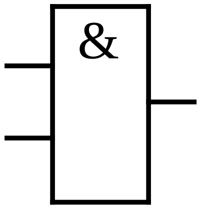
Schaltalgebra

2. Es existieren Operatoren mit folgender Zuordnung:

Operator	$\wedge$	$\vee$	/
Boolesche Verknüpfung	AND	OR	NOT
Schreibweise in Funktionen	$\cdot$	$+$	/

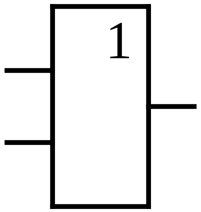
3. Einführung von Schaltsymbolen für die Darstellung der Operatoren:

Konjunktion



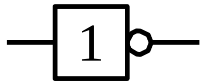
AND

Disjunktion



OR

Negation



NOT

4. Es gelten die Gesetze der Booleschen Algebra.

Gesetze der Booleschen Algebra

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

Boolesch: Eigenschaft von Variablen o. Funktionen, binär (zweiwertig) zu sein.

Es existiert eine Menge $B = \{a, b, c, \dots, e, n\}$ und die eindeut. Operatoren: ∧,∨,/.

Bindung im Ausdruck nimmt von / nach ∧ und nach ∨ ab.

e - neutrales Element bzgl. der Operation ∧ (Eins-Element)
n - neutrales Element bzgl. der Operation ∨ (Null-Element)

Gesetze der Booleschen Algebra 1

Kommutativgesetze

$$a \circ b = b \circ a$$

Assoziativgesetze

$$(a \circ b) \circ c = a \circ (b \circ c)$$

Distributivgesetze

$$a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c) \qquad a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c)$$

Absorbtionsgesetze

$$a \wedge (a \vee b) = a \qquad a \vee (a \wedge b) = a$$

Neutrale Elemente

$$a \wedge e = a \qquad e = 1, n = 0, e = \bar{n} \qquad a \vee n = a$$

Tautologie

$$a \circ a = a$$

Komplementäres Element

$$a \wedge \bar{a} = n \qquad \bar{\bar{a}} = \overline{(\bar{a})} = a \qquad a \vee \bar{a} = e$$

Gesetze der Booleschen Algebra 2

Dualitätsprinzip

Ist A eine Aussage der Booleschen Algebra (Boolesche Funktion), so auch die Aussage $\bar{A}$, die man durch Vertauschen von ∧ gegen ∨ sowie n gegen e erhält.

De Morgansche Regeln

$\overline{a \wedge b} = \bar{a} \vee \bar{b}$ $\qquad$ $\overline{a \vee b} = \bar{a} \wedge \bar{b}$

Schaltfunktion

Eine Schaltfunktion ordnet den Wertekombinationen der Schaltvariablen $(x_1, \dots, x_n) \in D, x_1, \dots, x_n \in B \; mit \; D \subset B^n, B = \{0,1\}$ eindeutig einen Funktionswert y entsprechend der Zuordnungsvorschrift f zu:

$f: D \to B$ oder $y = f(x_1, \dots, x_n)$ mit $D \subset B^n$ und $y, x_1, \dots, x_n \in B, B = \{0,1\}$.

y ist eine Funktion von $x_1, \dots, x_n$ mit dem Funktionswert $f(x_1, \dots, x_n) \in B$ und dem Definitionsbereich $D \subset B^n$ mit $B = \{0,1\}$ (eine Abbildung von D in B).

Bei n Schaltvariablen enthält die Produktmenge (kartesisches Kreuzprodukt) B^n genau 2^n verschiedene Elemente, Wertekombinationen dieser Schaltvariablen.

Jeder dieser Wertekombinationen können zwei verschiedene Funktionswerte zugeordnet werden (0 oder 1). Damit gibt es genau 2^{2^n} mögliche Schaltfunktionen dieser n Schaltvariablen.

Beispiel: Schaltfunktionen von zwei Schaltvariablen ($n=2$)

Wertekombinationen der Schaltvariablen $x_1, x_2 \in B, (n = 2)$:

$B^2 = \{(00), (01), (10), (11)\} \Longrightarrow 2^2 = 4$ verschiedene Wertekombinationen.

Schaltfunktionen der Schaltvariablen $x_1, x_2 \in B, (n = 2)$:

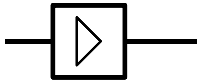
$K(x_1, x_2)$ Schaltfunktionen $y_v = f_v(x_1, x_2), (v = 1, \dots, 16)$

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}

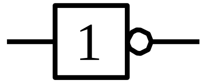
$\Longrightarrow 2^{2^2} = 16$ verschiedene Schaltfunktionen.

Schaltsymbole für Verknüpfungsglieder $(n = 1 \ldots 2)$

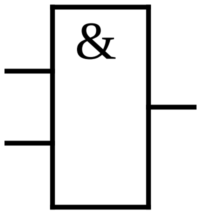
Identität (Buffer)

x  $y = x$

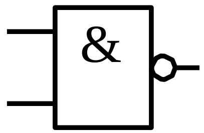
Negation (NOT)

x  $y = \bar{x}$

Konjunktion (AND)

x_1, x_2  $y = x_1 \wedge x_2$

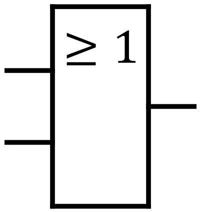
Antikonjunktion (NAND)

x_1, x_2 

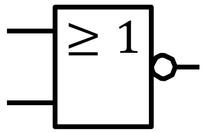
$y = \overline{x_1 \wedge x_2}$

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

Disjunktion (OR)

x_1, x_2  $y = x_1 \vee x_2$

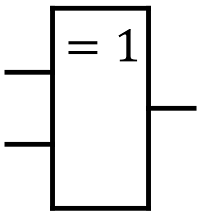
Antidisjunktion (NOR)

x_1, x_2 

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

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

Antivalenz (XOR)

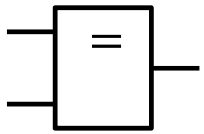
x_1, x_2 

$y = x_1 \not\equiv x_2$

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

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

Äquivalenz (XNOR)

x_1, x_2 

$y = x_1 \equiv x_2$

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

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

Darstellung von Schaltfunktionen

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

Schaltfunktionen Schaltnetze, Netzwerke von Verknüpfungsgliedern.
Signalrückführungen sind nicht zugelassen (Eindeutigkeit, Zeitverhalten).

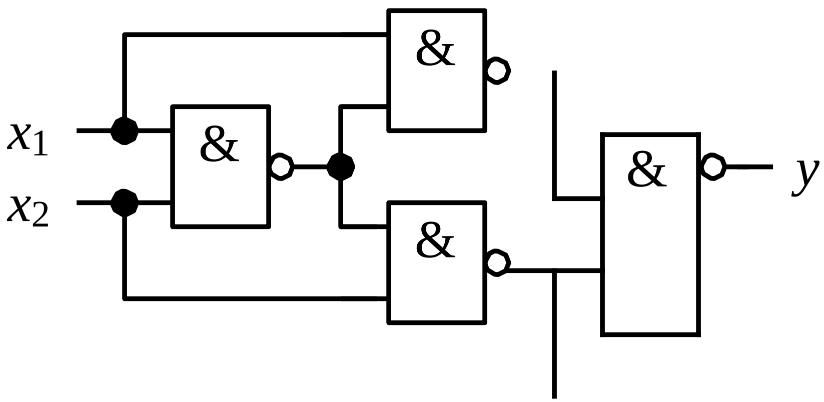
Wertetabelle
(Wahrheitstabelle)

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

Boolesche Gleichung
(Funktionsgleichung)

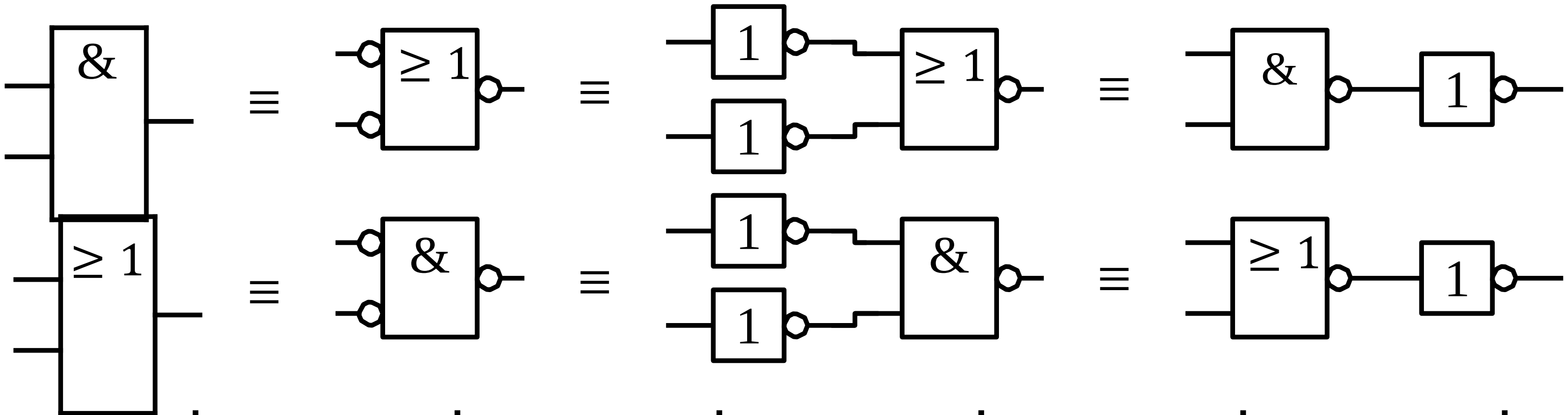
$$y = (x_1 \wedge \overline{x_2}) \vee (\overline{x_1} \wedge x_2)$$

Schaltplan
(Logikplan)



Grundoperationssysteme (Basissysteme)

Jede beliebige Schaltfunktion kann durch die Verknüpfungsglieder AND, OR und NOT realisiert werden. Verknüpfungsglieder mit mehr als zwei Schaltvariablen können auf Verknüpfungen mit zwei (oder einer) Schaltvariablen zurückgeführt werden.



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

Minterme und Maxterme

Minterme

Minterme sind Vollkonjunktionen mit dem Wert 1, UND-Verknüpfungen, die alle Schaltvariablen genau einmal in unnegierter oder negierter Form enthalten.

Jeder Minterm hat nur für eine Kombination der Schaltvariablen (in der Wertetabelle) den Wert 1, sonst 0.

Maxterme

Maxterme sind Volldisjunktionen mit dem Wert 0, ODER-Verknüpfungen, die alle Schaltvariablen genau einmal in unnegierter oder negierter Form enthalten.

Jeder Maxterm hat nur für eine Kombination der Schaltvariablen (in der Wertetabelle) den Wert 0, sonst 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 (Schaltfunktion = 1 in der Wertetabelle).

Kanonisch Konjunktive Normalform (KKNF)

Konjunktion (UND-Verknüpfung) aller Maxterme der Schaltfunktion (Schaltfunktion = 0 in der Wertetabelle).

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 Normalformen

x_1	x_2	x_3	y	Minterme	Maxterme
0	0	0	0		$x_1 \lor x_2 \lor x_3$
0	0	1	1	$\overline{x_1} \land \overline{x_2} \land x_3$	
0	1	0	1	$\overline{x_1} \land x_2 \land \overline{x_3}$	
0	1	1	0		$x_1 \lor \overline{x_2} \lor \overline{x_3}$
1	0	0	1	$x_1 \land \overline{x_2} \land \overline{x_3}$	
1	0	1	0		$\overline{x_1} \lor x_2 \lor \overline{x_3}$
1	1	0	0		$\overline{x_1} \lor \overline{x_2} \lor x_3$
1	1	1	1	$x_1 \land x_2 \land x_3$	

Kurzschreibweise für Schaltfunktionen: $\land \rightarrow \cdot$ und $\lor \rightarrow +$

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

KKNF: $y = (x_1 + x_2 + x_3)(x_1 + \overline{x_2} + \overline{x_3})(\overline{x_1} + x_2 + \overline{x_3})(\overline{x_1} + \overline{x_2} + x_3)$

Don't Care Terme

Nicht für jede Kombination der Schaltvariablen einer Schaltfunktion ist der Funktionswert eindeutig definiert. Er kann entwerder 0 oder 1 sein. Die Wahl ist beliebig, don’t care.

Don’t care Terme werden mit X gekennzeichent. X steht für 0 oder 1.

Don’t care Terme sind bei der Vereinfachung von Schaltfunktionen von besonderer Bedeutung.

In technischen Schaltnetzen kommen don’t care Terme relativ häufig vor.

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 mit den Rechenregeln der Schaltalgebra

Auflösung von Negationen (De Morgansche Regel)

$\overline{ab} = \bar{a} + \bar{b}$ oder $\overline{a + b} = \bar{a}\bar{b}$

Erweitern von Termen ($b + \bar{b} = 1$ **oder** $b\bar{b} = 0$**)**

$a = a\,(b + \bar{b}) = ab + a\bar{b}$ oder $a = a + b\bar{b} = (a+b)(a + \bar{b})$

Kürzen von Termen, Absorbtionsgesetz ($b \quad \bar{b} \quad 1$ **oder** $b\bar{b} \quad 0$**)**

$ab + a\bar{b} = a\,(b + \bar{b}) = a$ oder $(a+b)(a + \bar{b}) = a + (b\bar{b}) = a$

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

Verfahren von Quine-McCluskey

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

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

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.

Karnaugh-Veitch-Diagramm (KV-Diagramm)

Das KV-Diagramm stellt eine Schaltfunktion in KDNF zweidimensional als Matrix dar und erlaubt so eine einfache grafische Vereinfachung dieser Schaltfunktion. Es ist nur für die Vereinfachung von Schaltfunktionen bis maximal 4 (6) Schaltvariablen sinnvoll geeignet.

Konstruktion des KV-Diagramms

Bei n Schaltvariablen enthält die Matrix 2^n Felder. Jedes Feld entspricht genau einem Minterm oder Maxterm, gekennzeichnet durch Zeile und Spalte.

Aufteilung der Schaltvariablen an den Köpfen der Zeilen und Spalten der Matrix.

Benachbarte Zeilen oder Spalten dürfen sich nur in der Belegung einer einzigen Schaltvariablen unterscheiden (unnegiert, negiert).

Erste und letzte Zeile bzw. Spalte der Matrix sind ebenfalls benachbart.

Konstruktion des KV-Diagramms

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

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: $\bar{a}bcd$

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

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

$y = f(abc)$

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

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

$y = f(ab)$

Minterm: ab

Vereinfachung von Schaltfunktionen im KV-Diagramm

Benachbarte "1" (Minterme) im KV-Diagramm können zu einem Block (2er-Block) zusammengefasst werden (nur in Zeilen oder Spalten direkt benachbart).

Benachbarte 2er-Blöcke können zu einem 4er-Block zusammengefasst werden und 4er-Blöcke entsprechend zu einem 8er-Block usw.

Es ist nur die Bildung von 1er-, 2er-, 4er-, 8er-,. . . Blöcken zulässig. Die Blöcke dürfen sich beliebig überlagern.

Zur Minimierung einer Schaltfunktion sind möglichst wenig und möglichst große Blöcke zu bilden. Es müssen alle Minterme der Schaltfunktion umfaßt werden.

Jeder Block wird durch einen Term beschrieben, der nur noch die Variablen enthält, die in allen Mintermen, die der Block umfaßt, identisch sind.

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

$y = \bar a b \bar c_1 + \bar b c d_2 + ac_3$

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, alternative Beschriftung des KV-Diagramms

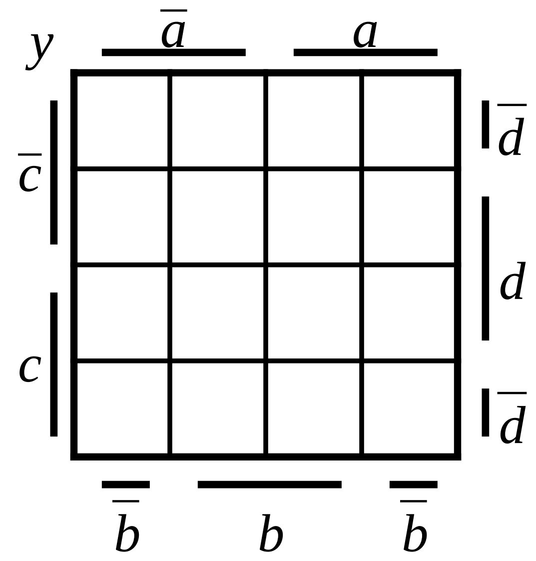
Don’t Care Terme X werden wie entweder 0 oder 1 behandelt. Die Wahl ob als 0 oder als 1 erfolgt entsprechend der Vereinfachungsmöglichkeit. Ist eine 1 für die Blockbildung günstiger, so wird X wie eine 1 verwendet, ansonten wie eine 0.

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

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

don’t care Terme

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



alternative Beschriftungsarten des KV-Diagramms

Realisierung als Schaltnetz

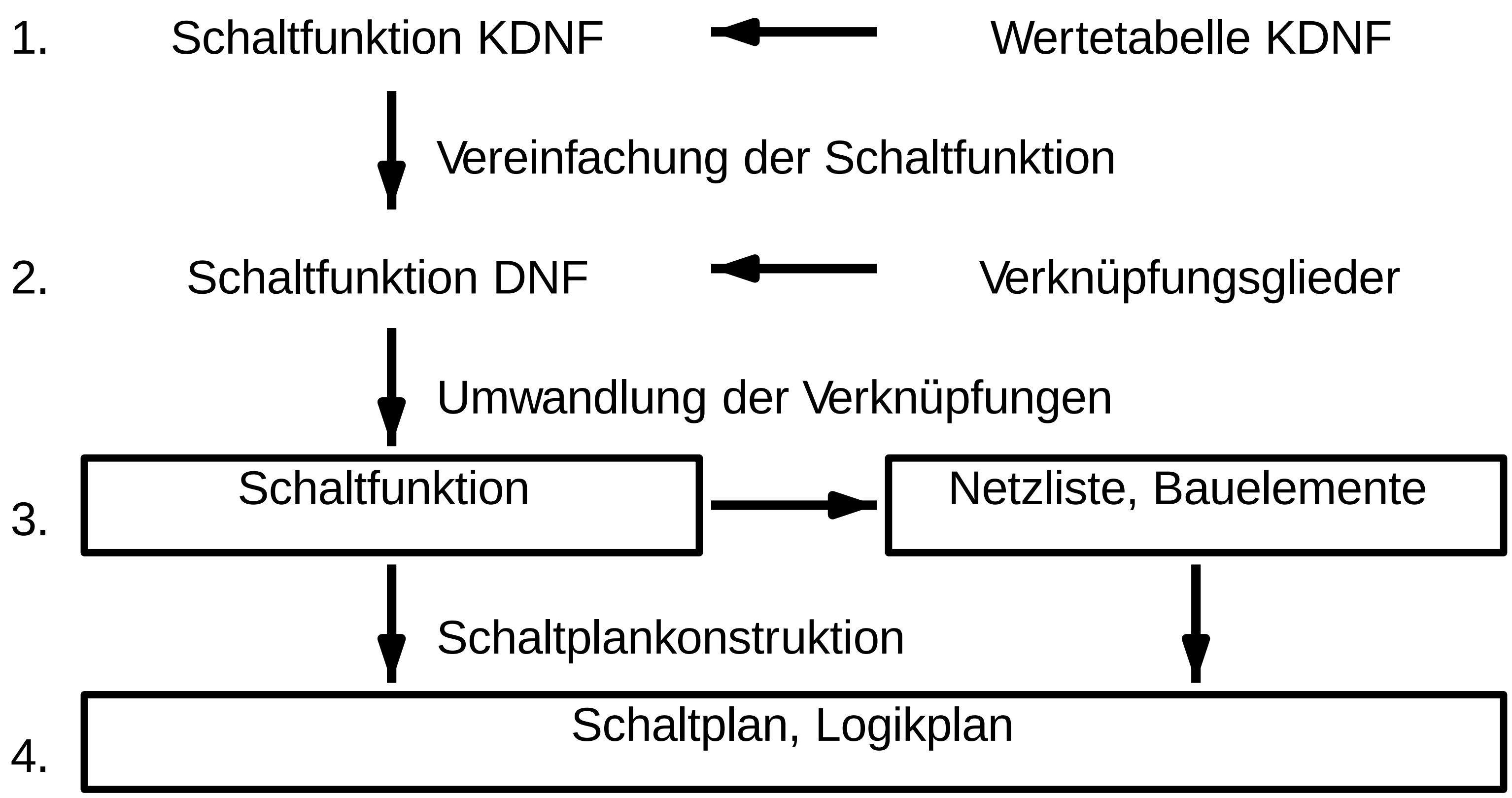
Schaltfunktionen können durch ein Netzwerk von Verknüpfungsgliedern realisiert werden. Diese Realisierungsform wird als Schaltnetz bezeichnet.

Die Schaltfunktion wird nach Vereinfachung so umgeformt, dass nur noch Verknüpfungen Verwendung finden, die den vorgegebenen Verknüpfungsgliedern entsprechen.

Entsprechend der so umgeformten Schaltfunktion kann ein Schaltplan konstruiert werden oder auch eine entsprechende Netzliste (Verdrahtungsplan) erstellt werden.

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

Synthese eines Schaltplanes aus eine Schaltfunktion



Beispiel zur Schaltplansynthese

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

1. Schaltfunktion in KDNF:

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

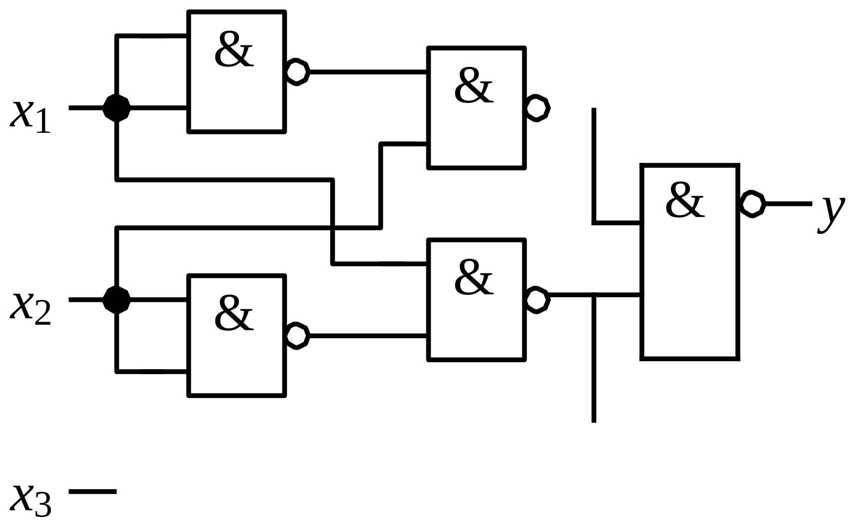
2. Vereinfachte Schaltfunktion in DNF:

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

3. Umgeformte Schaltfunktion unter Einbeziehung der Verknüpfungsglieder:

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

4. Schaltnetz mit 5 NAND:



Analyse von Schaltnetzen

Die Analyse von Schaltplänen 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 mit den Eingängen als Schaltfunktion aufgestellt.
>
> Das System aller so aufgestellten Verknüpfungsfunktionen wird nach der gesuchten Schaltfunktion aufgelöst (beginnend von den Eingängen bis hin zu den Ausgängen werden alle Hilfsvariablen schrittweise eleminiert).

Eine Analyse über alle möglichen Belegungen aller Ein- und Ausgänge (über die Wertetabellen) ist ebenfalls möglich.

Analyse einer Schaltfunktion aus einem Schaltplan

1. Schaltplan → Verknüpfungsglieder

   ↓ Hilfsvariablen für Verknüpfungsglieder

2. Schaltfunktionssystem ← Verknüpfungsfunktion

   ↓ Auflösung nach Schaltfunktion

3. Schaltfunktion → Wertetabelle KDNF

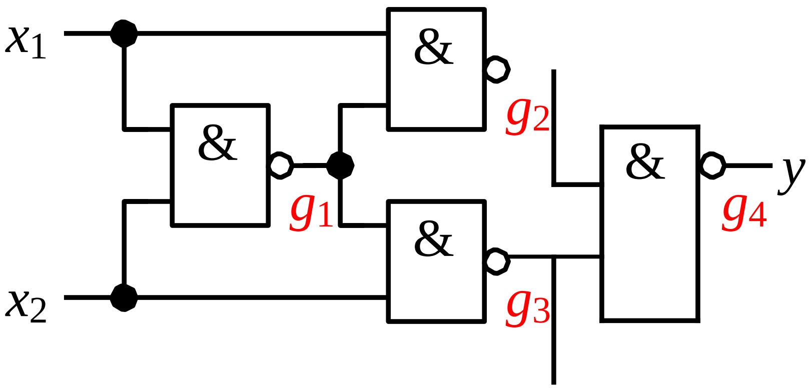
   ↓ Vereinfachung der Schaltfunktion ↓

4. $\boxed{\text{Schaltfunktion in DNF oder KDNF}}$

Beispiel zur Schaltplananalyse

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

1. Schaltnetz mit 4 NAND:



Eingänge, Schaltvariable: x_1, x_2

Ausgänge, Schaltfunktionen: $y = f(x_1, x_2)$

2. Verknüpfungsfunktionen:

$g_1 = \overline{x_1 x_2} \qquad g_2 = \overline{x_1 g_1}$

$g_3 = \overline{x_2 g_1} \qquad g_4 = \overline{x_2 g_3} \qquad y = g_4$

3. Auflösung nach $y = f = (x_1, x_2)$

$g_1 = \overline{x_1 x_2}$

$g_2 = \overline{x_1 \overline{x_1 x_2}} \quad g_3 = \overline{x_2 \overline{x_1 x_2}}$

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

4. Vereinfachung der Schaltfunktion

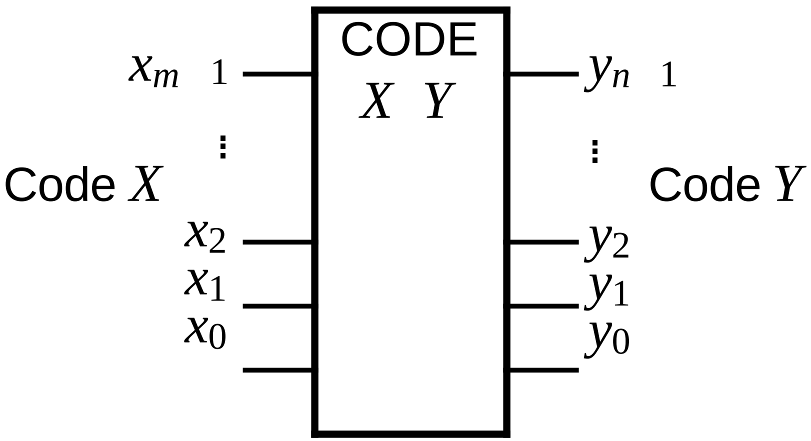
$y = x_1 \overline{x_2} + \overline{x_1} x_2$ (XOR)

Beispielschaltnetze

Encoder / Decoder / Codec

Encoder sind Schaltnetze, die Schaltvariablen eines Codes (Menge von Eingangsvariablen X) in Variablen eines anderen Codes (Menge von Ausgangsvariablen Y) eindeutig umsetzen (Kodierer, Kodeumsetzer, Vektor-Schaltfunktion). Encoder sind zuordnende Schaltnetze.

Blockschaltbild:

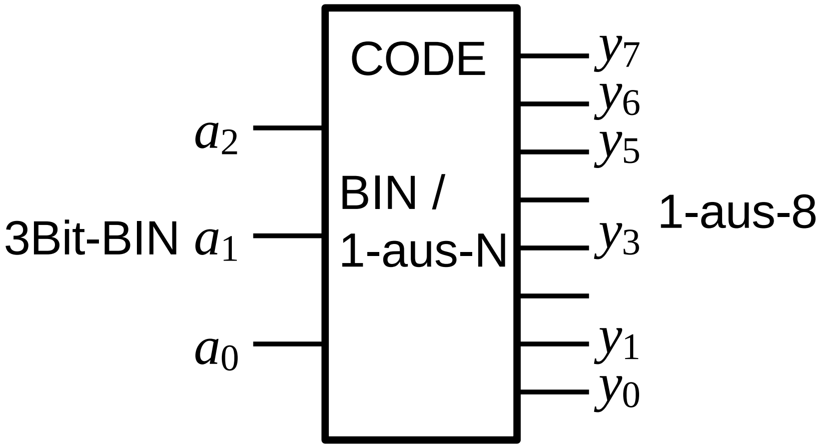


Beispiele für Kodes:

BIN	Binärkode
DEC	Dezimalkode
BCD	Binär kodierte Dezimalzahlen
GRAY	Gray-Kode
1-aus-N	1-aus-N Kode

Beispiel 3Bit-Adressdecoder

Der Aressdecoder kodiert binäre Adressen in eine 1-aus-N Kodierung. Bei diesem Encoding ist nur ein Ausgang entsprechend dem Eingangsbinäräquivalent mit 1 belegt, alle anderen mit 0.
Bei m-Bit Adressen gibt es 2^m Ausgänge.



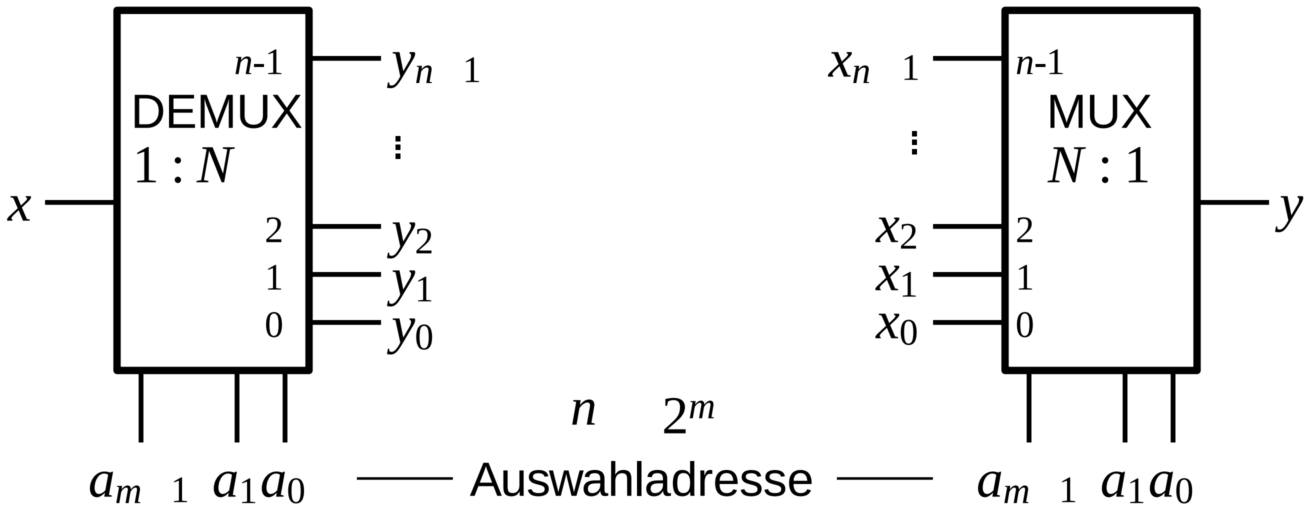
$a_2a_1a_0$	$y_7y_6y_5y_4y_3y_2y_1y_0$
0 0 0	0 0 0 0 0 0 0 1
0 0 1	0 0 0 0 0 0 1 0
0 1 0	0 0 0 0 0 1 0 0
0 1 1	0 0 0 0 1 0 0 0
1 0 0	0 0 0 1 0 0 0 0
1 0 1	0 0 1 0 0 0 0 0
1 1 0	0 1 0 0 0 0 0 0
1 1 1	1 0 0 0 0 0 0 0

$y_0 = \overline{a_2}\,\overline{a_1}\,\overline{a_0}$ $y_1 = \overline{a_2}\,\overline{a_1}a_0$ $y_2 = \overline{a_2}a_1\overline{a_0}$ $y_3 = \overline{a_2}a_1a_0$

$y_4 = a_2\overline{a_1}\,\overline{a_0}$ $y_5 = a_2\overline{a_1}a_0$ $y_6 = a_2a_1\overline{a_0}$ $y_3 = a_2a_1a_0$

Multiplexer

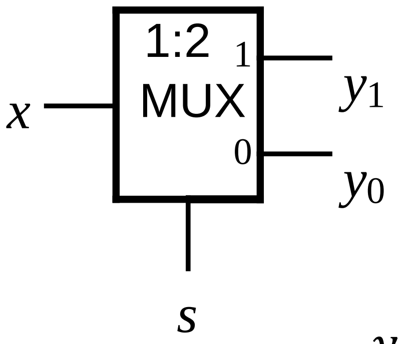
Ein Multiplexer schaltet entsprechend einer binär kodierten Steueradresse den Eingang auf den binäräquivalenten Ausgang (Demultiplexer) durch oder umgekehrt, den binäräquivalenten Eingang auf den Ausgang. Multiplexer sind auswählende Schaltnetze.



Beispiel 1:2 Multiplexer, 2:1 Multiplexer

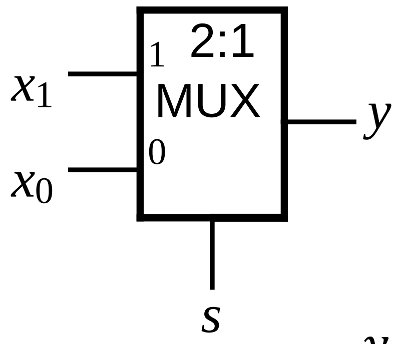
1:2 Multiplexer bzw. 2:1 Multiplexer (Demultiplexer) dienen zur Umschaltung von Datenleitungen.

Zusammengefaßt als Multiplexerbäume können mit ihnen komplexe Schaltnetze realisiert werden.



s	$y_1 y_0$
0	0 x
1	x 0

$$y_0 = sx$$
$$y_1 = \bar{s}x$$



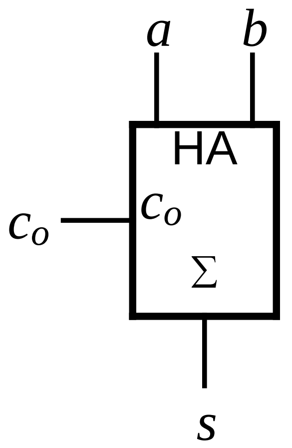
s	y
0	x_0
1	x_1

$$y_0 = \bar{s}x_0 + sx_1$$

Addierer

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

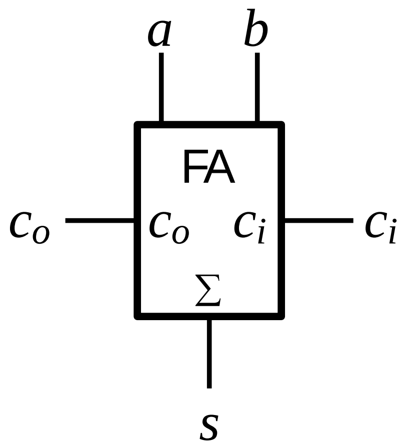
Half adder



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

$$s = a\bar{b} + \bar{a}b$$
$$c_0 = ab$$

Full adder

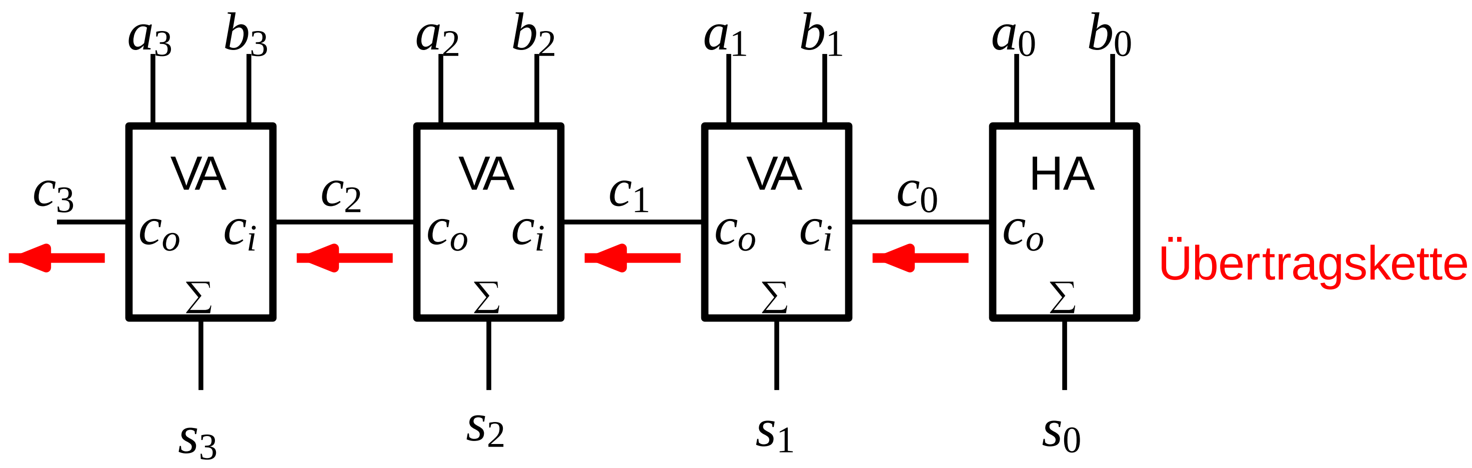


a	b	c_i	s	c_o
0	0	0	0	0
0	1	0	1	0
1	0	0	1	0
1	1	0	0	1
0	0	1	1	0
1	0	1	0	1
1	1	1	1	1

$$s = a\bar{b}\bar{c_i} + \bar{a}b\bar{c_1} + \bar{a}\bar{b}c_i + abc_i$$
$$c_0 = ab + ac_i + bc_i$$

Beispiel 4Bit Paralleladdierer (Ripple-Carry Adder)

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.

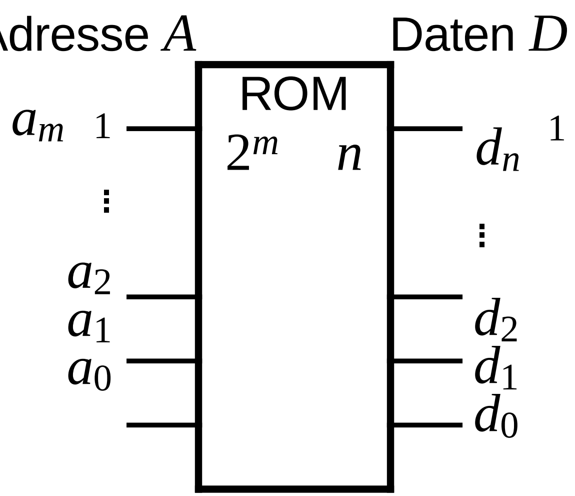


$$A = a_3 \cdot 2^3 + a_2 \cdot 2^2 + a_1 \cdot 2^1 + a_0 \cdot 2^0 \qquad B = b_3 \cdot 2^3 + b_2 \cdot 2^2 + b_1 \cdot 2^1 + b_0 \cdot 2^0$$

$$S = A + B = c_3 \cdot 2^4 + s_3 \cdot 2^3 + s_2 \cdot 2^2 + s_1 \cdot 2^1 + s_0 \cdot 2^0$$

Festwertspeicher (ROM)

ROM (Read Only Memory) sind addressierbare Festwertspeicher, die vorwiegend in der Computertechnik eingesetzt werden. Einer binären kodierten Adresse werden fest eingespeicherte (in Hardware) Daten zugewiesen. ROM sind wert-zuweisende Schaltnetze.



Speicherkapazität: 2^m m Bit

Wertetabelle des ROM

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	$d_{n-1,2^m}$	...	$d_{2,2^m}$	$d_{1,2^m}$	$d_{0,2^m}$

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