Dr.-Ing. Oliver Knodel

Digitale Logik und Schaltwerke

Dresden // April, 2024

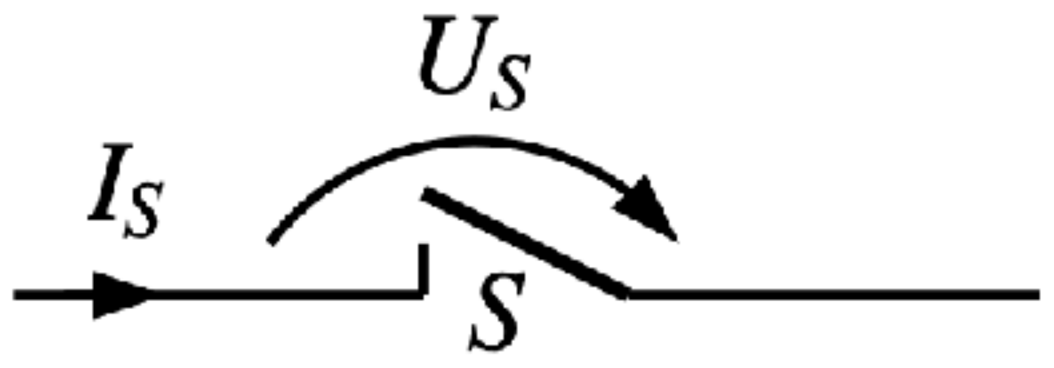
Einfache Schalter

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

$\Rightarrow$ Zentrale Bedeutung der Schalter bei logischen Verknüpfungen.

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

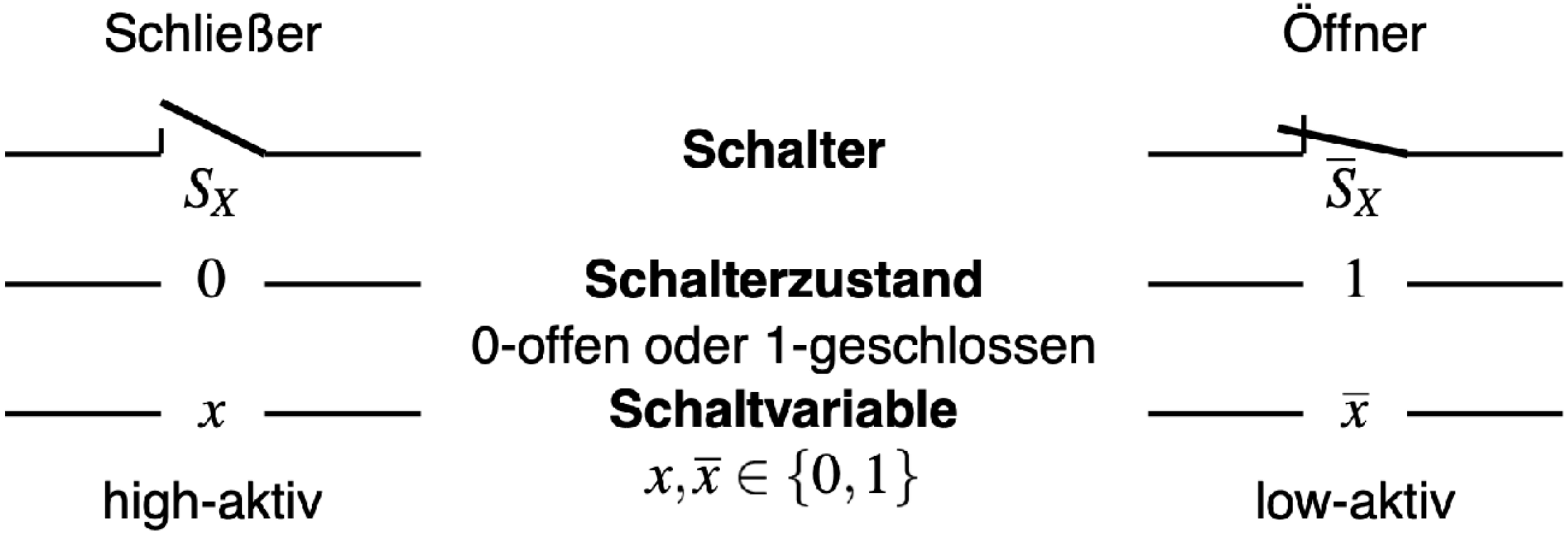
In der Computertechnik dominieren eindeutig elektrisch-elektronische Schalter.



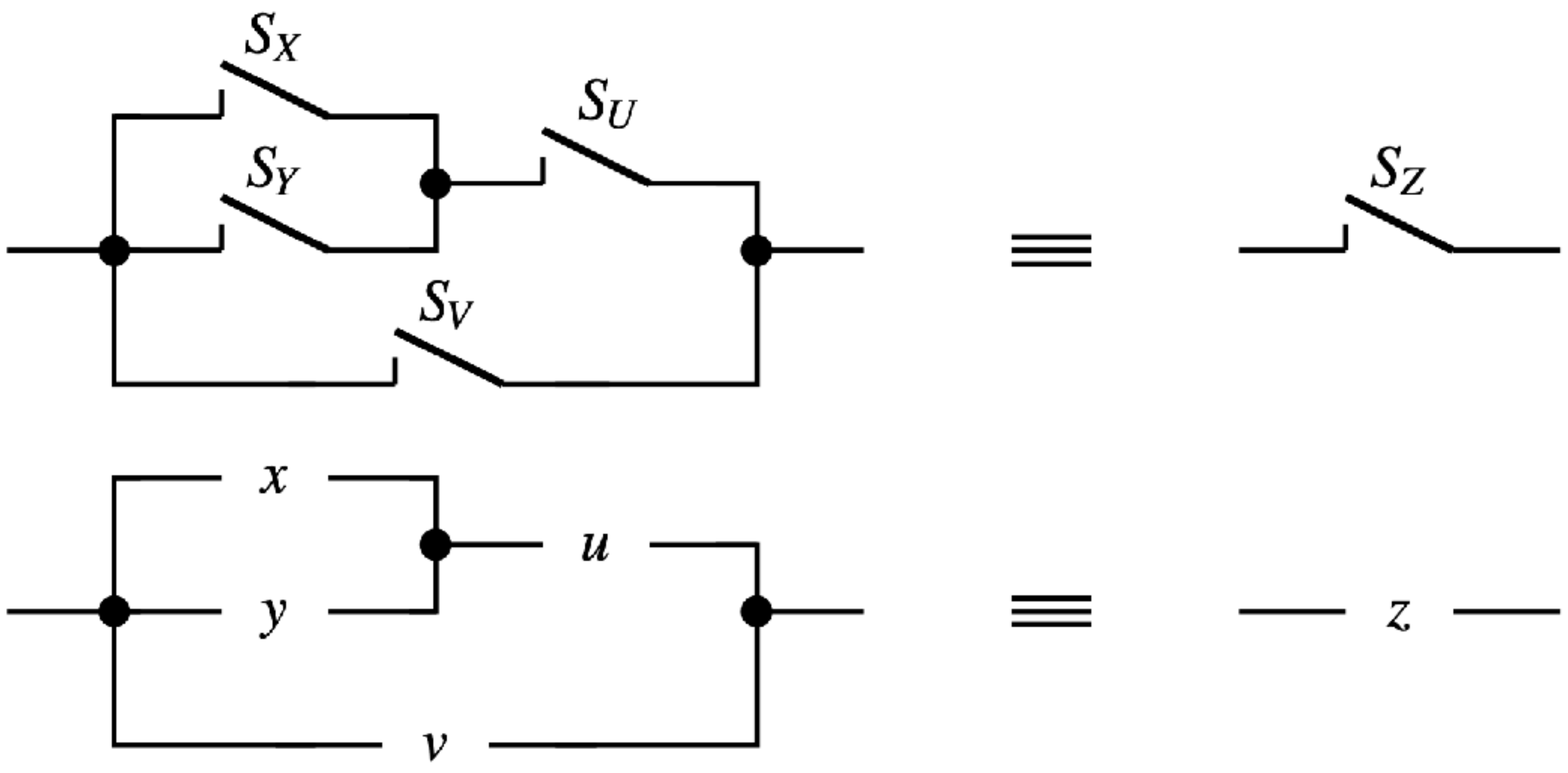
Schalter offen : $I_S = 0$ bei $U_S \neq 0 \;\Rightarrow\; S = 0$

Schalter geschlossen : $U_S = 0$ bei $I_S \neq 0 \;\Rightarrow\; S = 1$

Schalterzustände in der digitalen Logik

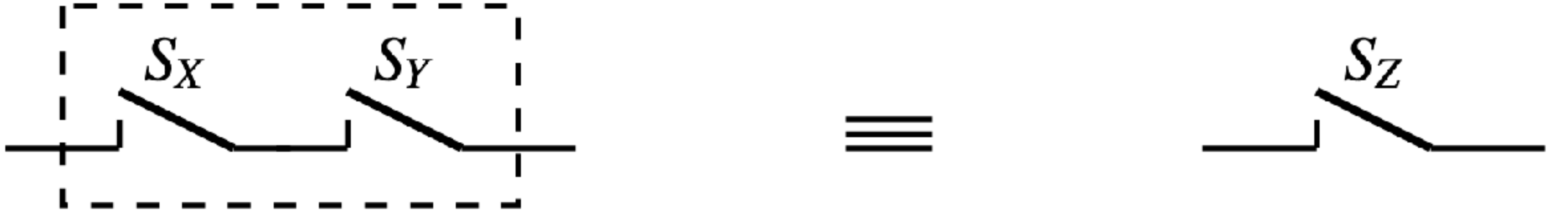


Komplexe Netzwerke von Schaltern



x	y	v	u		$z = v \vee (u \wedge (x \vee y))$
0	0	0	0		0
⋮	⋮	⋮	⋮	16 Zeilen	⋮
1	1	1	1		1

Reihenschaltung

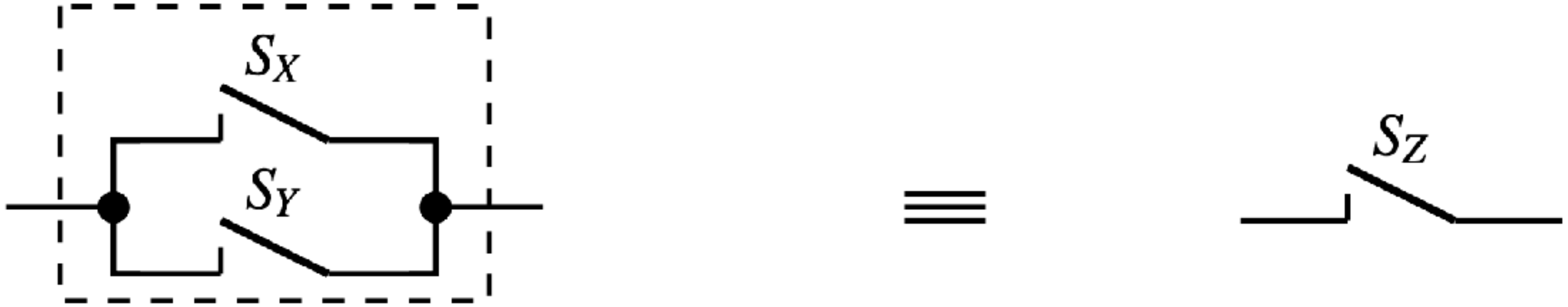


S_Z ist dann und nur dann geschlossen, wenn S_X **und** ($\wedge$) S_Y geschlossen sind.

— x — y — $\equiv$ — z —

x	y	$z = x \wedge y$
0	0	0
0	1	0
1	0	0
1	1	1

Parallelschaltung



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

x	y	$z = x \vee y$
0	0	0
0	1	1
1	0	1
1	1	1

Zusammenfassung Grundfunktionen für Schalter

NICHT

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

ODER

x	y	$z = x \vee y$
0	0	0
0	1	1
1	0	1
1	1	1

UND

x	y	$z = x \wedge y$
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).

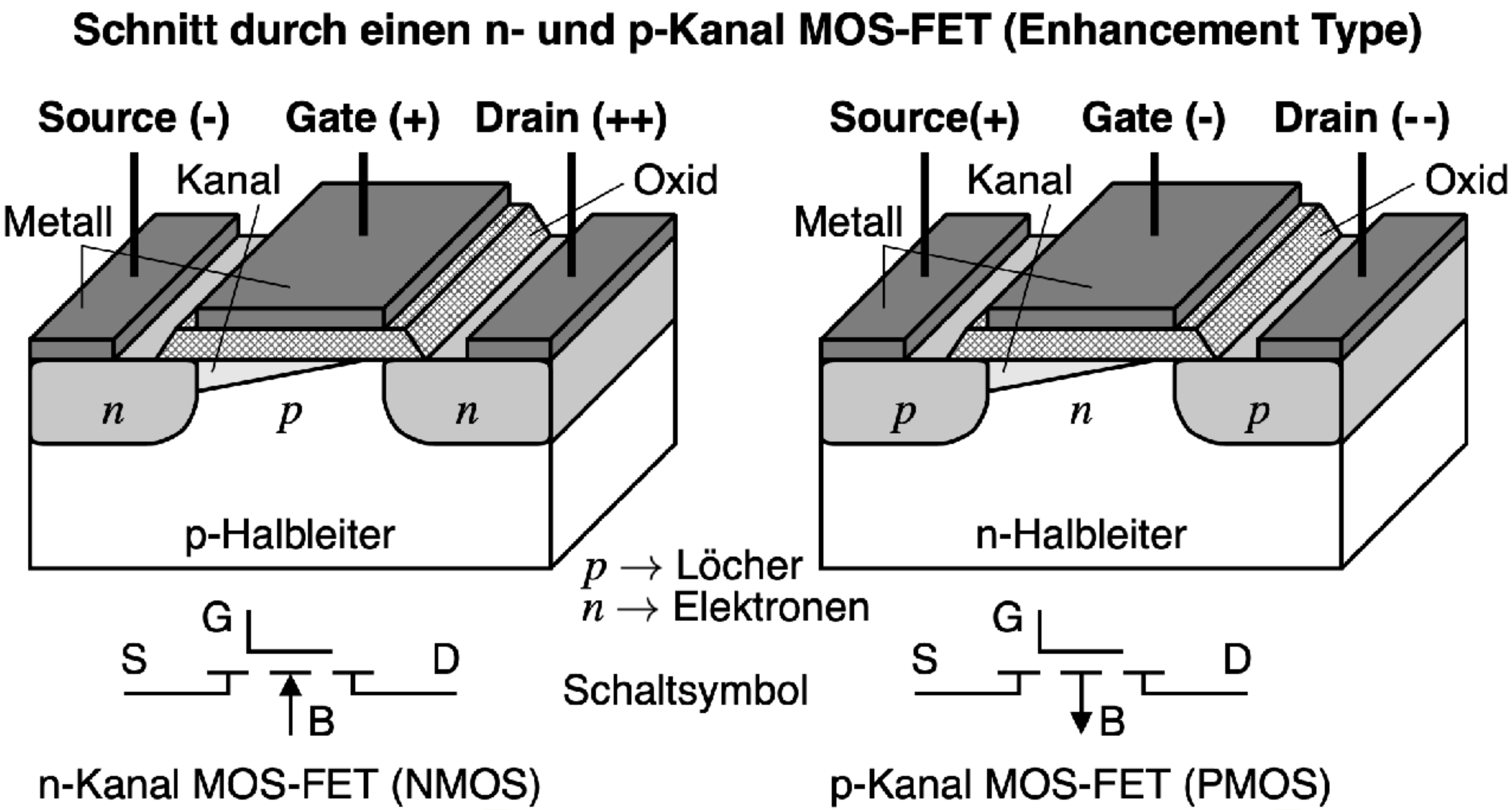
Transistor $\rightarrow$ **Trans** - Res**istor** (steuerbarer Widerstand)

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

Vorteile der CMOS-Technik (Comlementary MOS):

- kein statischer Querstrom und Ansteuerstrom, keine statische Verlustleistung,
- symmetrisches Schaltverhalten, ideale statische Pegel,
- geringe Abmaße, sehr gut geeignet für die Höchstintegration in Silizium.

Aufbau und Wirkungsweise von MOS-FET

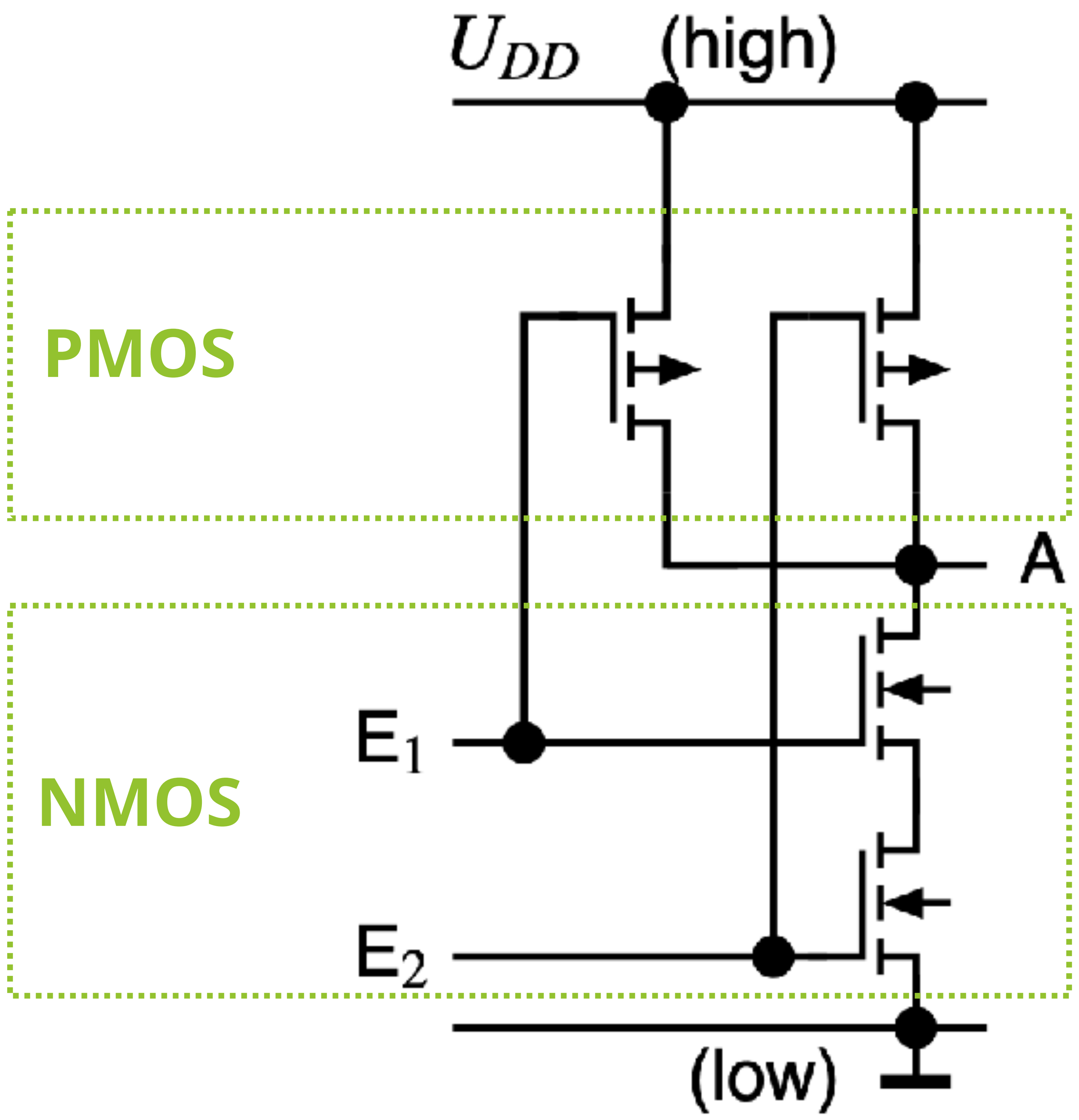


G = High
Transistor (B) = leitet

G = High
Transistor (B) = sperrt

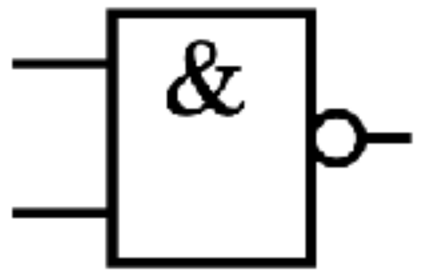
CMOS-NAND, CMOS-NOR

- Signal am Ausgang A liegt bei CMOS **invertiert** vor
- **Kein** Energieverlust durch Lastwiderstand (Energieumsatz nur bei Schaltvorgang)
- Transistoren verhalten sich **komplementär** zueinander (Schaltung immer doppelt vorhanden/doppelte Fläche auf Silizium)
- Hohes oder niedriges Potential wird zum Ausgang durchgeleitet

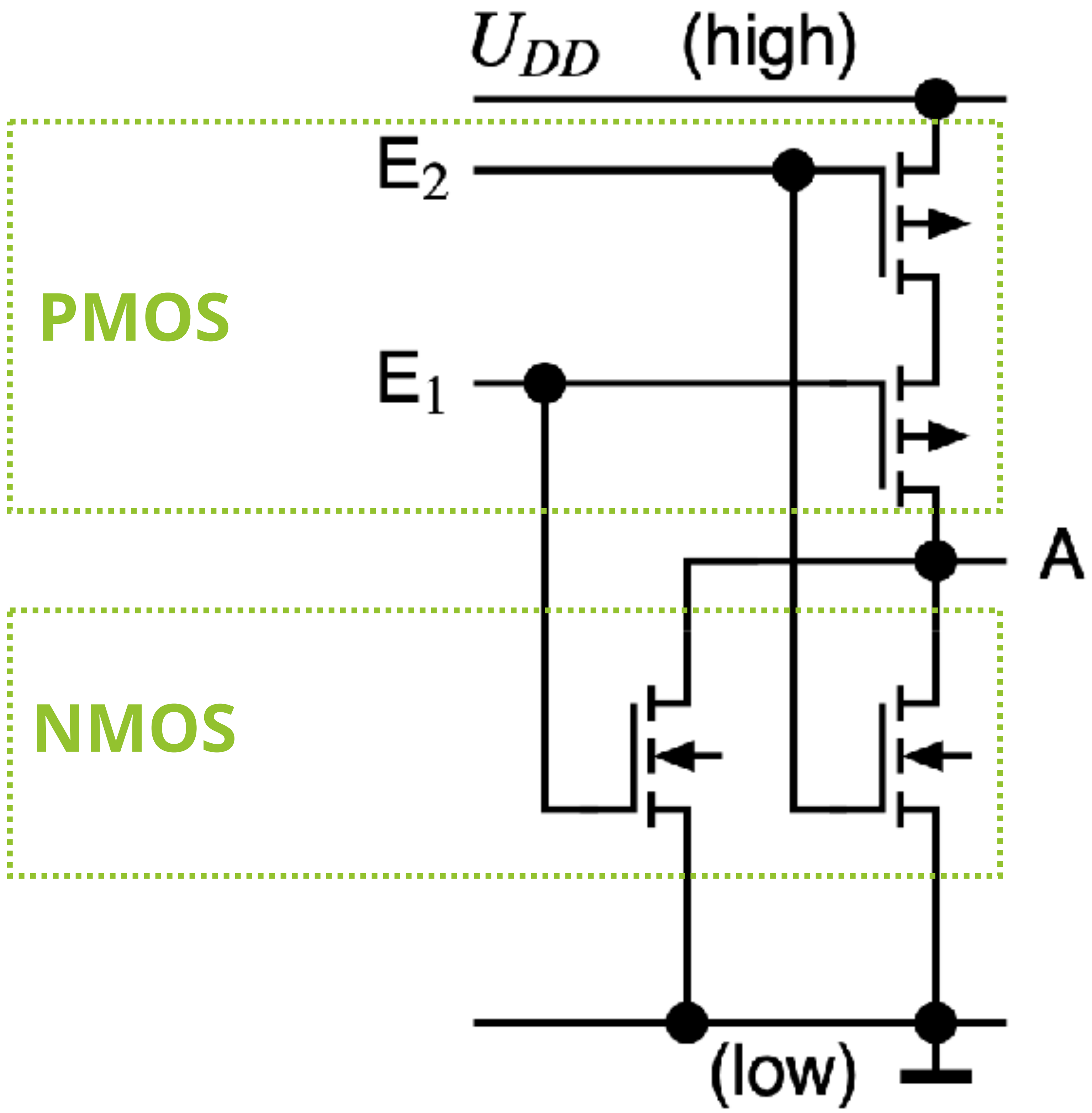


E_1	E_2	A
L	L	H
L	H	H
H	L	H
H	H	L

Schaltsymbol

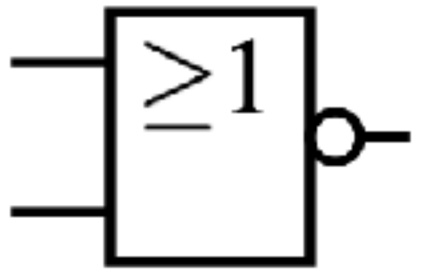


NAND



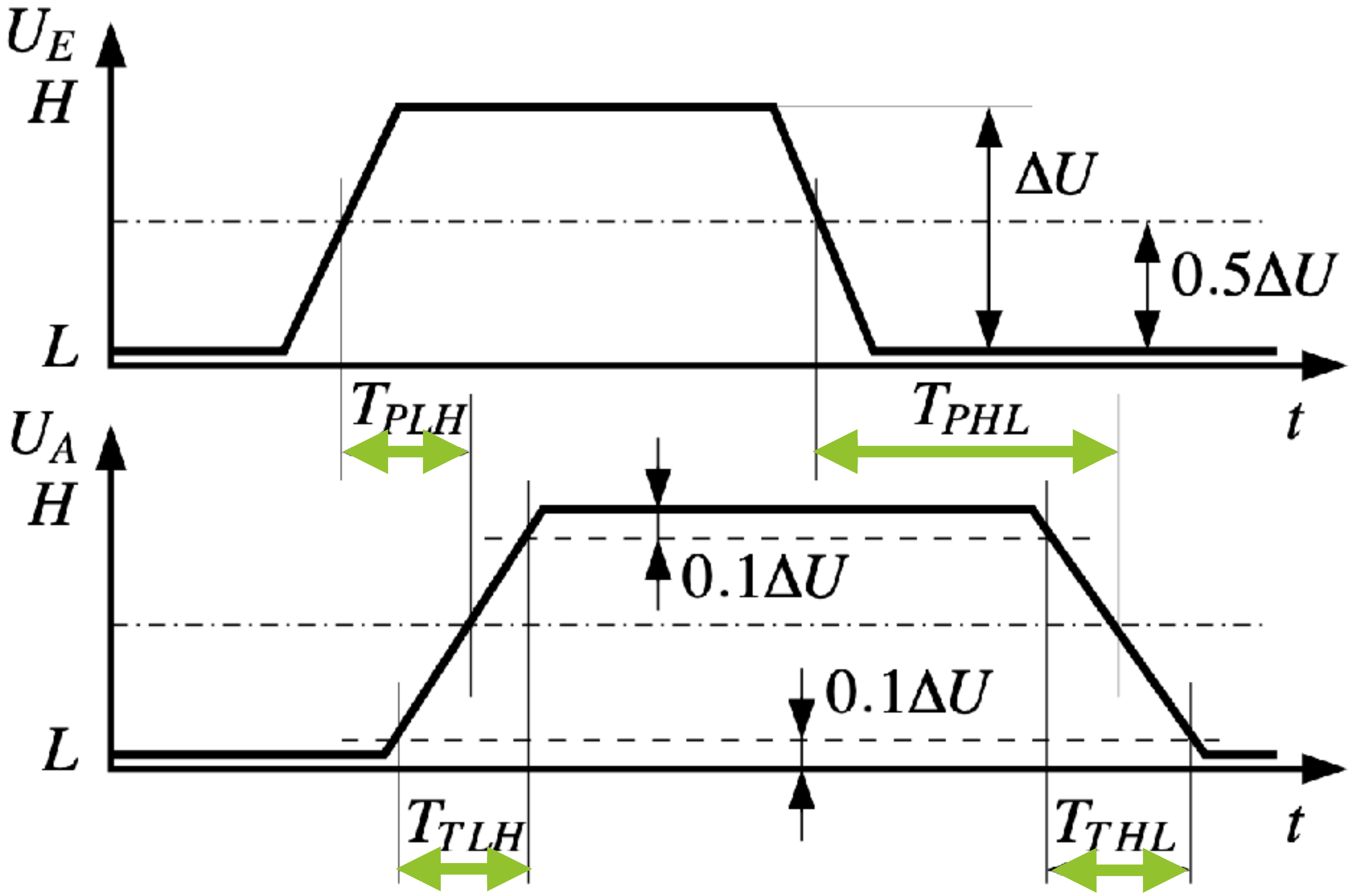
E_1	E_2	A
L	L	H
L	H	L
H	L	L
H	H	L

Schaltsymbol



NOR

Zeitverhalten in CMOS



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 (Boolesche Algebra)

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

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)

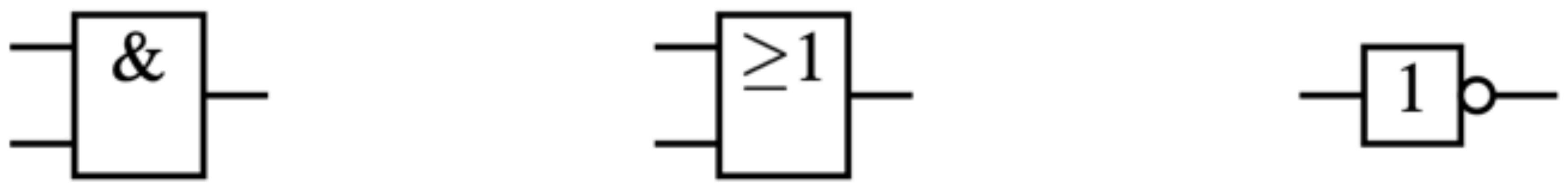
Allgemein wird nur positive Logik verwendent ($\rightarrow$ einheitlich verwenden).

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:

$\wedge$ $\vee$ $/$



AND OR NOT

Gesetze der Booleschen Algebra I

Kommutativgesetze

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

Assoziativgesetze

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

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

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

Tautologie

$$a \wedge a = a \qquad a \vee a = a$$

Komplementäres Element

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

Gesetze der Booleschen Algebra II

Dualitätsprinzip

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

De Morgansche Regeln

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

Weitere abgeleitete Regeln

$$(a \wedge b) \vee (a \wedge \overline{b}) = a \qquad (a \vee b) \wedge (a \vee \overline{b}) = a$$

$$a \wedge (\overline{a} \vee b) = a \wedge b \qquad a \vee (\overline{a} \wedge b) = a \vee b$$

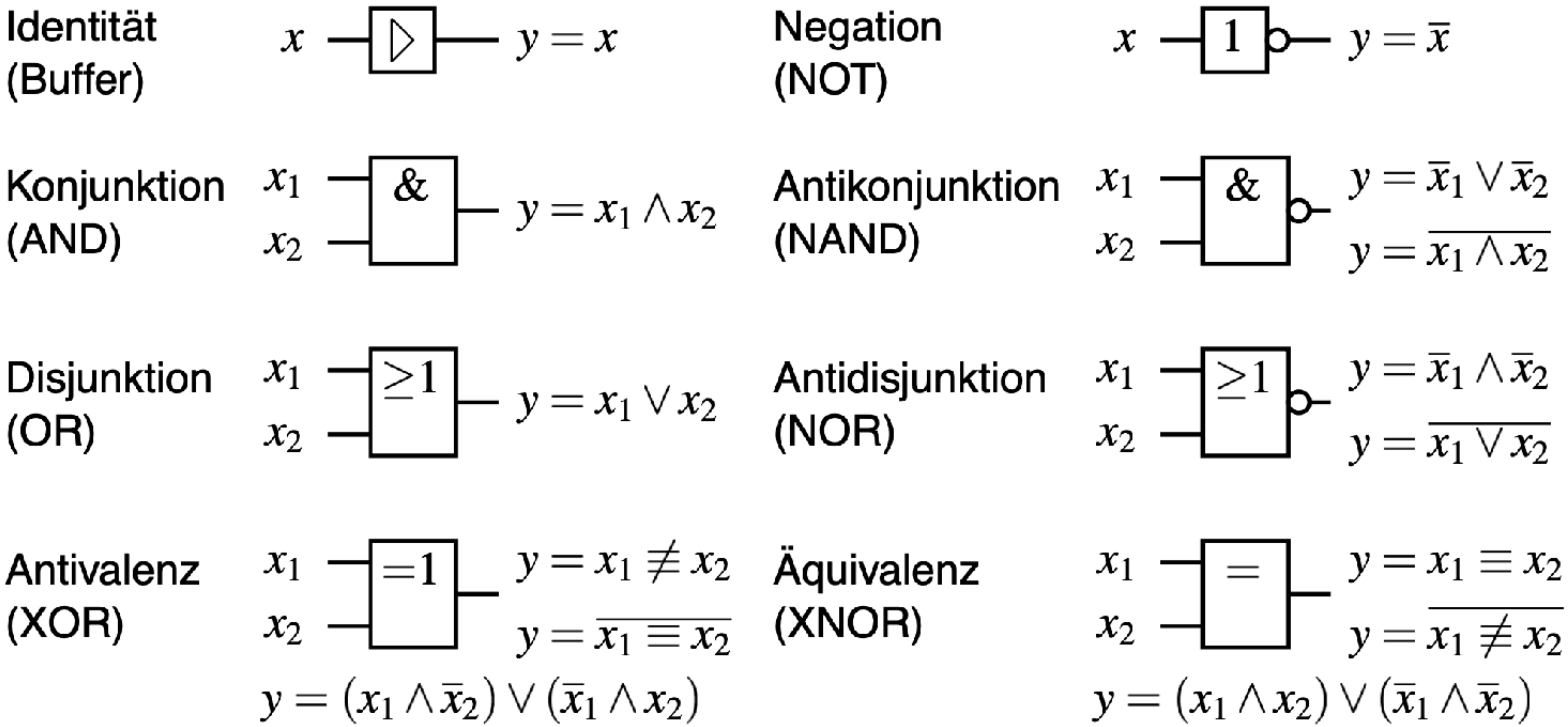
$$(a \vee b) \wedge (c \vee d) = (a \wedge (c \vee d)) \vee (b \wedge (c \vee d))$$

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

Verknüpfungen und Schaltfunktionen

$f(x_1, x_2)$	Funktion	Bezeichnung	Kürzel
f_1	$y_1 = 0$	Konstanz	0
f_2	$y_2 = x_1 \wedge x_2$	Konjunktion, Und	AND
f_3	$y_3 = x_1 \wedge \bar{x}_2$	Inhibition	
f_4	$y_4 = x_1$	Identität	
f_5	$y_5 = \bar{x}_1 \wedge x_2$	Inhibition	
f_6	$y_6 = x_2$	Identität	
f_7	$y_7 = x_1 \not\equiv x_2$	Antivalenz, Exlusives Oder	XOR
f_8	$y_8 = x_1 \vee x_2$	Disjunktion, Oder	OR
f_9	$y_9 = \overline{x_1 \vee x_2}$	Antidisjunktion, Antialternative	NOR
f_{10}	$y_{10} = x_1 \equiv x_2$	Äquivalenz	XNOR
f_{11}	$y_{11} = \bar{x}_2$	Negation, Nicht	NOT
f_{12}	$y_{12} = x_1 \vee \bar{x}_2$	Implikation	
f_{13}	$y_{13} = \bar{x}_1$	Negation, Nicht	NOT
f_{14}	$y_{14} = \bar{x}_1 \vee x_2$	Implikation	
f_{15}	$y_{15} = \overline{x_1 \wedge x_2}$	Antikonjunktion	NAND
f_{16}	$y_{16} = x_1$	Konstanz	1

Schaltsymbole für Verknüpfungsglieder



Darstellung von Schaltfunktionen

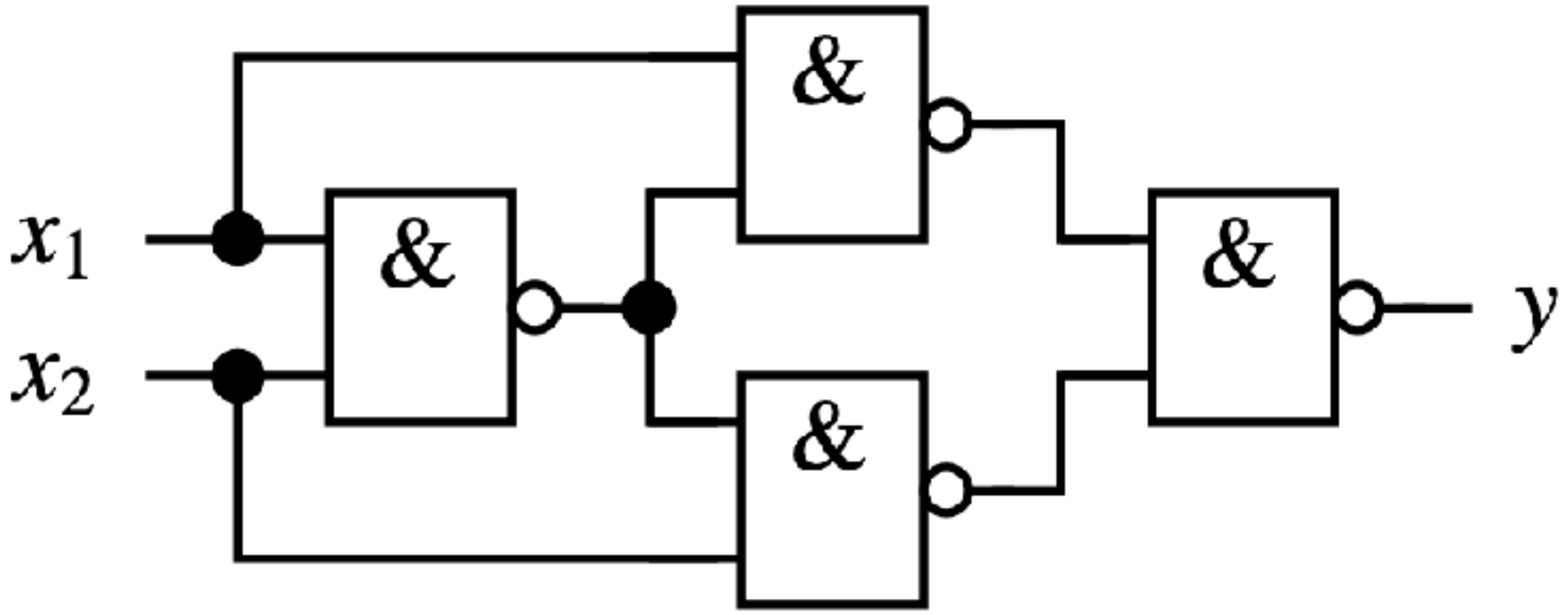
Wertetabelle
(Wahrheitstabelle)

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)



Normalformen

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).

Beispiel Normalformen

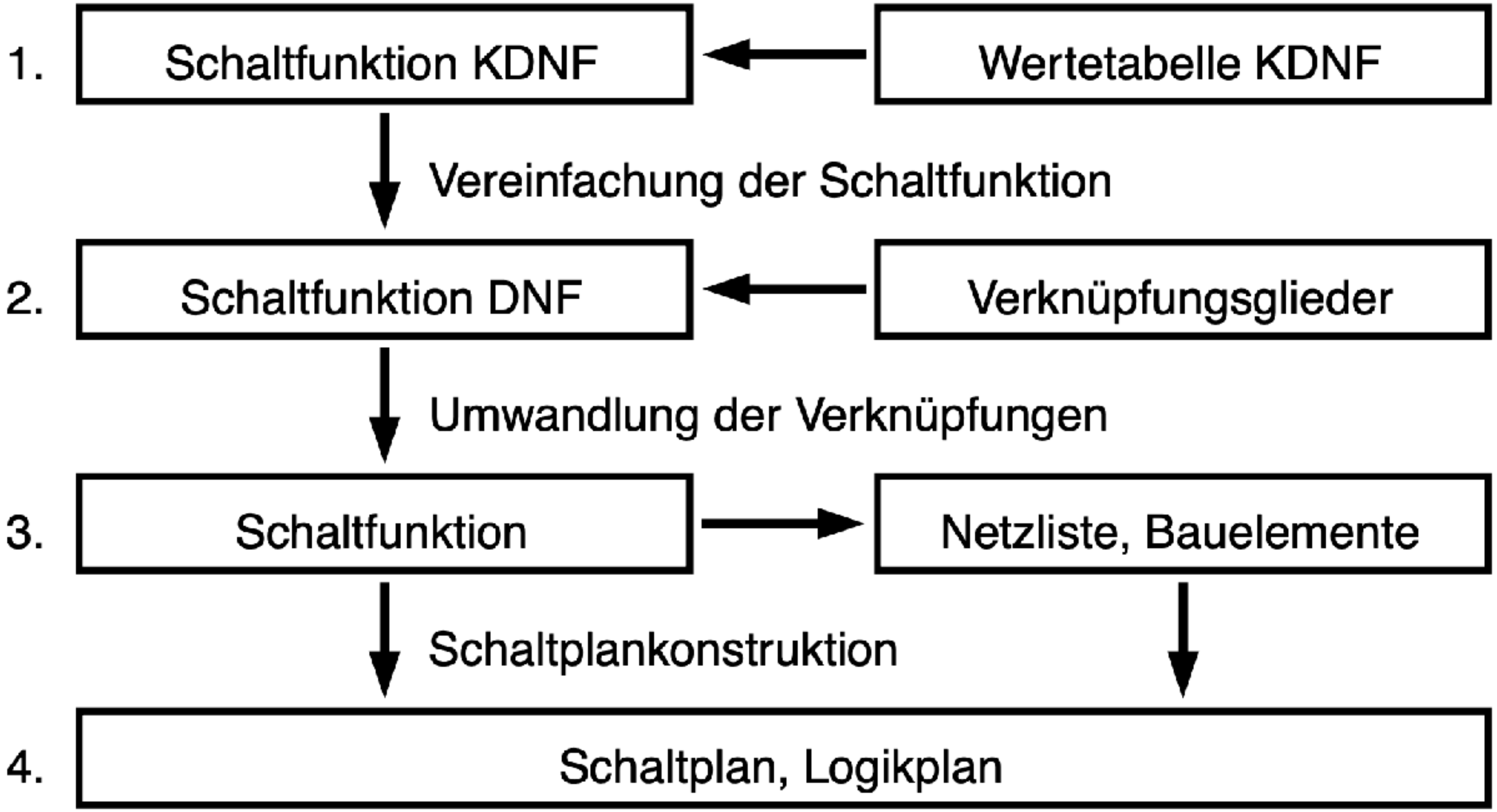
x_1	x_2	x_3	y	Minterme	Maxterme
0	0	0	0		$x_1 \vee x_2 \vee x_3$
0	0	1	1	$\bar{x}_1 \wedge \bar{x}_2 \wedge x_3$	
0	1	0	1	$\bar{x}_1 \wedge x_2 \wedge \bar{x}_3$	
0	1	1	0		$x_1 \vee \bar{x}_2 \vee \bar{x}_3$
1	0	0	1	$x_1 \wedge \bar{x}_2 \wedge \bar{x}_3$	
1	0	1	0		$\bar{x}_1 \vee x_2 \vee \bar{x}_3$
1	1	0	0		$\bar{x}_1 \vee \bar{x}_2 \vee x_3$
1	1	1	1	$x_1 \wedge x_2 \wedge x_3$	

Kurzschreibweise für Schaltfunktionen: $\wedge \;\rightarrow\; \cdot$ und $\vee \;\rightarrow\; +$

KDNF: $y = \bar{x}_1\bar{x}_2x_3 + \bar{x}_1x_2\bar{x}_3 + x_1\bar{x}_2\bar{x}_3 + x_1x_2x_3$

KKNF: $y = (x_1 + x_2 + x_3)(x_1 + \bar{x}_2 + \bar{x}_3)(\bar{x}_1 + x_2 + \bar{x}_3)(\bar{x}_1 + \bar{x}_2 + x_3)$

Synthese eines Schaltplanes aus einer Schaltfunktion



Aufgaben — Digitale Logik I

1. Gegeben sei die Schaltfunktion: $\mathbf{y = (x_2 + x_1)(x_1 + x_0)}$

a) Lösen Sie die Klammern auf und optimieren Sie die Gleichung.

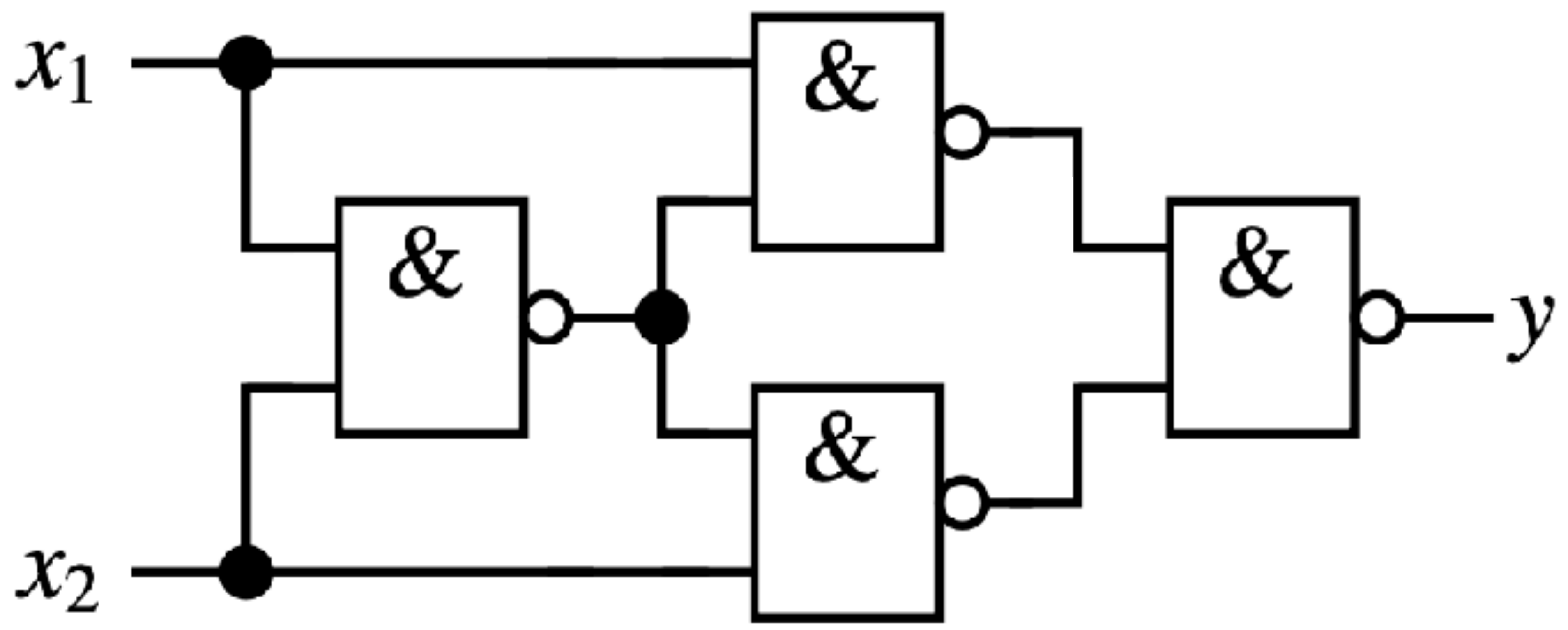
b) Stellen Sie die Funktion als Wertetabelle dar.

c) Realisieren Sie die Funktion durch eine logische Schaltung.

d) Formen Sie die Funktion so um, dass sie ausschließlich mit NAND-Gattern mit zwei Eingängen realisiert werden kann.

Aufgaben — Digitale Logik II

2. Das nachfolgende Schaltnetz ist zu analysieren:



a) Stellen Sie die vollständige Wertetabelle für die Ausgangsfunktion **y** auf (durch Simulation und symbolischer Berechnung der Schaltung).

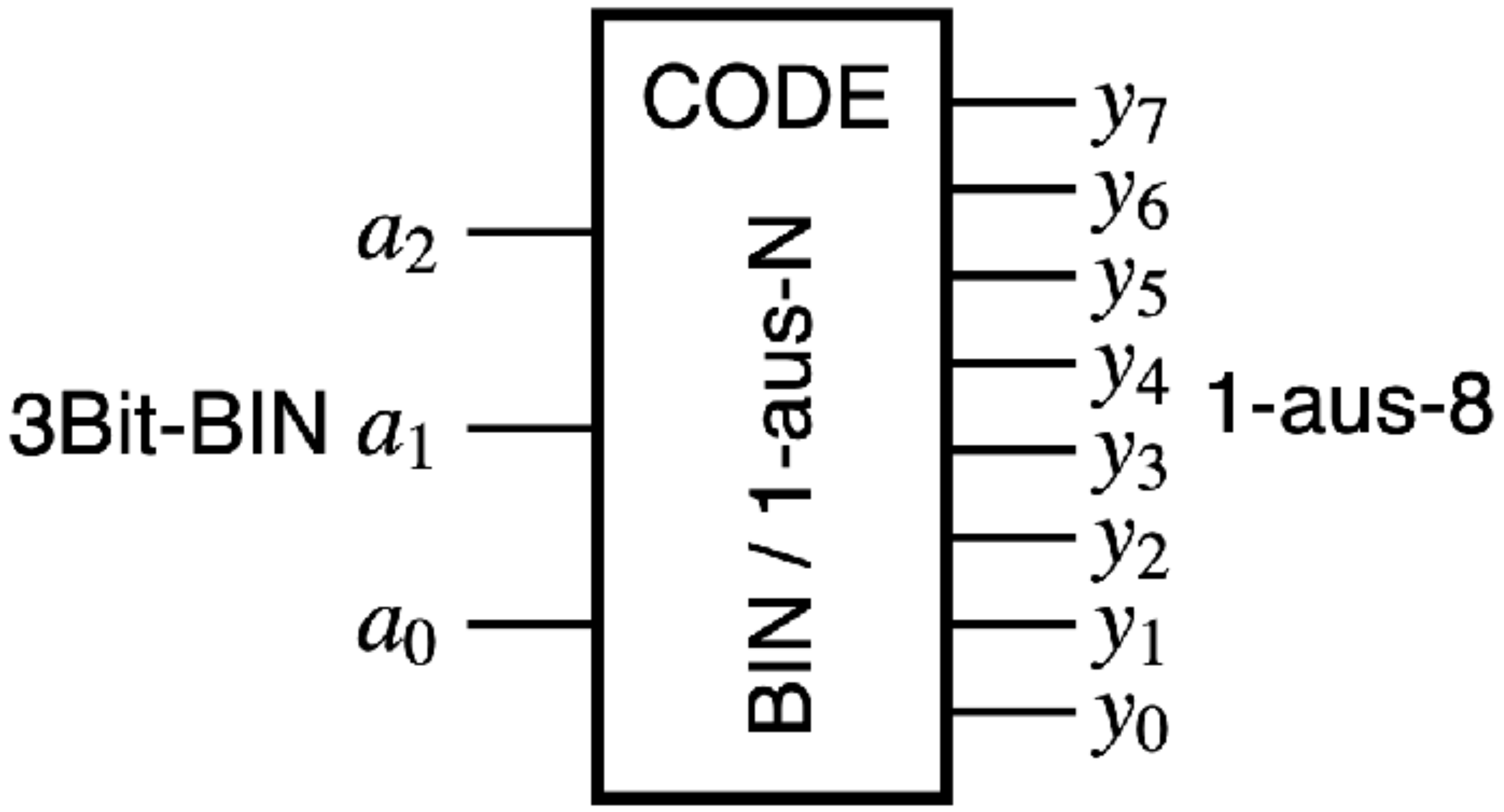
b) Als disjunktive Normalform (Oder-Verknüpung aller Vollkonjunktionen mit dem logischen Wert 1)

Aufgaben — Digitale Logik III

3. Der Aressdekodierer kodiert binäre Adressen in eine 1-aus-N Kodierung. Bei dieser Kodierung ist nur ein Ausgang entsprechend dem Eingangsbinäräquivalent mit 1 belegt.

a) Stellen Die die Wertetabelle auf.

b) Geben Sie die acht Funktionen für den Kodierer an.

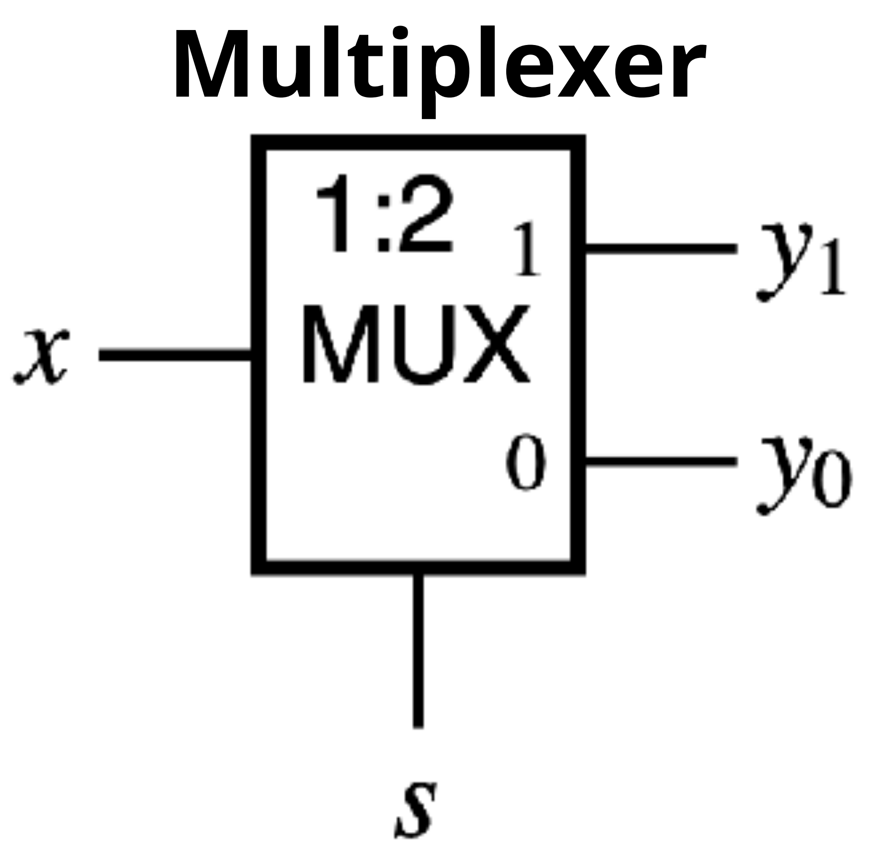


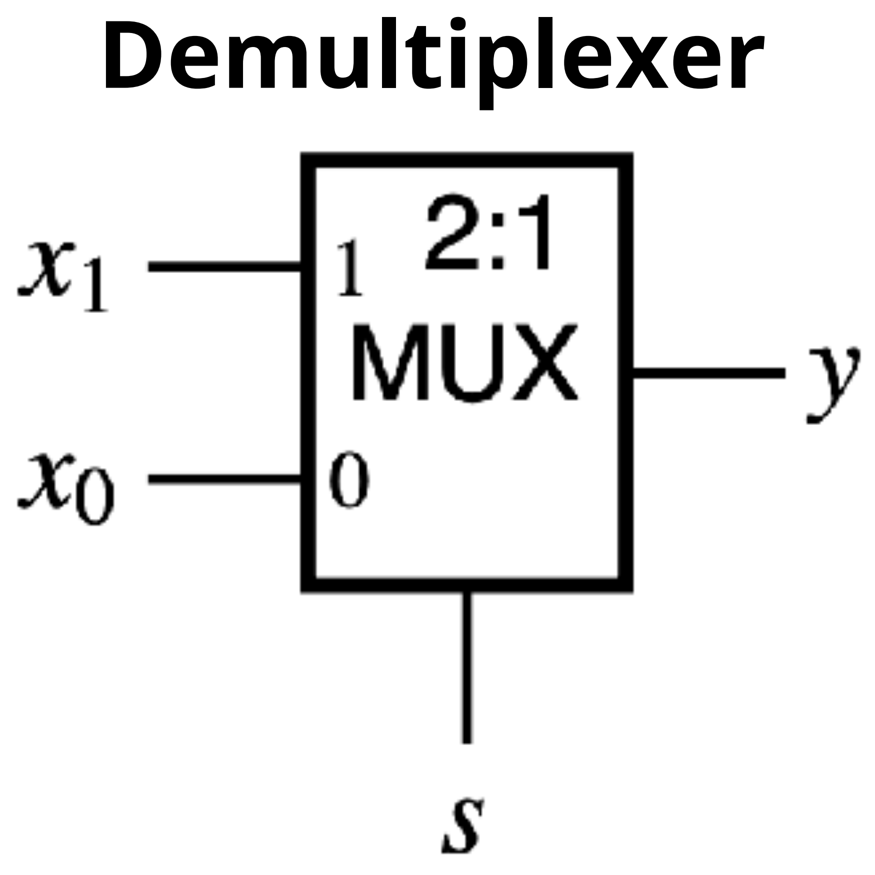
Aufgaben — Digitale Logik IV

4. 1:2 Multiplexer bzw. 2:1 Multiplexer (Demultiplexer) dienen zur Umschaltung von Datenleitungen. Zusammengefaßt als Multiplexerbäume können mit ihnen komplexe Schaltnetze realisiert werden.

a) Stellen Sie die Wertetabellen auf.

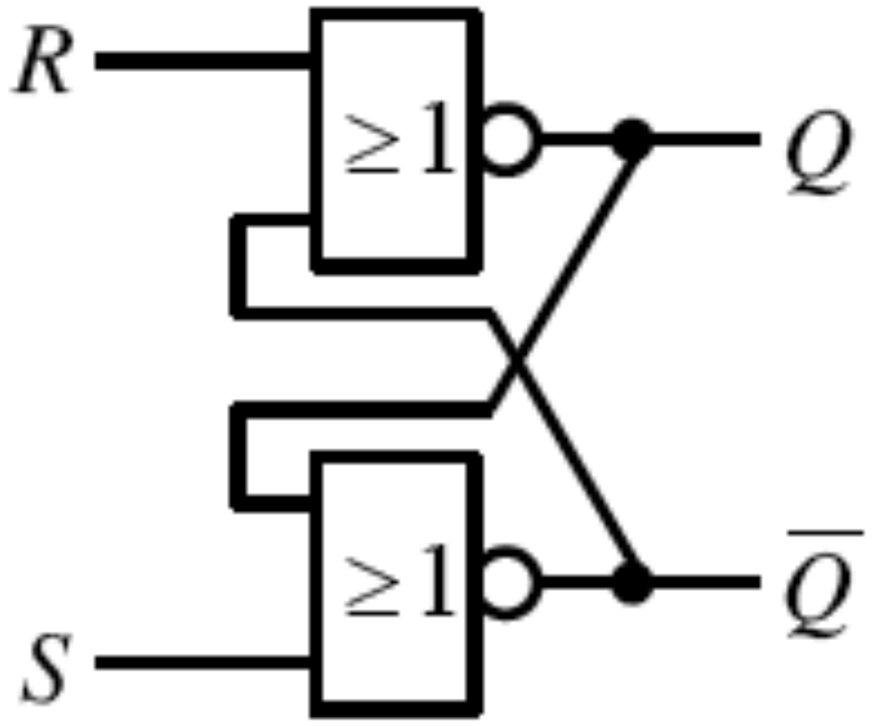
b) Geben Sie die Funktionen y für beide Multiplexer und Demultiplexer an.





FlipFlops (einfache Speicherglieder)

SR-FlipFlop - Zustandstabelle



SR-Flipflop
(Basis-Flipflop)

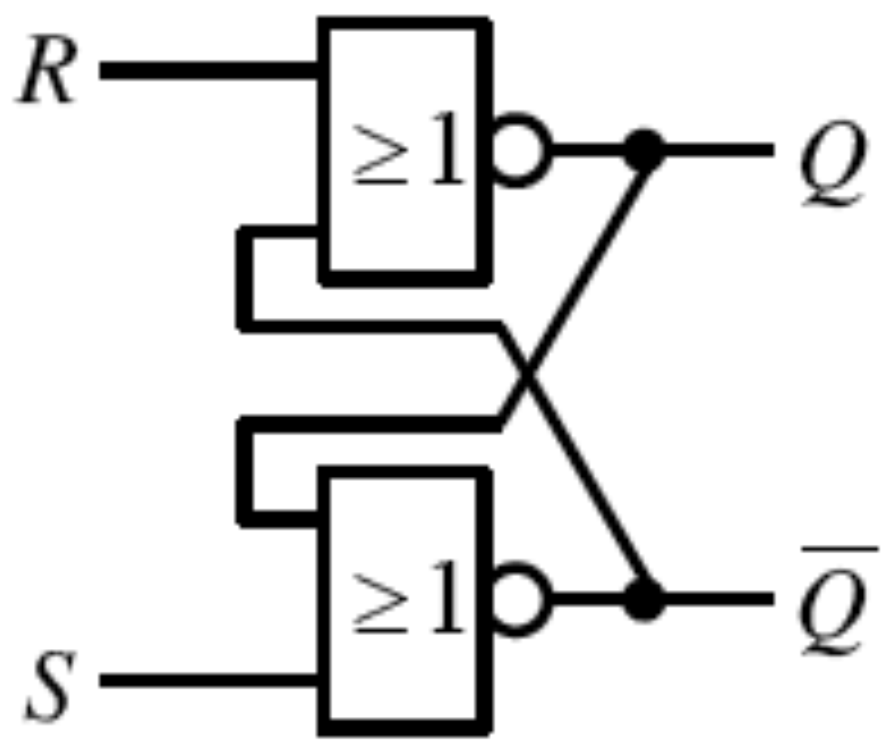
Kurzdarstellung

$$R(t_n) \rightarrow R$$
$$S(t_n) \rightarrow S$$
$$Q(t_n) \rightarrow Q$$
$$Q(t_{n+1}) \rightarrow Q^+$$

	(t_n)		(t_{n+1})	
R	S	Q	Q^+	**Zustandsfolgetabelle**
0	0	0	0	speichern
0	1	0	1	setzen
1	0	0	0	rücksetzen
1	1	0	–	nicht zulässig
0	0	1	1	speichern
0	1	1	1	setzen
1	0	1	0	rücksetzen
1	1	1	–	nicht zulässig

Abbildung aktueller Zeitpunkt → Folgezeitpunkt

SR-FlipFlop - Zustandsübergänge



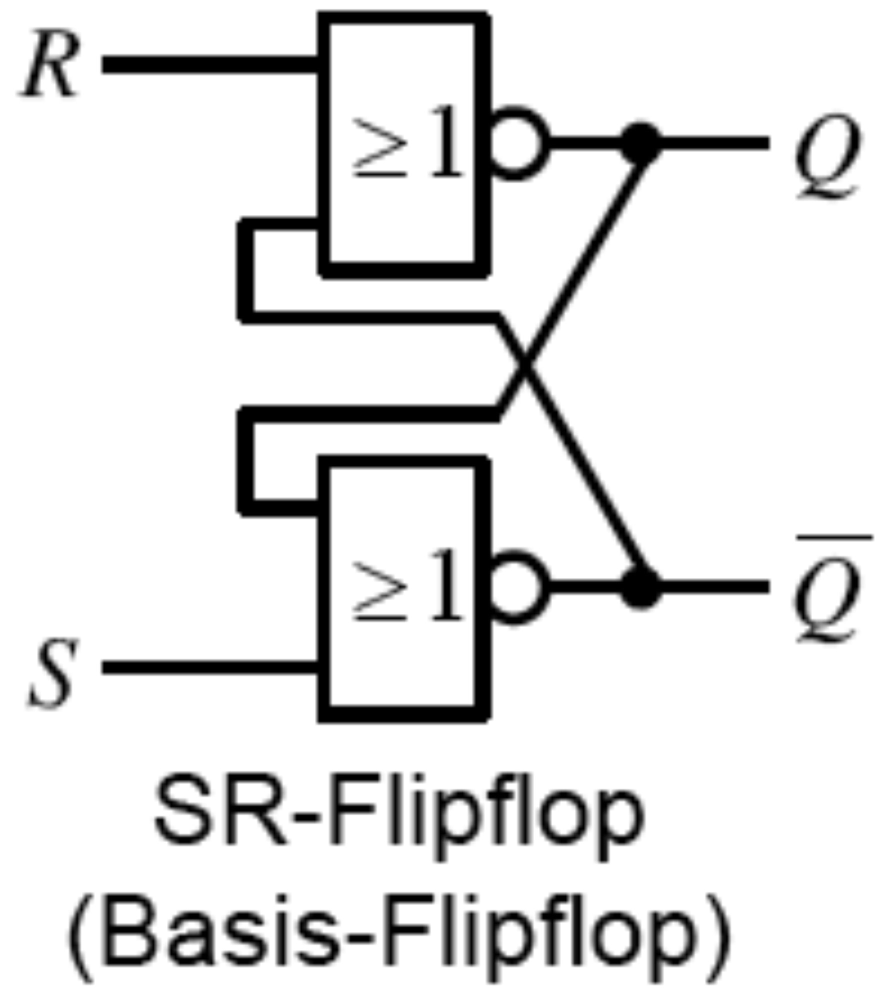
SR-Flipflop
(Basis-Flipflop)

Zustandsübergangstabelle

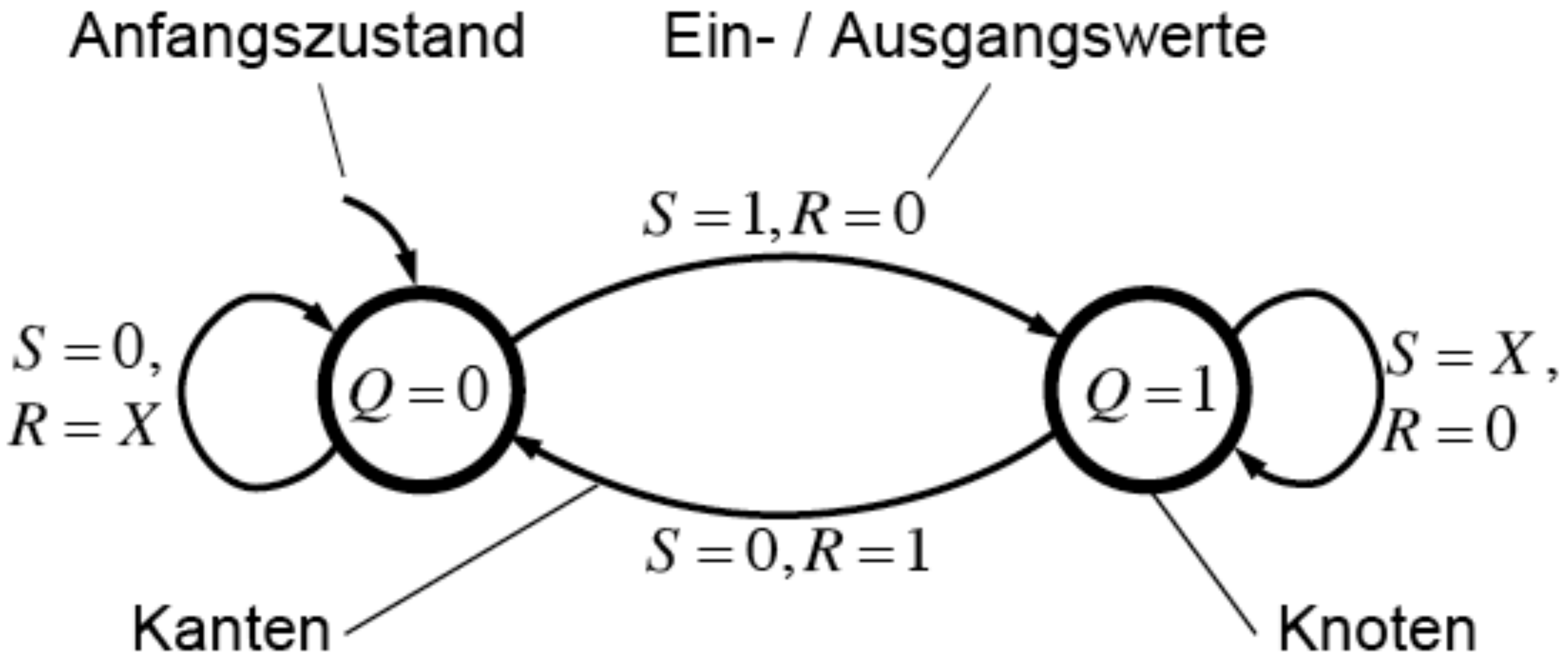
(t_n)	$\rightarrow$	(t_{n+1})	(t_n)		
Q	$\rightarrow$	Q^+	R	S	
0	$\rightarrow$	0	X	0	speichern oder rücksetzen, nicht setzen
0	$\rightarrow$	1	0	1	setzen
1	$\rightarrow$	0	1	0	rücksetzen
1	$\rightarrow$	1	0	X	speichern oder setzen, nicht rücksetzen

Antwort auf die Frage: Welche Belegung der Eingänge R und S ist erforderlich, um vom Zustand Q in den Zustand Q^+ zu gelangen (→ Zustandsübergang).

SR-FlipFlop - Zustandsgraph



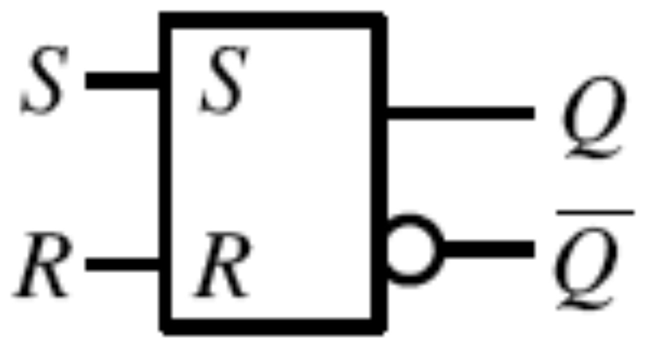
Knoten: entsprechen den Zuständen
Kanten: entsprechen den Zustandsübergängen
Anfangszustand, einseitige Kante



Zustandsfolgetabelle → Zustandsübergangstabelle → Zustandsgraph

SR-FlipFlop - Darstellungen

Schaltsymbol



SR-Flipflop

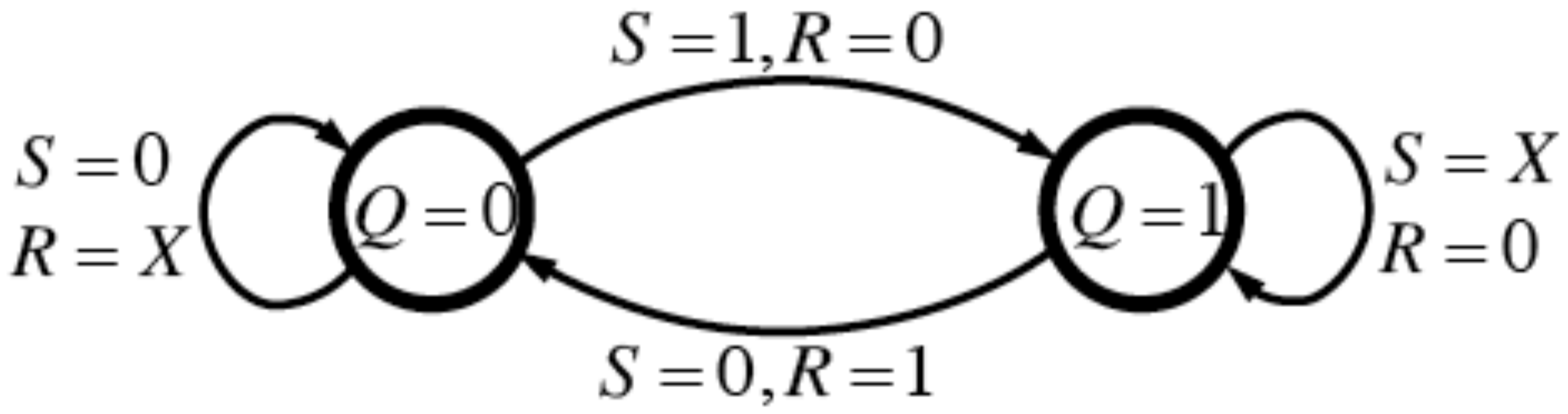
Zustandsfolgetabelle

R	S	Q^+	$\overline{Q}^+$
0	0	Q	Q
0	1	1	0
1	0	0	1
1	1	–	–

Zustandsübergangstabelle

Q	$\rightarrow$	Q^+	R	S
0	$\rightarrow$	0	X	0
0	$\rightarrow$	1	0	1
1	$\rightarrow$	0	1	0
1	$\rightarrow$	1	0	X

Zustandsgraph



Boolesche Gleichung

$$Q^+ := (S \wedge \overline{R}) \vee (\overline{S} \wedge \overline{R} \wedge Q)$$

mit $S \wedge R = 0$

Synchrone Speicherglieder

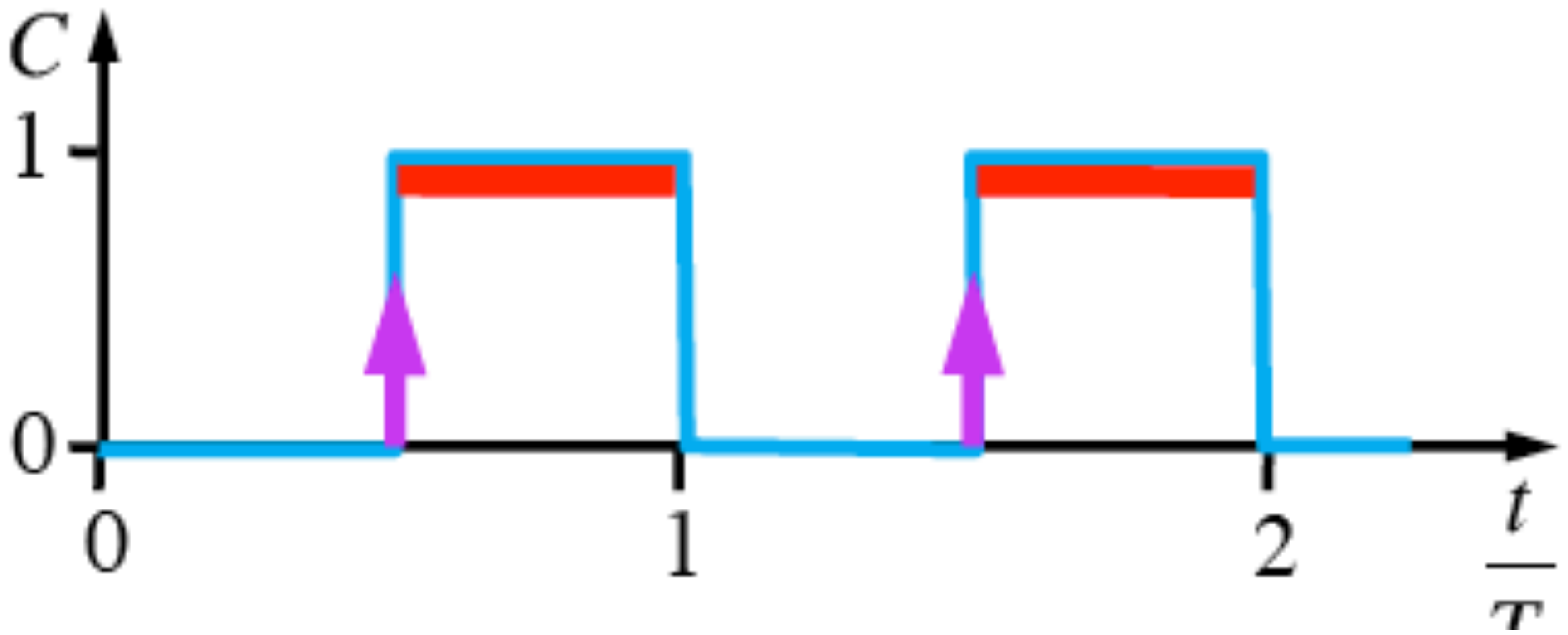
Asynchrone Speicherglieder:

Der Zustand und die Ausgangswerte ändern sich unmittelbar nach der Änderung der Eingangswerte → **Datensteuerung** (bisherige Betrachtung)

Synchrone Speicherglieder:

Der Zustand und die Ausgangswerte ändern sich synchron zu einem Taktsignal → **Taktsteuerung** (Einführung eines Taktsignales)

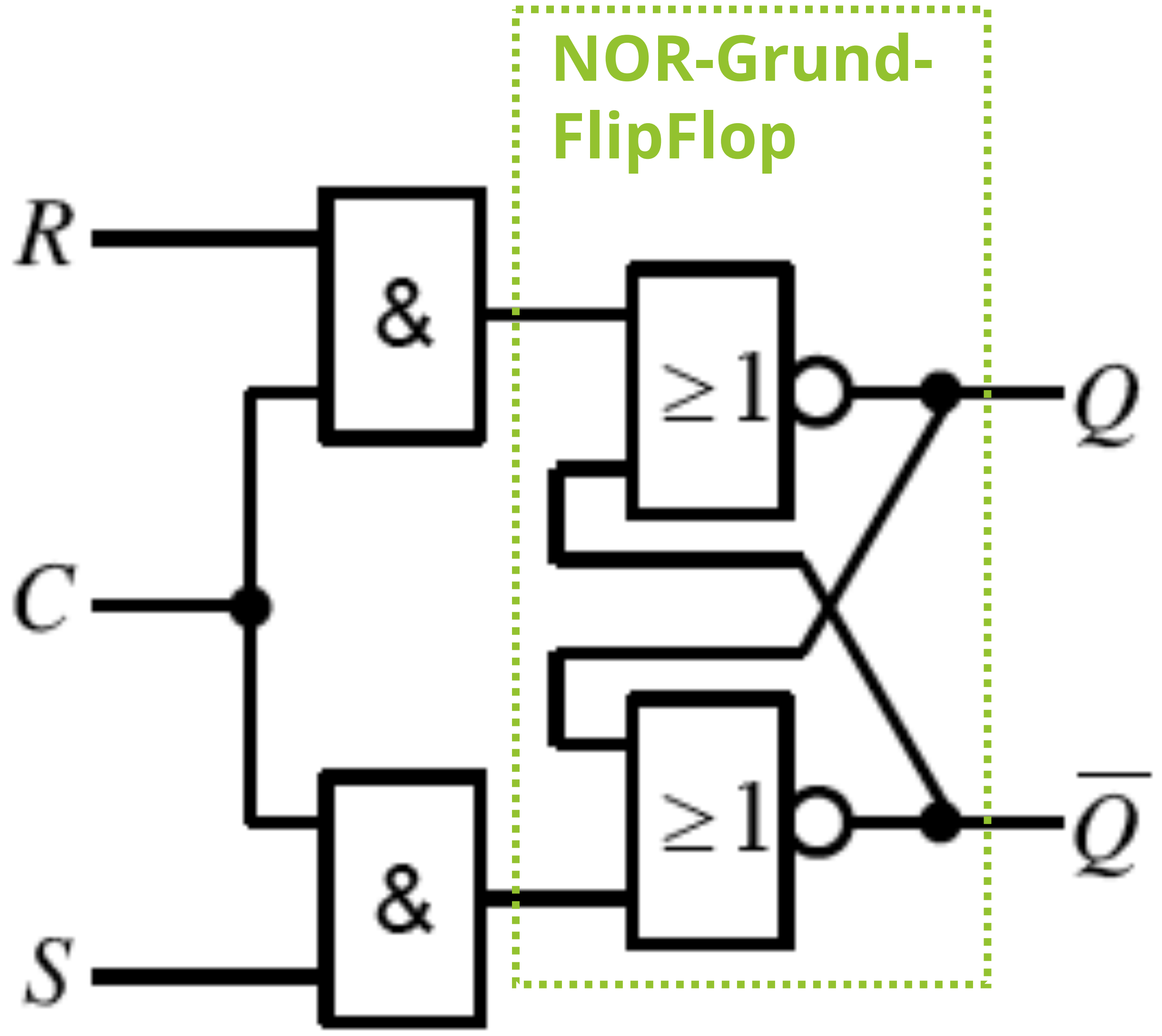
Taktsignal (Clock, Synchronisationssignal):



aktiver Taktzustand $C=1$
→ Taktzustandssteuerung

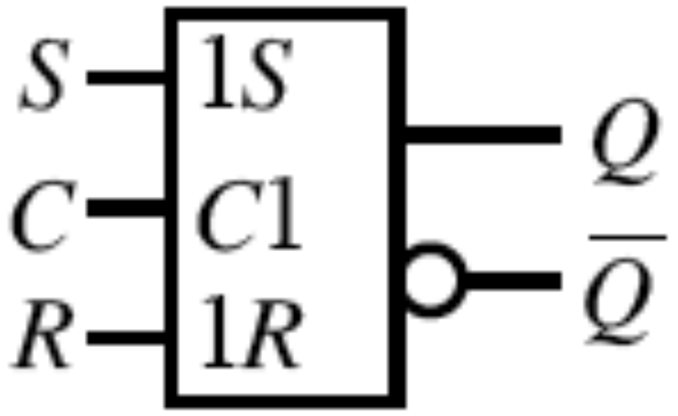
aktive Taktflanke $C\uparrow$ o. $C: 0 \rightarrow 1$
→ Taktflankensteuerung

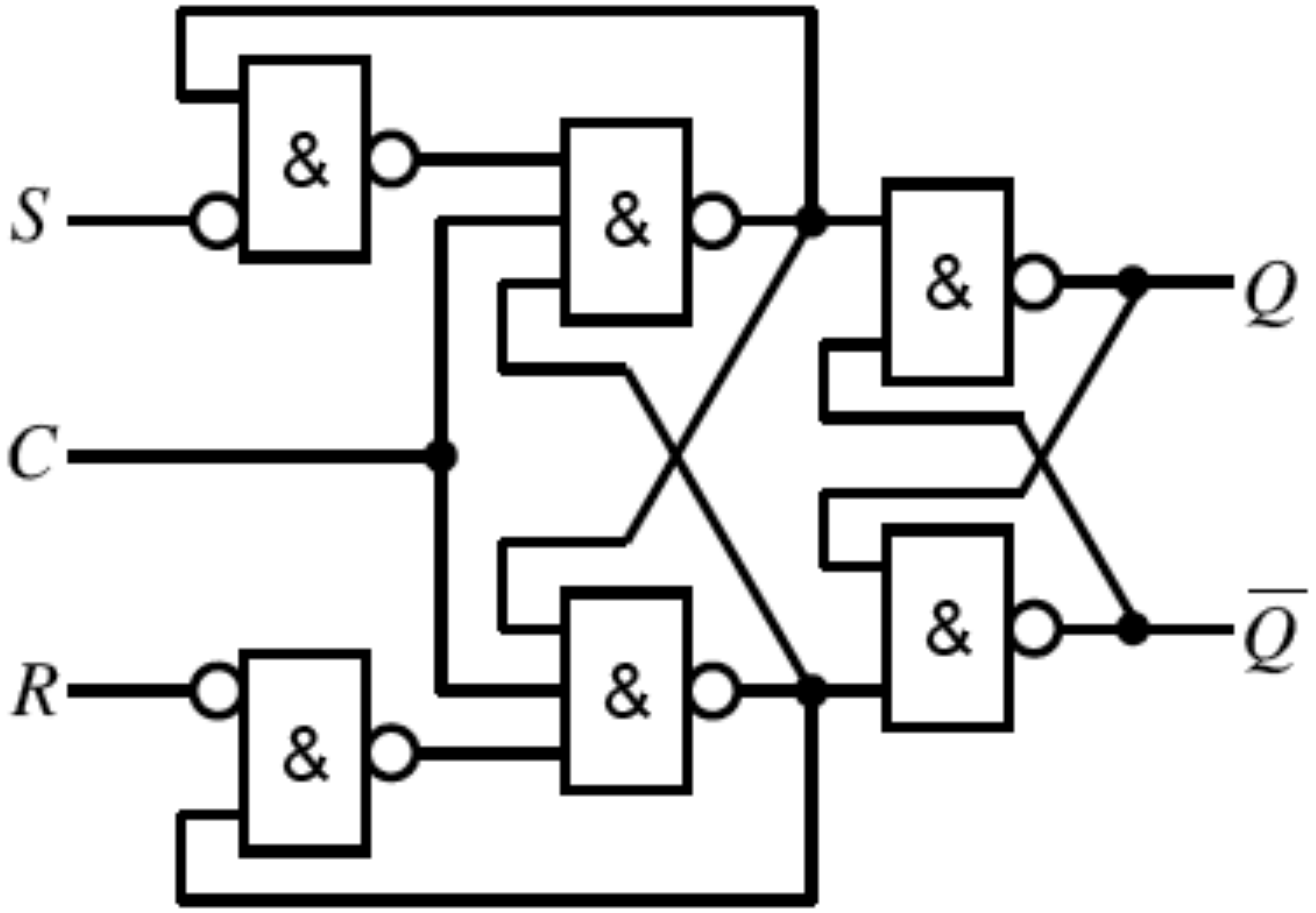
Synchrones SR-FlipFlop I



taktzustandsgesteuert

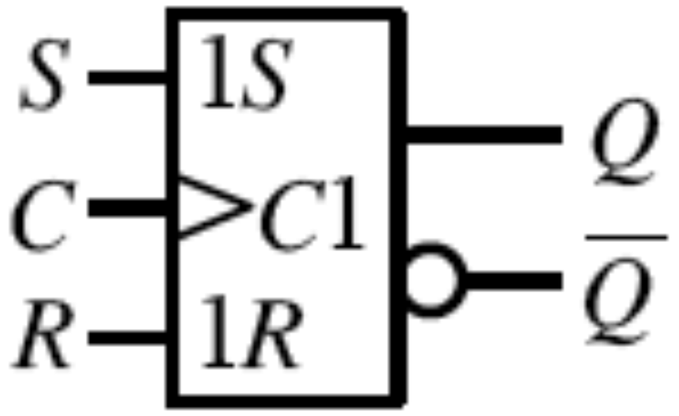
Schaltsymbol SR-FF TZS





taktflankengesteuert

Schaltsymbol SR-FF TFS



Synchrones SR-FlipFlop II

Taktzustandsgesteuertes SR-Flipflop

R	S	C	Q^+	
0	0	1	Q	speichern
0	1	1	1	setzen
1	0	1	0	rücksetzen
1	1	1	–	nicht zulässig
X	X	0	Q	speichern

Taktflankengesteuertes SR-Flipflop

R	S	C	Q^+	
0	0	$\uparrow$	Q	speichern
0	1	$\uparrow$	1	setzen
1	0	$\uparrow$	0	rücksetzen
1	1	$\uparrow$	–	nicht zulässig
X	X	sonst	Q	speichern

Auf die Angabe des Taktes in der Zustandsfolgetabelle und im Zustandsgraphen kann verzichtet werden, wenn als Nebenbedingung die Art der Taktsteuerung mit angegeben wird (TZS oder TFS).

Ist der Takt nicht aktiv (Zustand oder Flanke), so speichert das Flipflop in jedem Fall den aktuellen Zustand.

Weitere Speicherglieder

Ansteuerschaltungen für das SR-Flipflop zur Vermeidung der nicht zulässigen Eingangsbelegung $S{=}R{=}1$ und für spezielle Ansteuervarianten und Funktionen

Flipflop Varianten:

Typ	Eingänge	Funktionen
T-FF	1	invertieren, speichern
D-FF	1	setzen, rücksetzen
SR-FF	2	speichern, setzen, rücksetzen
JK-FF	2	speichern, setzen, rücksetzen, invertieren

Die einzelnen Flipflop-Typen lassen sich durch Zusatzbeschaltungen ineinander überführen. Alle Flipflop-Typen sind gleichwertig anwendbar.

In der Computertechnik dominieren D-Flipflop (Delay-Flipflop) mit Taktflankensteuerung (teilweise auch mit Taktzustandssteuerung).

Aufgaben — FlipFlops

5. Machen Sie sich mit der Funktionsweise eines NOR-Grund-FlipFlops vertraut.

a) Stellen Sie die Wertetabelle auf.

b) Was unterscheidet ein FlipFlop von einem Schaltnetz?

c) Stellen Sie die Zustandsübergangstabelle auf.

d) Was passiert bei der Eingangsbelegung R = S = 1

e) Entwerfen Sie eine kombinatorische Vor-Schaltung bei der das Setzen bei der Eingangsbelegung R = S = 1 dominiert.

f) Entwerfen Sie eine kombinatorische Vorschaltung, welche aus einem RS-FlipFlop ein D-FlipFlop macht.

Schaltwerke

Schaltwerke - Übersicht

- Schaltwerke – Hauptbestandteile von Computern (v. Neumann Rechner)
- Schaltwerke enthalten Schaltnetze, Speicherglieder (Flipflops) und Signalrückführungen
- der in den Speichergliedern gespeicherte Zustand heißt „innerer Zustand" des Schaltwerkes
- bei einem Schaltwerk hängen die Werte der Ausgangsvariablen von denen der Eingangsvariablen und vom inneren Zustand (der Vorgeschichte) ab
- einer Wertefolge der Eingangsvariablen ist damit eindeutig eine Wertefolge der Ausgangsvariablen zugeordnet (→ Anfangszustand)
- charakteristisch für ein Schaltwerk ist die funktionelle Bedeutung der Zeit (diskrete Zeitpunkte werden durch ein Taktsignal realisiert)
- Flipflops sind einfache Schaltwerke

Schaltwerke - Definition

Schaltwerk (nach DIN44300):

Ein Funktionseinheit zum Verarbeiten von Schaltvariablen, bei der die Werte aller Schaltvariablen am Ausgang (Ausgangsvariablen) zu einem bestimmten Zeitpunkt t_0 nach Verstreichen der Lauf- und Verzögerungs-zeit Δt nur von den Werten aller Schaltvariablen am Eingang (Eingangsvariablen) zum Zeitpunkt t-Δt abhängen und zu endlich vielen vorangegangenen Zeitpunkten sowie ggf. vom Anfangszustand.

(Ein Schaltwerk hat eine endlich Anzahl von inneren Zuständen und ist abstrakt gesehen ein endlicher Automat. Falls keine besonderen Vorkehrungen getroffen werden, können Schaltwerke beim Einschalten einen unbestimmten Anfangszustand annehmen.)

Speicherglied (nach DIN44300) :

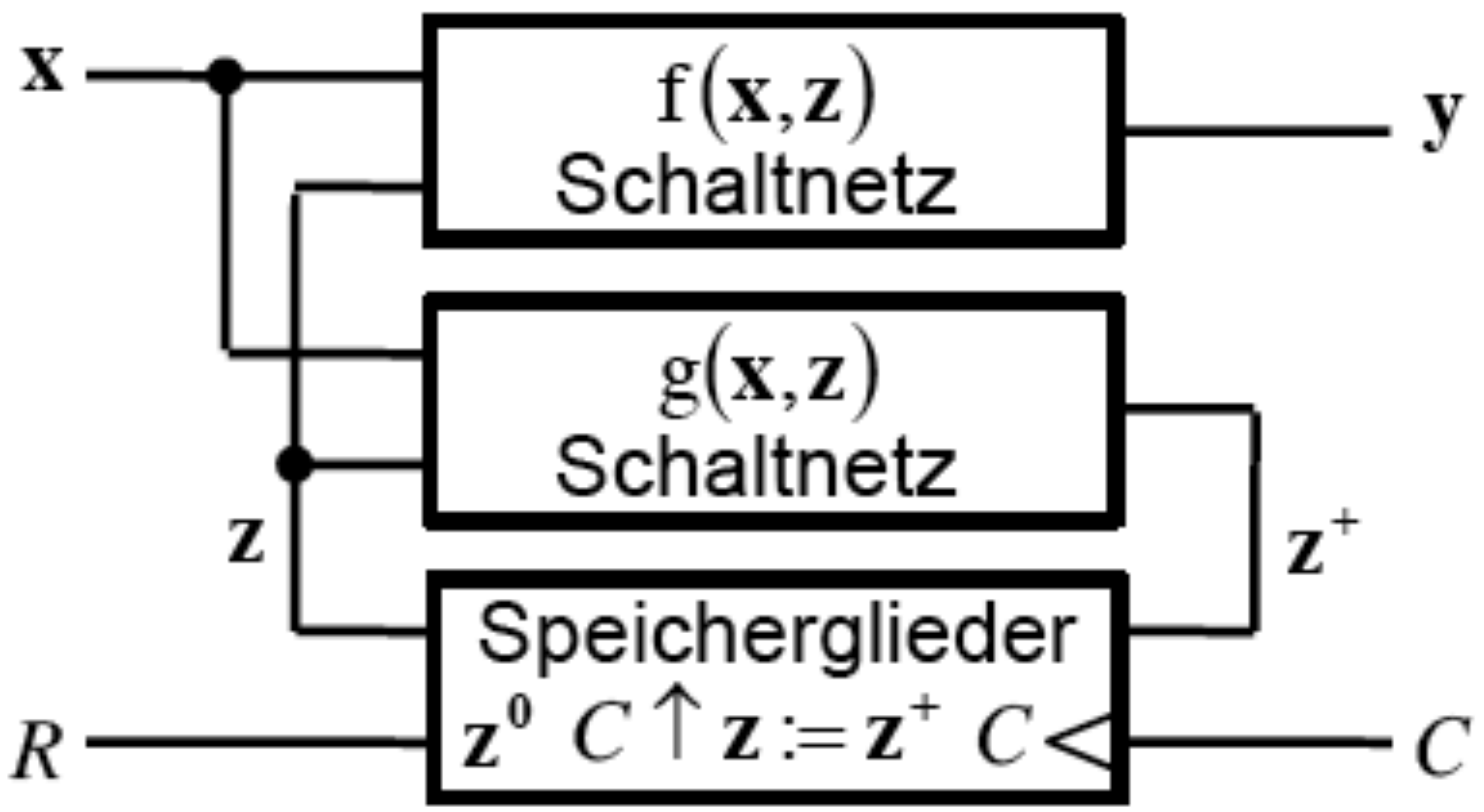
Ein Bestandteil eines Schaltwerkes, der Werte von Schaltvariablen aufnimmt, aufbewahrt und abgibt.

Schaltwerke

Bestandteile von Schaltwerken:

- Speicherglieder (Flipflop)
- Schaltnetze
- Verbindungen, Rückführungen
- Taktsignal (→ synchrone Schaltwerke).

Allgemeine Darstellungsform (Huffman-Modell)



$\mathbf{x}$:	Eingangsvektor
$\mathbf{y}$:	Ausgangsvektor
$\mathbf{z}$:	Zustandsvektor
$\mathbf{z}^0$:	Anfangszustand
$\mathbf{z}^+$:	Folgezustandsvektor
$\mathbf{y} = \mathrm{f}(\mathbf{x},\mathbf{z})$	Ausgangsfunktion
$\mathbf{z}^+ = \mathrm{g}(\mathbf{x},\mathbf{z})$	Übergangsfunktion
C :	Taktsignal (clock)
R :	Rücksetzsignal (reset)

Automatentheorie

Automat – Modellmaschine

- abstraktes mathematisches Modell → Modellmaschine
- Kennzeichnung durch einen inneren Zustand → Zustandsautomat
- Automaten arbeiten sequentiell, gesteuert durch Takt oder Eingang.
- Automaten durchlaufen eine Abfolge von Zuständen, beginnend mit einem Anfangszustand, in Abhängigkeit von Eingangswerten oder Takt.
- Die Ausgabewerte der Modellmaschine sind von den aktuellen Eingabewerten und vom momentanen inneren Zustand der Maschine abhängig.

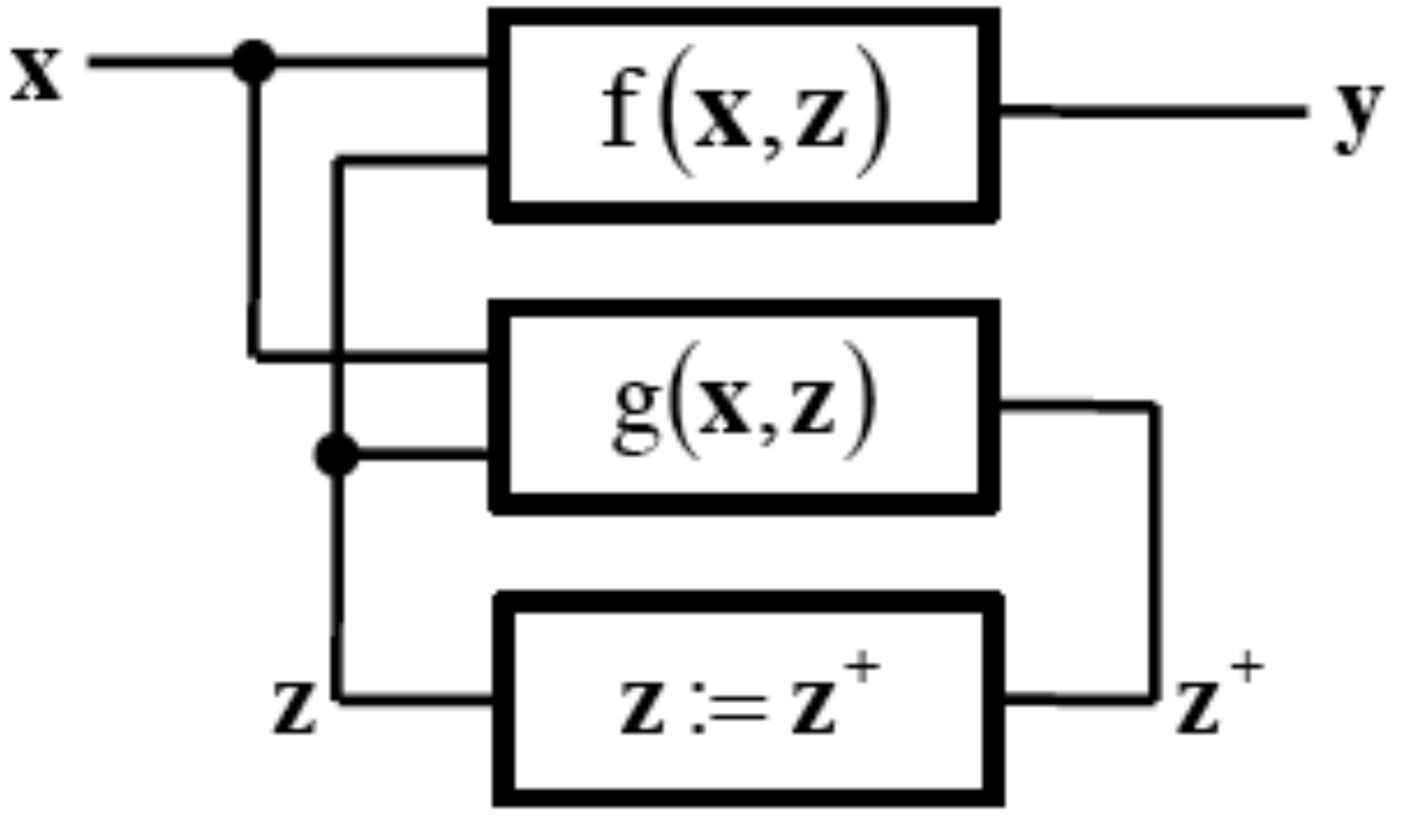
Endliche Automaten

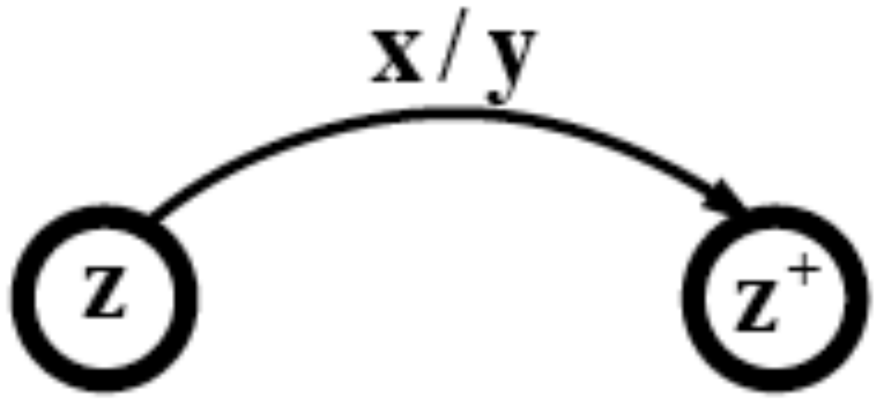
Die Menge der möglichen Eingabezeichen (Eingabealphabet), der Ausgabezeichen (Ausgabealphabet) und die Zahl der möglichen inneren Zustände (Zustandsmenge) sind endlich.

Deterministische Automaten

Das Verhalten der Modellmaschine ist deterministisch, für eine gegebene Folge von Eingangswerten komplett vorhersagbar, determiniert.

Mealy-Automaten



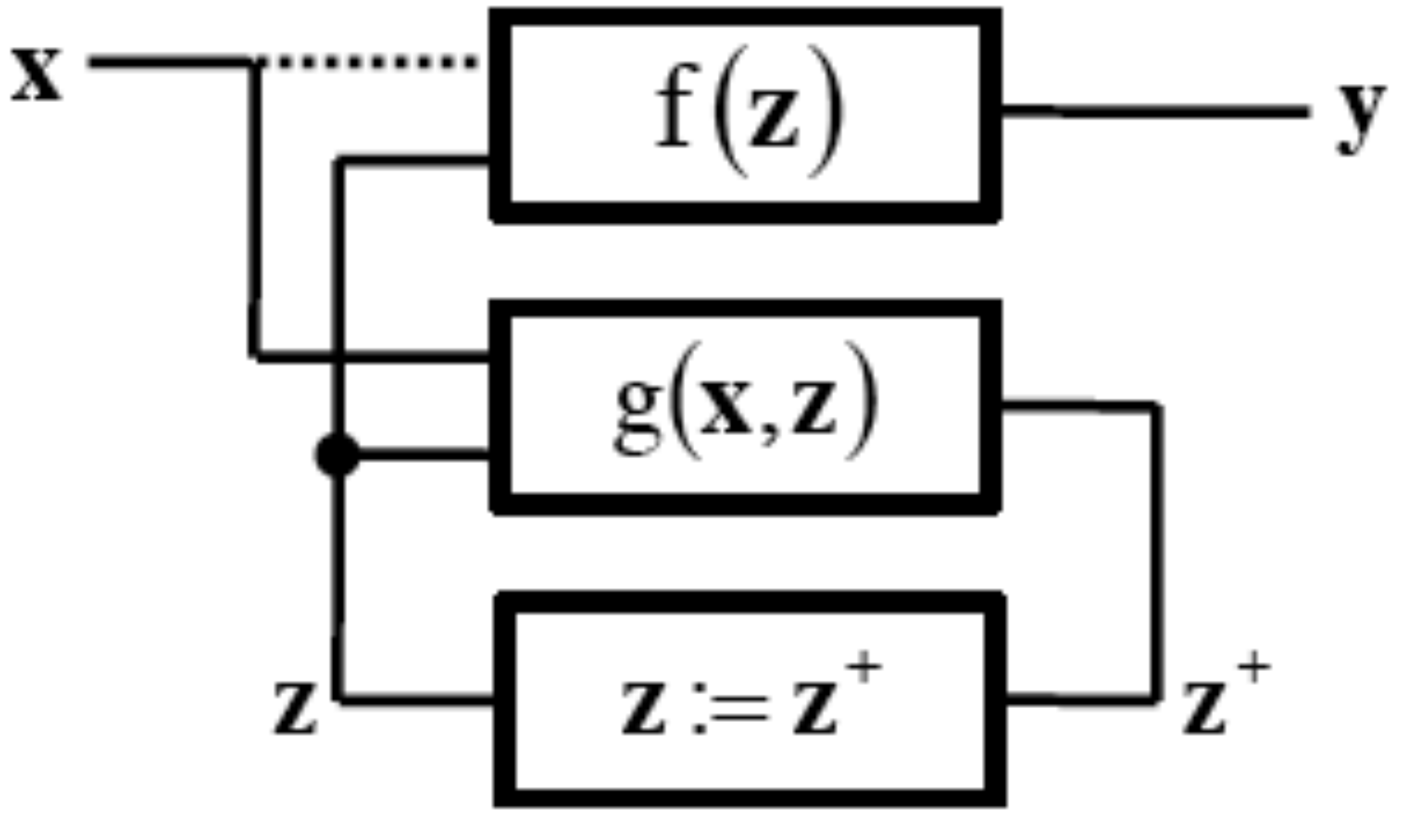


Zustandsgraph

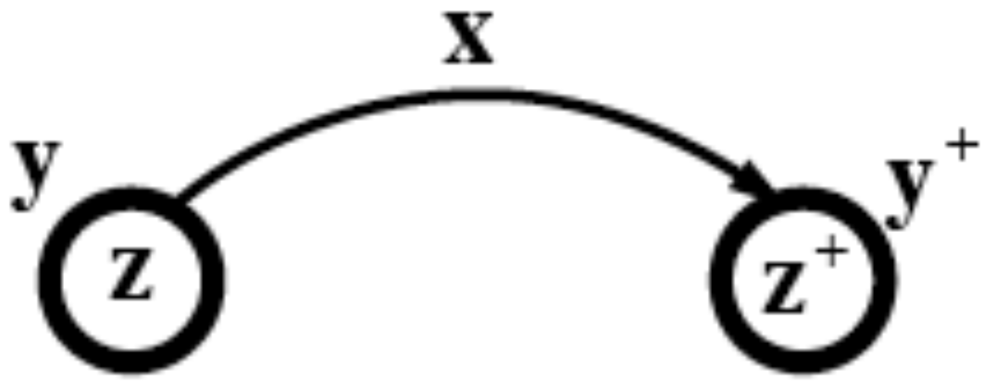
$$z^+ = g(x,z)$$
$$y = f(x,z)$$
$$z^0 = (\ )$$
$$y^+ = f(x^+, z^+) = f(x^+, g(x,z))$$

Mealy-Automaten sind **übergangsorientiert**.
Änderungen des Einganges beeinflussen sofort den Ausgang. Sie stellen die allgemeinste Form der deterministischen endlichen Automaten dar.

Moore-Automaten



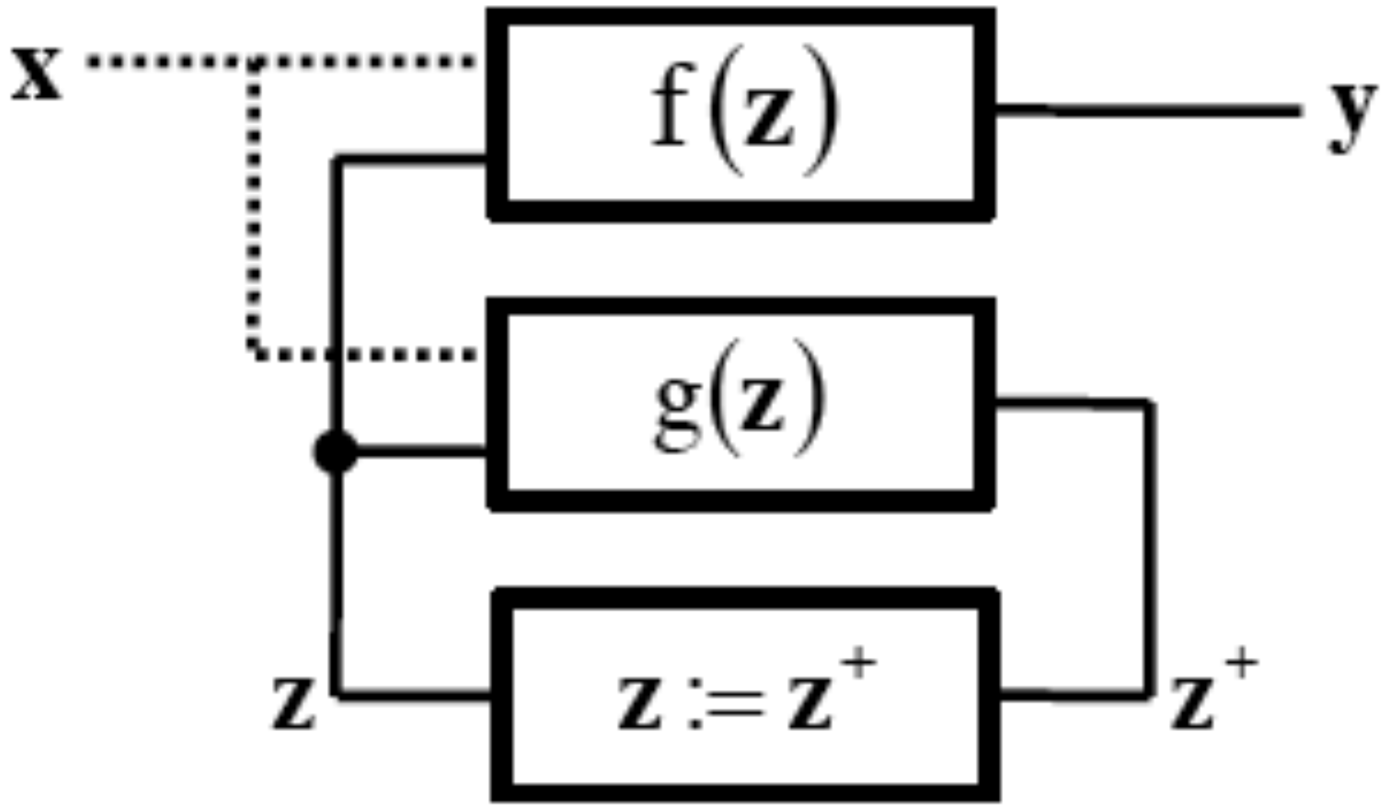
$$\mathbf{z}^+ = g(\mathbf{x},\mathbf{z})$$
$$\mathbf{y} = f(\mathbf{z})$$
$$\mathbf{z}^0 = (\ )$$
$$\mathbf{y}^+ = f(\mathbf{z}^+) = f(g(\mathbf{x},\mathbf{z}))$$



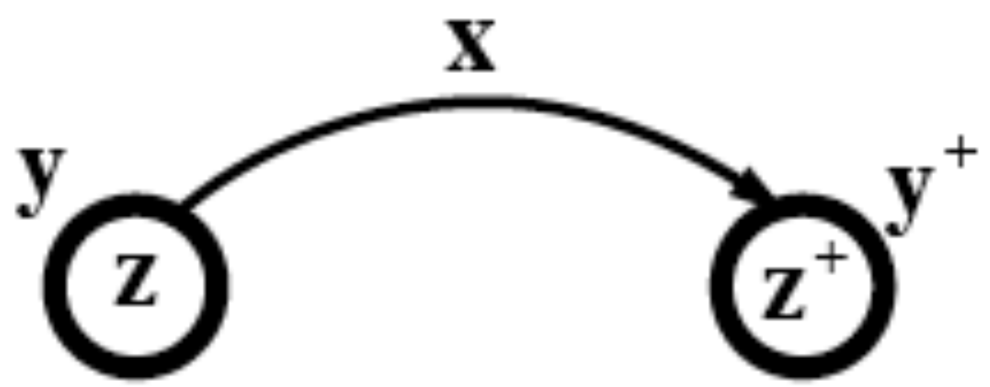
Zustandsgraph

Moore-Automaten sind **zustandsorientiert**.
Änderungen des Einganges beeinflussen den Ausgang erst zum Folgezustand. Sie stellen einen Sonderfall des Mealy-Automaten.

Autonome-Automaten



$$z^{+} = g(z)$$
$$y = f(z)$$
$$z^{0} = (\ )$$
$$y^{+} = f(z^{+}) = f(g(z))$$



Zustandsgraph
(nur eine abgehende Kante pro Knoten)

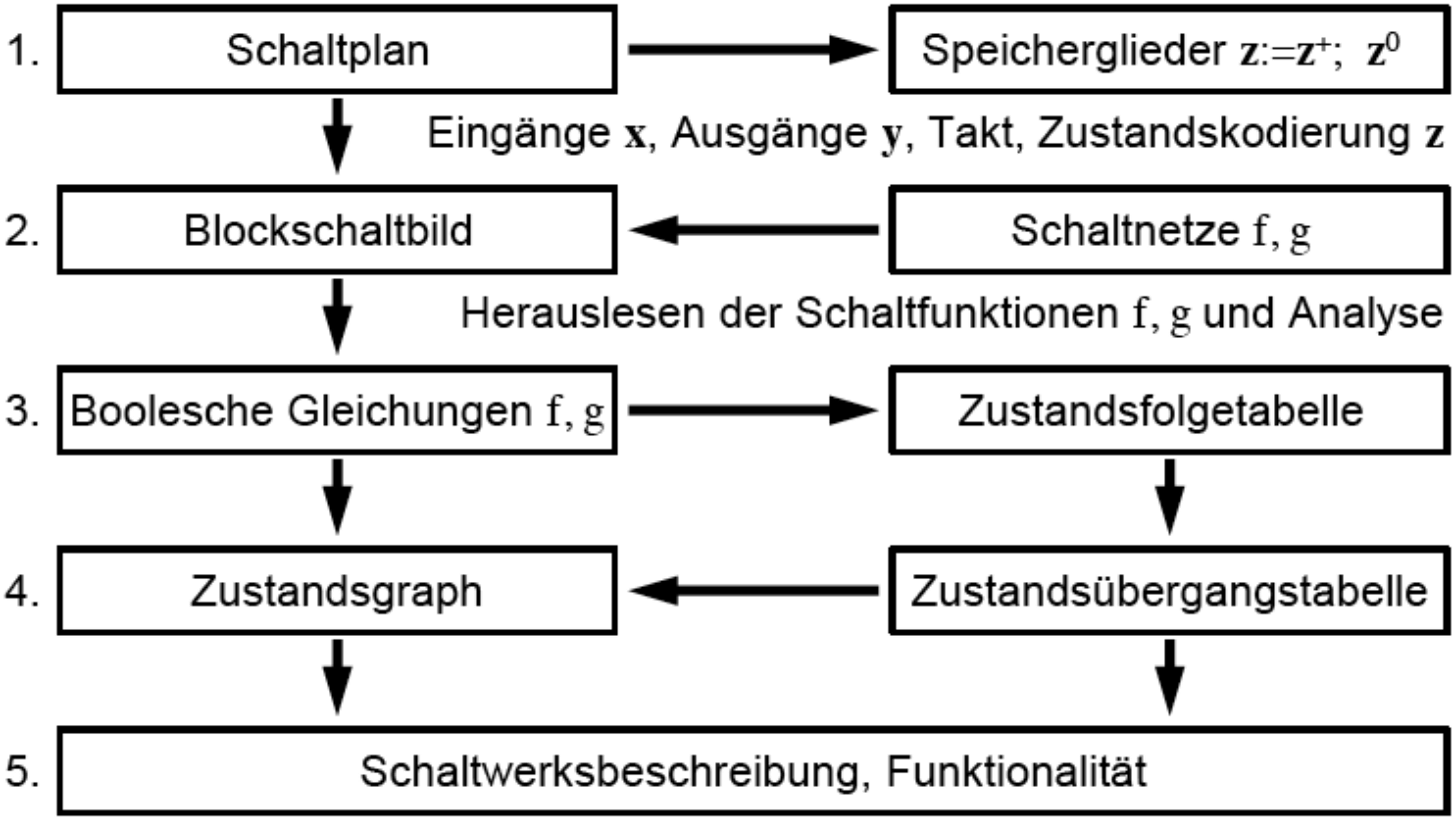
Autonome Automaten sind nicht eingangsgesteuert.
Änderungen des Einganges beeinflussen somit weder den Ausgang noch den Folgezustand. Sie stellen einen Sonderfall des Moore-Automaten und verfügen nicht über einen Eingang.

Darstellung von Schaltwerken

Es gibt verschiedene mögliche Formen der Schaltwerksdarstellung. Alle Varianten sind inhaltlich gleichwertig und können ineinander überführt werden. Bezüglich der Anschaulichkeit und Nutzbarkeit gibt es Unterschiede.

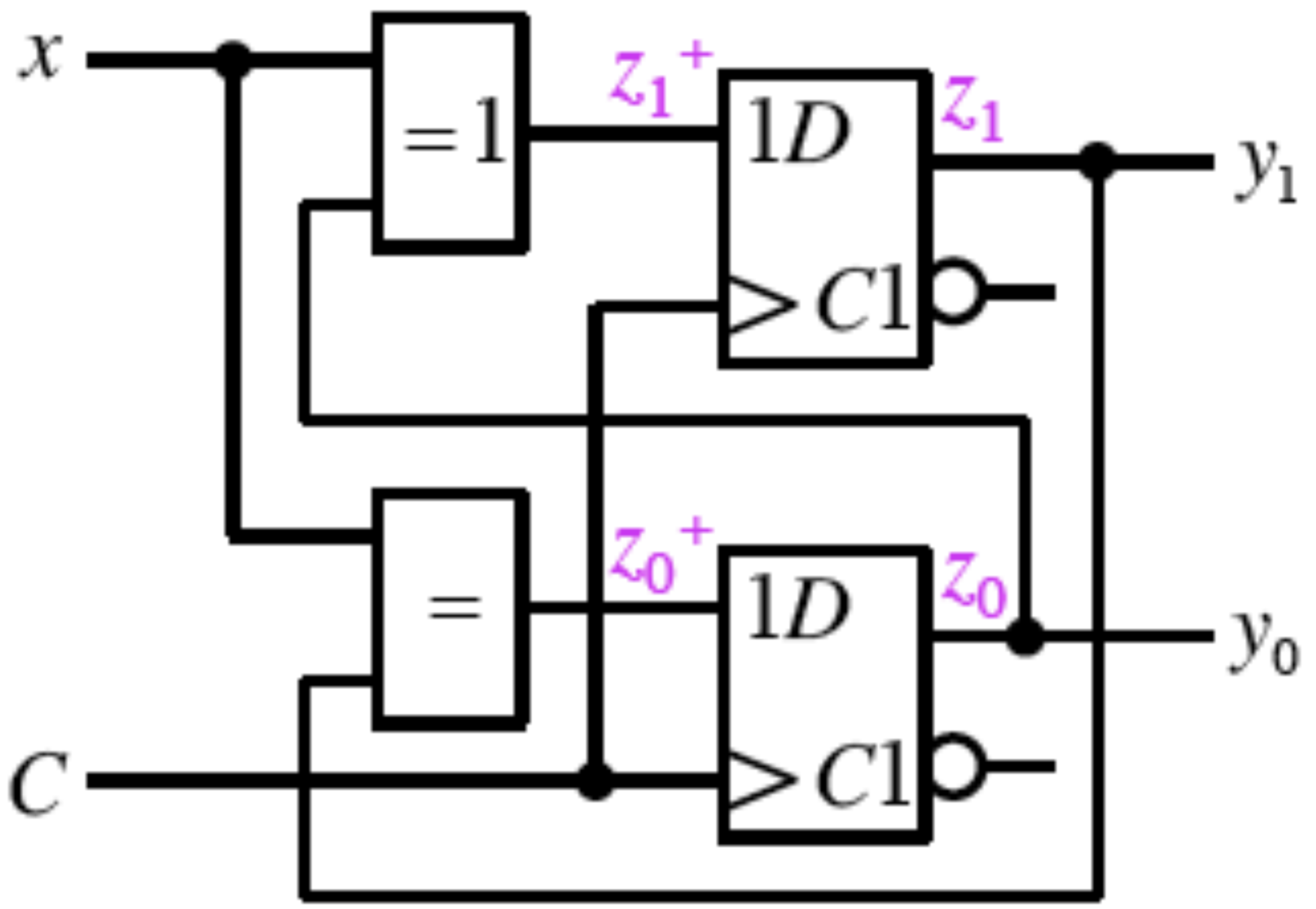
- Boolesche Gleichungen (Automat)
- Zustandsfolgetabelle (Automatentabelle)
- Zustandsübergangstabelle
- Zustandsgraph
- Impulsfolgediagramm
- Schaltplan (Logikplan)
- Schaltsymbol (DIN-Norm)
- Hardwarebeschreibungssprachen (VHDL, Verilog, SystemC, …)
- Programmiersprachen
- Binäre Entscheidungsgraphen (BDD), Speicherelemente

Analyse von Schaltwerken

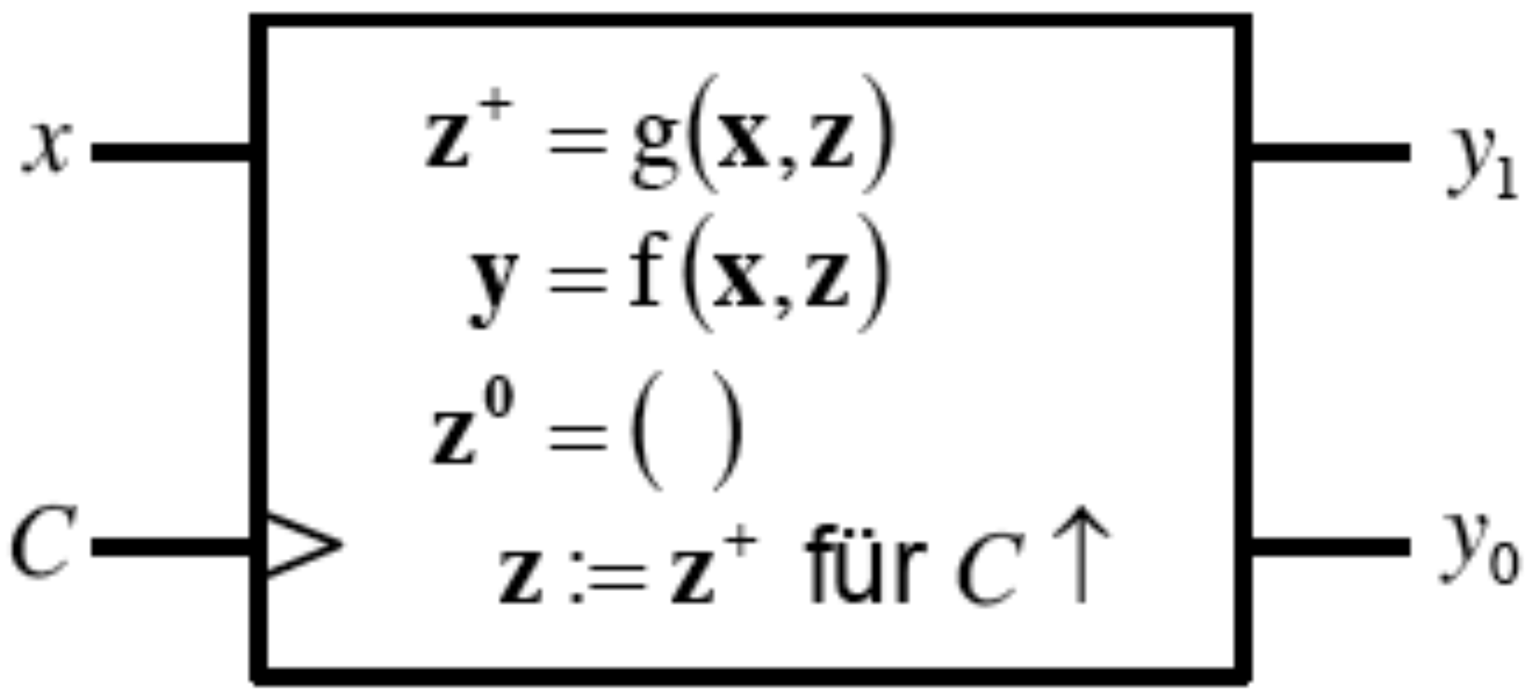


Analysebeispiel - Ansteuerung

Ausgangspunkt Schaltplan



Blockschaltbild



Boolesche Gleichungen

Zustandskodierung:

$$\mathbf{z} = (z_1, z_0)$$
$$\mathbf{z}^+ = (z_1^+, z_0^+)$$
$$\mathbf{x} = (x)$$
$$\mathbf{y} = (y_1, y_0)$$

Moore-Automat: $\mathbf{y} = f(\mathbf{z})$

$$z_1^+ = x\bar{z}_0 + \bar{x}z_0 \ ; \quad z_1^0 = 0$$
$$z_0^+ = xz_1 + \bar{x}\bar{z}_1 \ ; \quad z_0^0 = 0$$
$$y_1 = z_1$$
$$y_0 = z_0$$

Anfangszustand

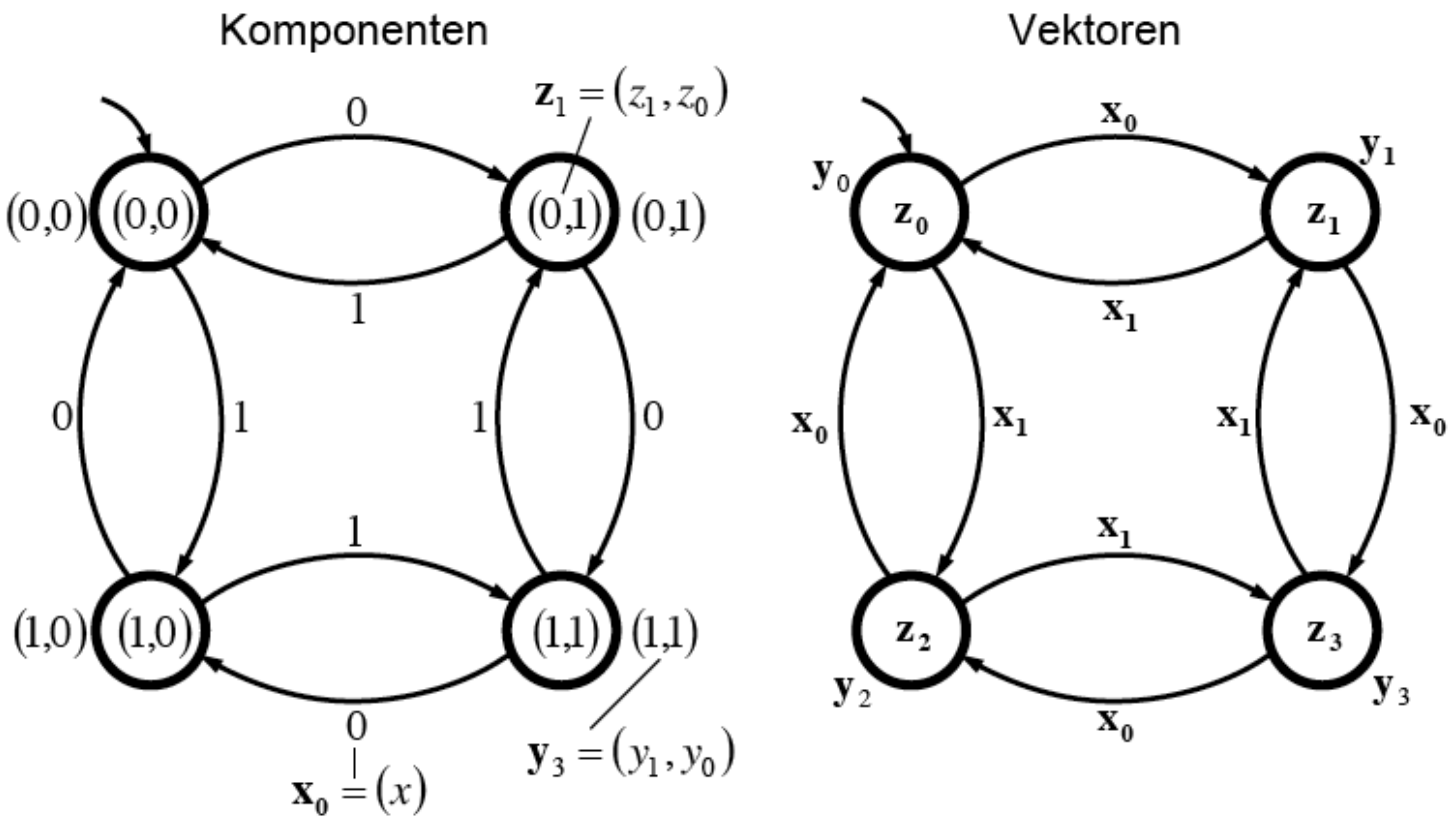
Analysebeispiel - Zustandsfolge

x	$\mathbf{x}$	z_1	z_0	$\mathbf{z}$	$\mathbf{z}^+$	z_1^+	z_0^+	y_1	y_0	$\mathbf{y}$
0	$\mathbf{x}_0$	0	0	$\mathbf{z}_0$	$\mathbf{z}_1$	0	1	0	0	$\mathbf{y}_0$
0	$\mathbf{x}_0$	0	1	$\mathbf{z}_1$	$\mathbf{z}_3$	1	1	0	1	$\mathbf{y}_1$
0	$\mathbf{x}_0$	1	0	$\mathbf{z}_2$	$\mathbf{z}_0$	0	0	1	0	$\mathbf{y}_2$
0	$\mathbf{x}_0$	1	1	$\mathbf{z}_3$	$\mathbf{z}_2$	1	0	1	1	$\mathbf{y}_3$
1	$\mathbf{x}_1$	0	0	$\mathbf{z}_0$	$\mathbf{z}_2$	1	0	0	0	$\mathbf{y}_0$
1	$\mathbf{x}_1$	0	1	$\mathbf{z}_1$	$\mathbf{z}_0$	0	0	0	1	$\mathbf{y}_1$
1	$\mathbf{x}_1$	1	0	$\mathbf{z}_2$	$\mathbf{z}_3$	1	1	1	0	$\mathbf{y}_2$
1	$\mathbf{x}_1$	1	1	$\mathbf{z}_3$	$\mathbf{z}_1$	0	1	1	1	$\mathbf{y}_3$

Anfangszustand.

$$
\begin{aligned}
\mathbf{x}_0 &= (0)\\
\mathbf{x}_1 &= (1)\\
\mathbf{y}_0 &= (0,0)\\
\mathbf{y}_1 &= (0,1)\\
\mathbf{y}_2 &= (1,0)\\
\mathbf{y}_3 &= (1,1)\\
\mathbf{z}_0 &= (0,0)\\
\mathbf{z}_1 &= (0,1)\\
\mathbf{z}_2 &= (1,0)\\
\mathbf{z}_3 &= (1,1)\\
\mathbf{z}^0 &= \mathbf{z}_0
\end{aligned}
$$

Analysebeispiel - Zustandsgraph



Analysebeispiel - Ergebnis

Das im Schaltplan gegebene Schaltwerk realisiert folgende Eigenschaften:

- zwei Speicherglieder realisieren vier verschiedene Zustände
- Moore-Automat
- folgende Zustandsfolge wird in Abhängigkeit vom Eingang realisiert:

  $x=0: \ (0,0) \to (0,1) \to (1,1) \to (1,0) \to (0,0) \to (0,1) \to \ldots$

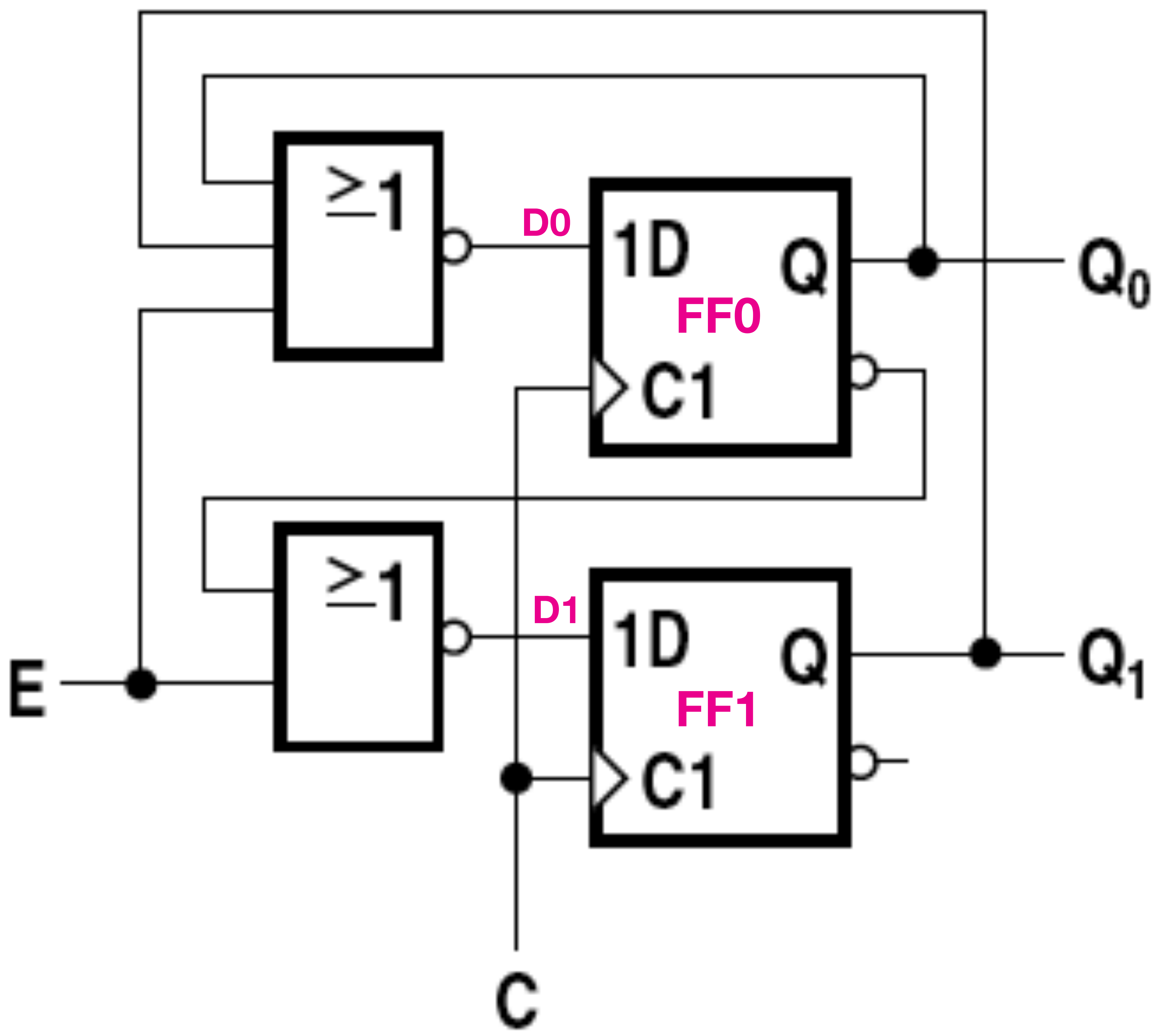
  $x=1: \ (0,0) \to (1,0) \to (1,1) \to (0,1) \to (0,0) \to (1,0) \to \ldots$
- es handelt sich um einen 2-bit Gray-Kode zyklischen Synchronzähler mit Vor- und Rückwärtssteuerung (Zustandskodierung Gray-Kode)

  $x=0:$ vorwärts zählen

  $x=1:$ rückwärts zählen
- Zählrichtung kann in jedem beliebigen Zustand umgekehrt werden
- ausgegeben wird direkt der Zählerzustand (Gray-Kode)
- Startzustand ist $\mathbf{z}^0=\mathbf{z}_0=(0,0)$.

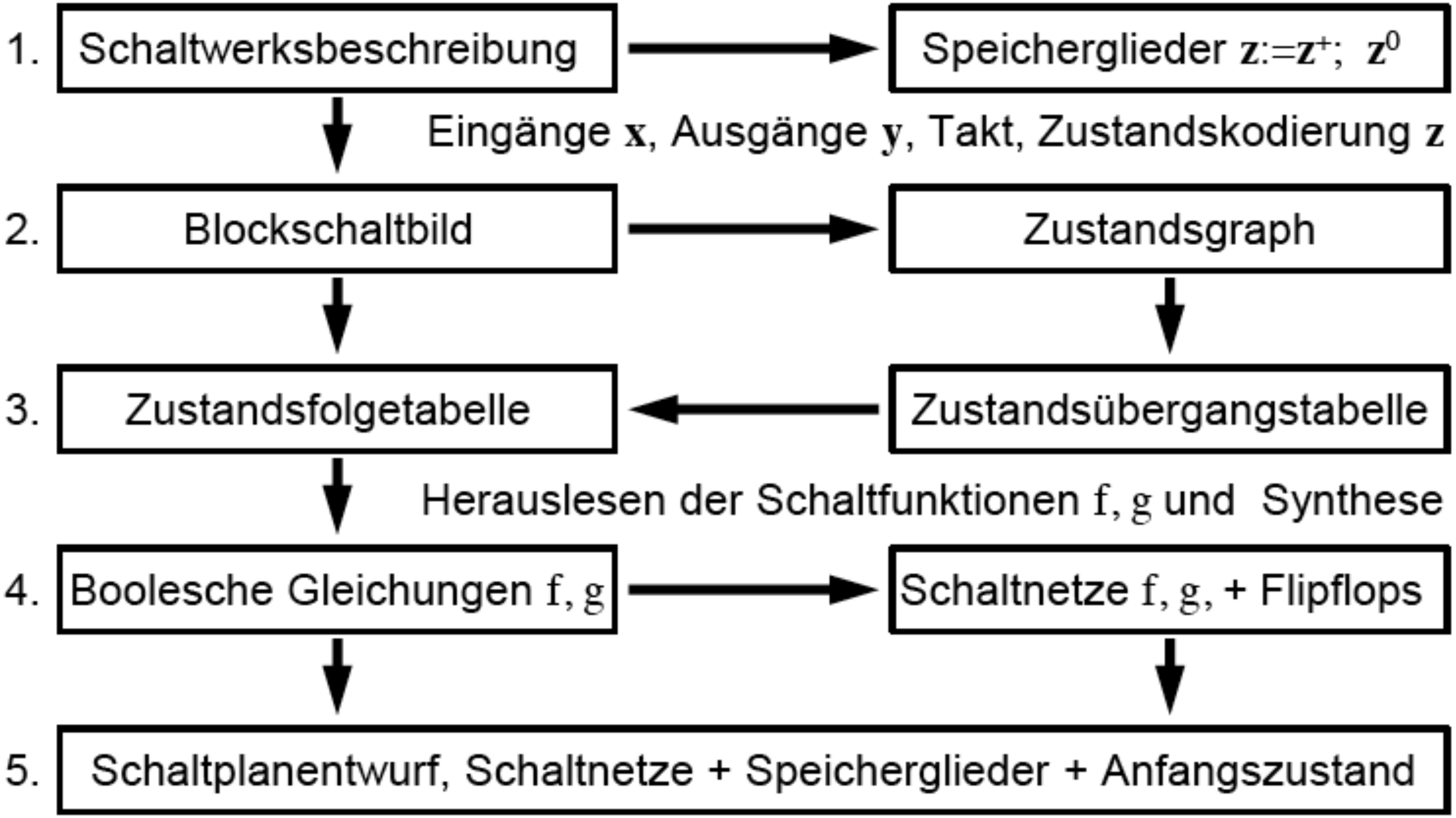
Aufgaben — Synthese und Analyse von Schaltwerken I

6. Gegeben sei folgende Schaltung:



a) Ermitteln Sie die Ansteuerfunktionen für die FlipFlops.
b) Stellen Sie die Zustandsübergangstabelle auf.
c) Zeichnen Sie den Zustandsgraphen.
d) Geben Sie eine geeignete Bezeichnung für die Schaltung an.
e) Welcher Typ (Art der Ausgänge) liegt vor?

Synthese von Schaltwerken



Aufgaben — Synthese und Analyse von Schaltwerken II

!

7. Zur Überwachung einer Signalleitung **x** soll ein Zustandsautomat eingesetzt werden. Der erwartete Signalverlauf ist dabei die strikt alternierende Folge von Nullen (0) und Einsen (1). Eine beliebige Abweichung von diesem Muster ist durch einen Impuls auf einem Fehlerindikationssignal **F** anzuzeigen.

   a) Wie wird eine Fehler identifiziert? Die unterstellte Fehlerzahl bei einer Abweichung soll möglichst gering sein. So weist etwa die Folge **0101000101** einen **Bitflip** von 1 nach 0 und nicht zwei Einschübe von 0 auf.

   b) Entwickeln Sie den Zustandsübergangsgraphen.

   c) Stellen Sie die Zustandsübergangstabelle auf. Es stehen D-FF mit einem gesonderten Rücksetzeingang zur Verfügung.

   d) Welcher Typ (Art der Ausgänge) liegt vor?

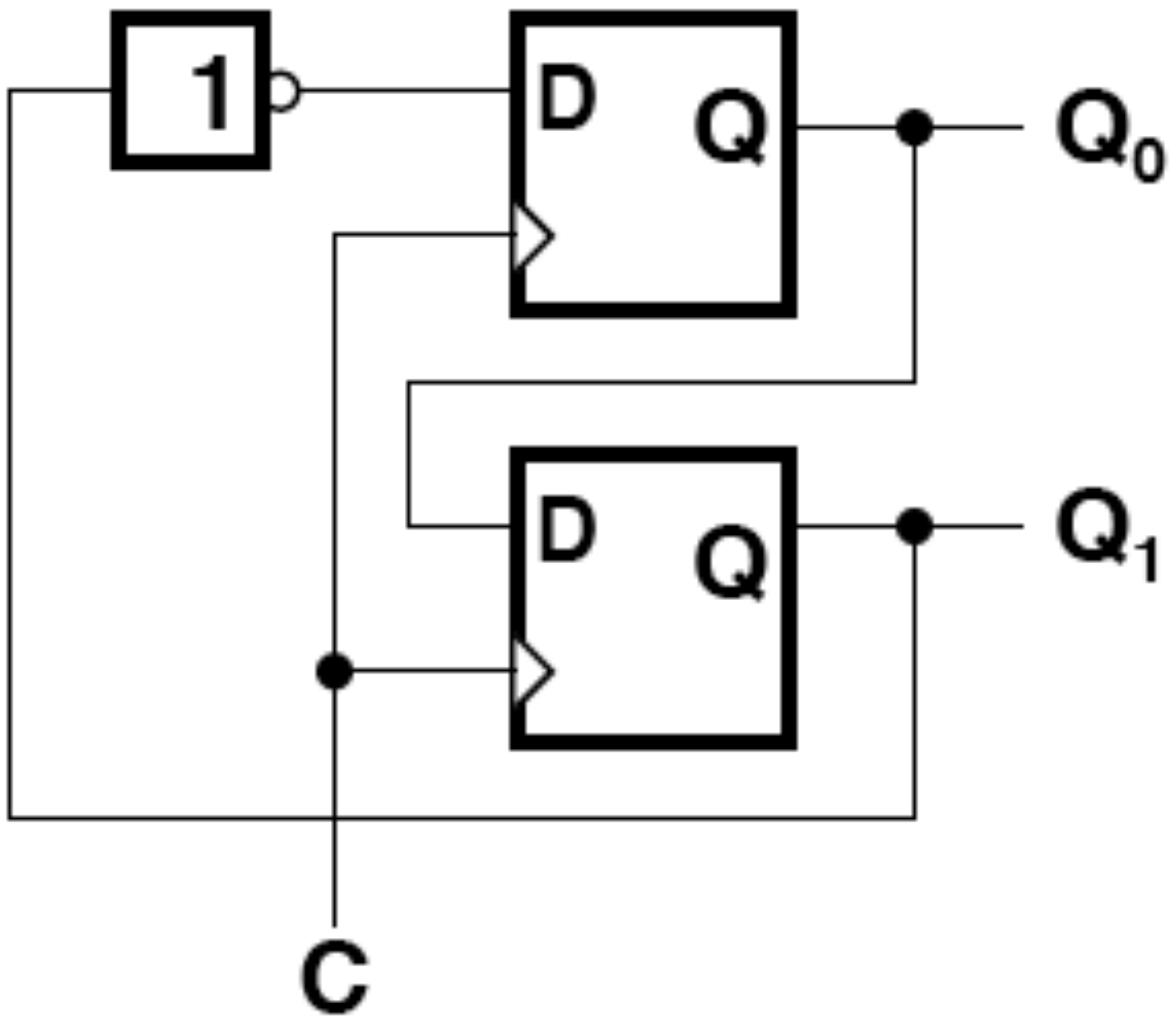
Aufgaben — Synthese und Analyse von Schaltwerken III

8. Ein getakteter serieller Datenbus **x** führt im Ruhezustand einen High-Pegel. Die Datenübertragung wird durch einen Low-Impuls einer Taktlänge eingeleitet und umfasst jeweils genau die vier Bits der direkt darauf folgenden Takte.

   Entwerfen Sie einen den Zustandsgraphen für einen Automaten, der den Abschluss der Übertragung (**E**) eines solchen Nibbles und dann auch dessen Parität (**P**) signalisiert.

Aufgaben — Synthese und Analyse von Schaltwerken IV

9. Gegeben sei folgende Schaltung:



a) Ermitteln Sie die Ansteuerfunktionen für die FlipFlops.

b) Welcher Typ (Art der Ausgänge) liegt vor?

c) Stellen Sie die Zustandsübergangstabelle auf.

d) Zeichnen Sie den Zustandsgraphen.

e) Nach welchem Code zählt dieses Schaltwerk?


END