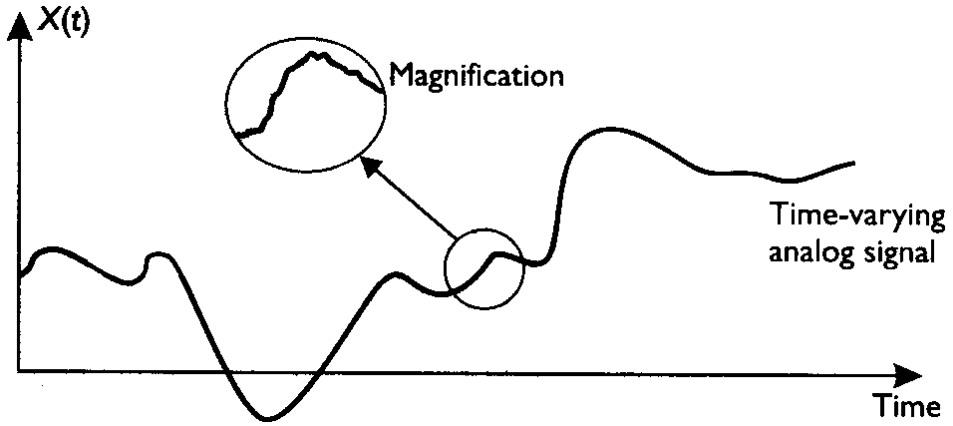
Schaltalgebra und kombinatorische Logik

1. Analog- und Digitaltechnik
2. Digitale elektrische Schaltungen
3. Logische Schaltungen
4. Boolesche Algebra
5. Schaltfunktionen
6. Synthese von Schaltungen
7. Minimierung von Schaltfunktionen
8. Schaltnetze

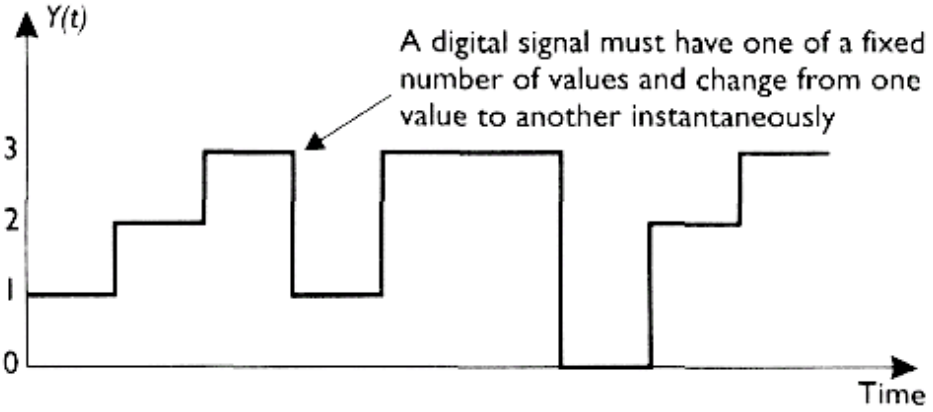
*Die Folien wurden unter Verwendung der Folien von A. Strey, Ulm, SS 03 erstellt

Analogtechnik und Digitaltechnik

- bei analoger Technik kontinuierliche Signale



- bei digitaler Technik diskrete, i.a. binäre Signale



(Bilder aus: Clements, The Principles of Computer Hardware)

Analogrechner: Meda 42A + Meda 42B

Hersteller: Aritma AT, Prag, Baujahr: ab 1968,
Serienfertigung bis Ende der 70er Jahre

Prof. M. Ludwig (TU Dresden), T. Falk

Die Eingabe erfolgte durch Stecken des Analogprogramms mittels Programmierschnüre, Kurzschlussstecker und Rechenimpedanzen (Widerstände für die Summatoren und Integratoren) auf einer Programmiertafel; die Einstellung der Konstantenpotentiometer wurde durch ein Digitalvoltmeter unterstützt.

Zur Auswertung stand zur Verfügung:
* 6-Strahl-Oszilloskop OPD 280 U
* X-Y-Schreiber BAK 5 T
* Digitalvoltmeter

http://www.uni-greifswald.de/~wwwmathe/RTS/gg20053.html

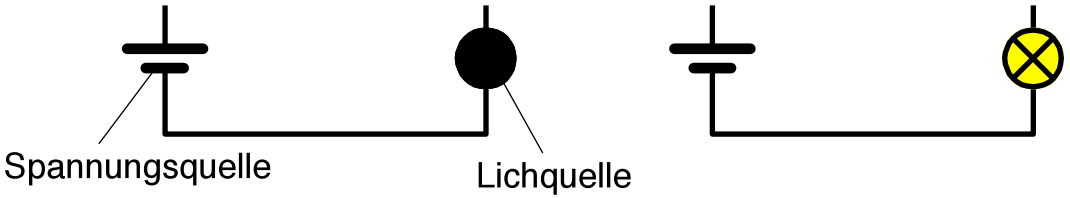
Analogtechnik und Digitaltechnik

- einige Vor- und Nachteile analoger Hardware:
  - \+ Multiplikation und Addition leicht zu realisieren
  - \+ vergleichsweise schnell
  - – temperaturabhängig
  - – nichtlineare Bauteile
  - – Präzision nur bei ca. 6-8 bit
  - – Langzeitspeicherung von Daten schwierig
- einige Vor- und Nachteile digitaler Hardware:
  - \+ unempfindlich gegenüber Störungen (z.B. Rauschen)
  - \+ einfacher Entwurf
  - \+ beliebig hohe Präzision möglich
  - – vergleichsweise hoher Energieverbrauch

⇒ Digitaltechnik ermöglicht eine einfache Realisierung robuster Hardware

Digitale elektrische Schaltungen

- eine einfache Schaltung:



- Strom fließt nur bei geschlossenem Stromkreis !
- zwei Zustände
  - Strom fließt ⇒ Lampe leuchtet
  - Strom fließt nicht ⇒ Lampe leuchtet nicht

Digitale elektrische Schaltungen

- eine einfache Reihenschaltung:

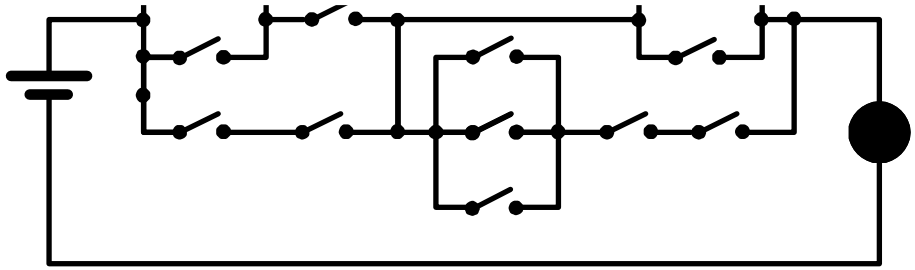
⇒ Lampe leuchtet, wenn der erste und der zweite Schalter geschlossen werden

- eine einfache Parallelschaltung:

⇒ Lampe leuchtet, wenn der erste oder der zweite Schalter geschlossen wird

Digitale elektrische Schaltungen

- eine komplexe(re) Schaltung:



⇒ wann leuchtet die Lampe ?

⇒ wie kann man eine solche Schaltung beschreiben ?

Beschreibung durch logische Ausdrücke

- bei Betrachtung aus Sicht der Logik ergeben sich 2 Zustände:
  - Strom fließt / Strom fließt nicht
  - Lampe leuchtet / Lampe leuchtet nicht
  - an / aus
  - wahr / falsch
  - 1 / 0
  - 0 / 1
- im folgenden Betrachtung der Zustandsmenge {0,1}
- Zustände werden elektronisch realisiert:
  - Stromfluß / kein (oder ein sehr geringer) Stromfluß
  - Spannung / keine (oder eine sehr geringe) Spannung

Beschreibung durch logische Ausdrücke

Wie können logische Schaltungen beschrieben werden?
Welche logischen Grundverknüpfungen können identifiziert werden?

Logische Variablen nehmen den Wert „wahr“ oder „falsch“ an.
Andere Werte gibt es nicht: „Tertium non datur“

Verknüpfung von Variablen:

A	B	A UND B
falsch	falsch	falsch
falsch	wahr	falsch
wahr	falsch	falsch
wahr	wahr	wahr

Wahrheitstafel

Entwicklung:

- Aristoteles 384-322 v.Chr. begründet „Syllogistik“ Lehre von den logischen Schlußformen
- Später bilden die Stoiker die Syllogistik als Aussagenlogik weiter aus.
- Im Mittelalter → Scholastik
- George Boole (1815-1864) → 1854 mathematische Formalisierung in „An Investigation of the Laws of Thought on which are founded the Mathematical Theories of Logic and Probabilities".

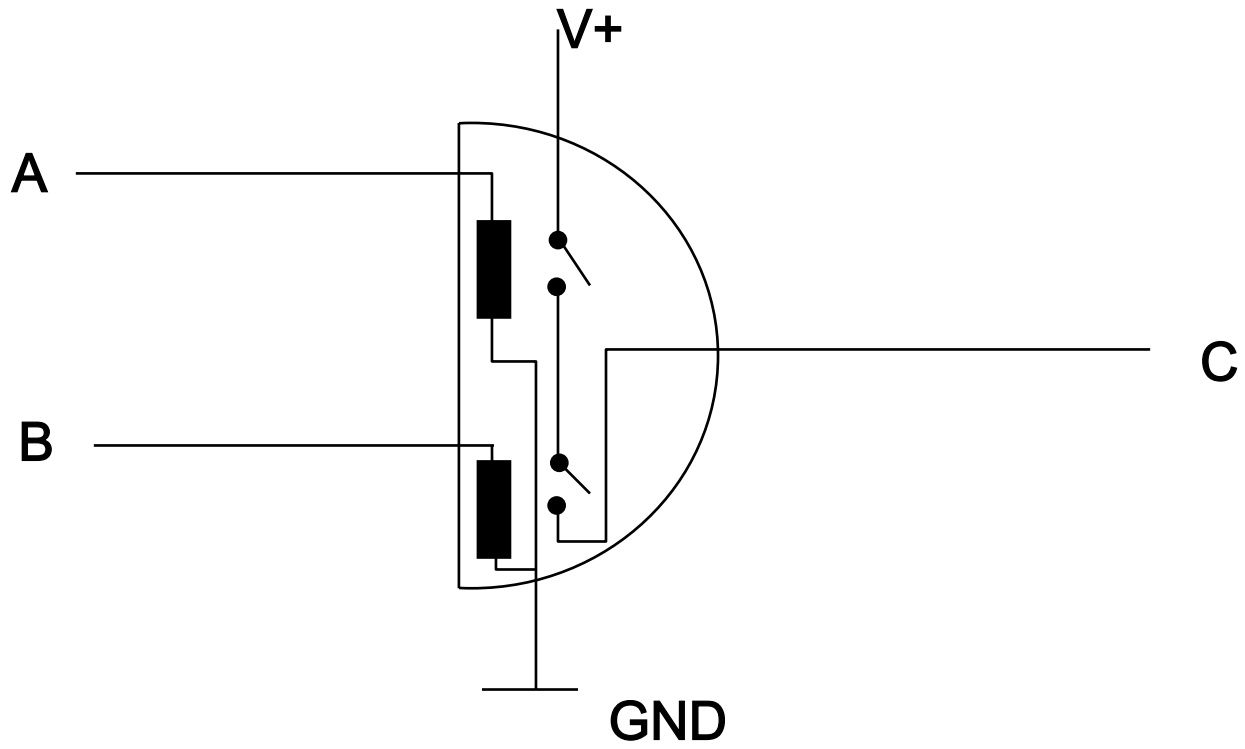
- Claude Shannon (1916-2001) hat im Rahmen seiner Diplomarbeit: „On the Symbolic Analysis of Relay and Switching Circuits (1940)“, gezeigt, dass man die Boolsche Algebra zur Beschreibung von Schaltkreisen anwenden kann.

Bild: http://de.wikipedia.org/wiki/Claude_Shannon

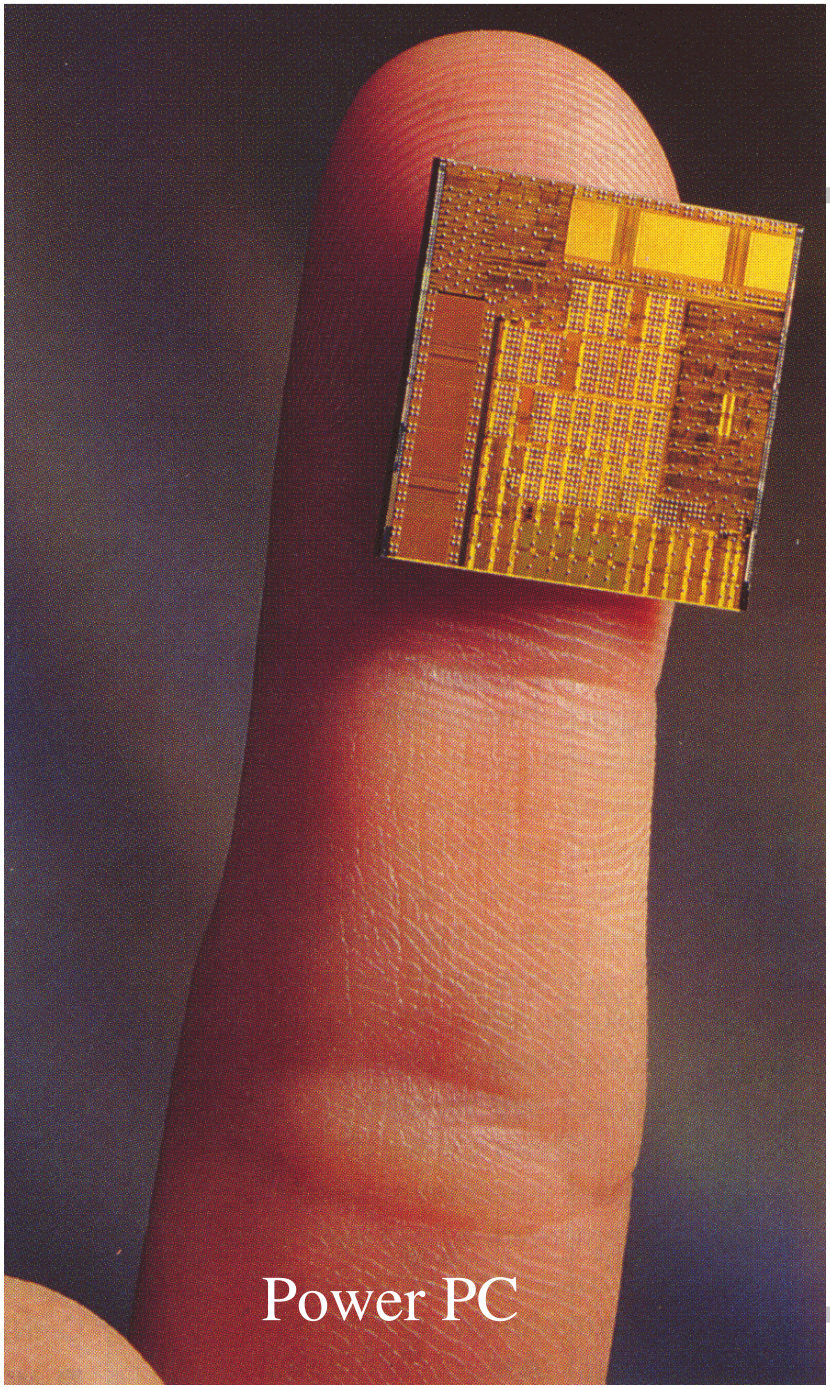
Wie realisiert man die Funktionen, die durch eine Wahrheitstafel beschrieben werden?

Realisierung einer UND-Funktion durch Gatter mit Relais:

A	B	A UND B
0	0	0
0	1	0
1	0	0
1	1	1



- elementare Funktionen auf {0,1} werden durch Gatter realisiert,
- komplexe Funktionen durch eine geeignete Verschaltung von mehreren Gattern

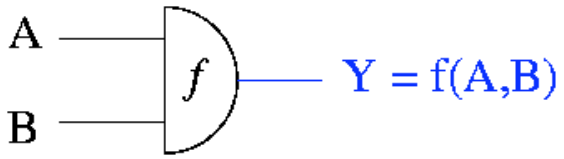


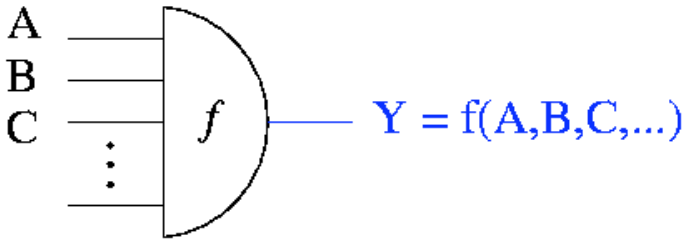
Realisierung durch
elektronische Schaltkreise:

klein
sparsam
schnell
zuverlässig
billig

Logische Schaltungen

- ein Gatter ist eine (elektrotechnische) „*Black Box*“ mit einem, zwei oder mehreren Eingängen A,B,C,... ∈ {0,1} und genau einem Ausgang Y∈{0,1} zur Realisierung einer Funktion Y = *f* (A,B,C,...)
  (⇒ die elektronische Realisierung tritt nun in den Hintergrund!)





A	B	Y
0	0	
0	1	
1	0	
1	1	

A	B	C	...	Y
0	0	0		
0	0	1		
⋮	⋮	⋮		
1	1	1		

- eine Wahrheitstabelle legt jeweils die Funktion fest.

Elementare Gatter

- ein UND-Gatter realisiert eine Konjunktion:
  - für zwei Eingänge:
    $Y=A \cdot B$
  - für *k* Eingänge:
    $Y=A_1 \cdot A_2 \cdot \ldots \cdot A_k$

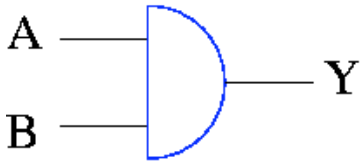
- Andere äquivalente Notationen:

  $Y = A \wedge B$

  $Y = AB$

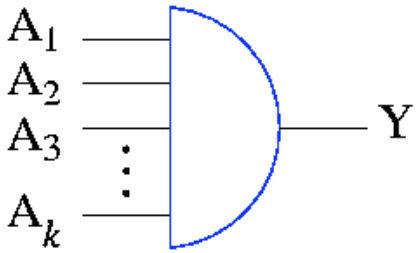
  $Y = A \,\&\, B$

für 2 Eingänge:

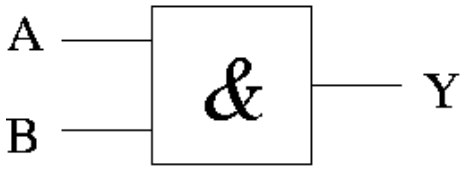


A	B	Y=A·B
0	0	0
0	1	0
1	0	0
1	1	1

für *k* Eingänge:



andere Darstellung:



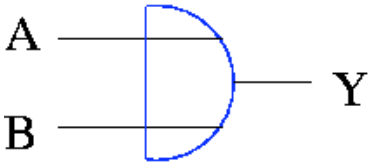
Elementare Gatter

- ein ODER-Gatter realisiert eine Disjunktion:
  - für zwei Eingänge: Y=A+B,
  - für *k* Eingänge: $Y=A_1+A_2+ \ldots +A_k$

- andere äquivalente Notationen:

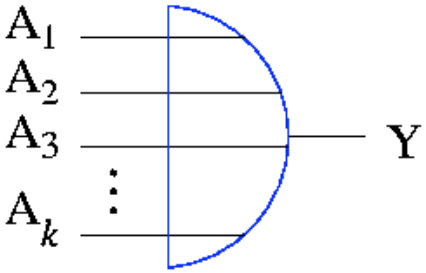
  $Y=A \vee B$

für 2 Eingänge:

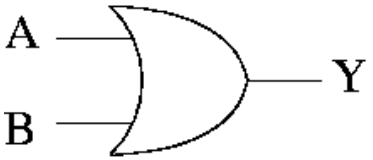


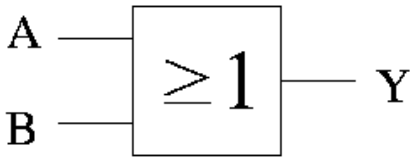
A	B	Y=A+B
0	0	0
0	1	1
1	0	1
1	1	1

für *k* Eingänge:



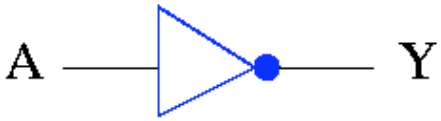
andere Darstellungen:





Elementare Gatter

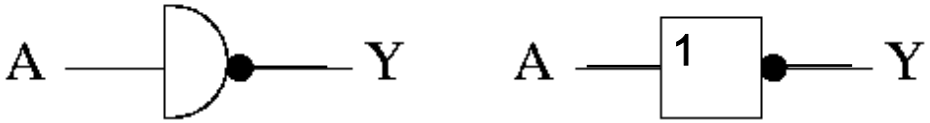
- ein NICHT-Gatter realisiert eine Negation:
  $Y=\overline{A}$



A	$Y=\overline{A}$
0	1
1	0

- andere äquivalente Notationen:
  Y=A'
  Y= ¬A

andere Darstellungen:



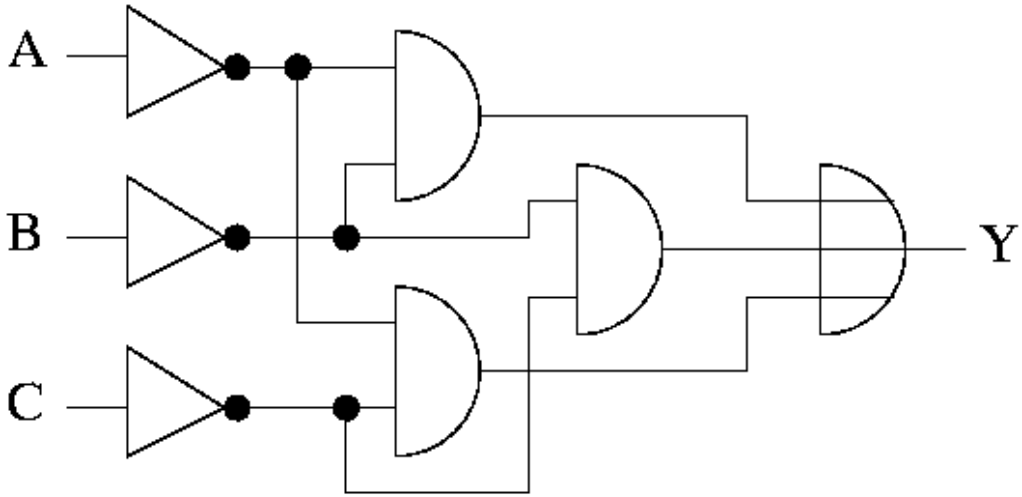
Beispiel einer logischer Schaltung

- Gesucht ist eine Schaltung, die eine logische 1 generiert, wenn höchstens einer von drei Eingängen A,B,C den Wert 1 aufweist.
- erste Lösungsvariante:

Wahrheitstabelle:

A	B	C	Y
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0

Realisierung:



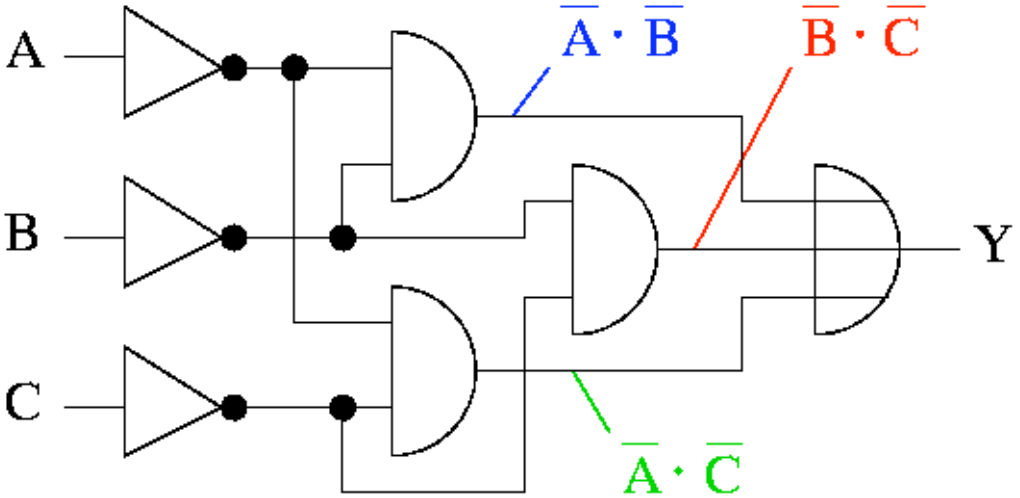
Beispiel einer logischen Schaltung

- Gesucht ist eine Schaltung, die eine logische 1 generiert, wenn höchstens einer von drei Eingängen A,B,C den Wert 1 aufweist.
- erste Lösungsvariante:

Wahrheitstabelle:

A	B	C	Y
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0

Realisierung:



$$Y = \overline{A} \cdot \overline{B} + \overline{B} \cdot \overline{C} + \overline{A} \cdot \overline{C}$$

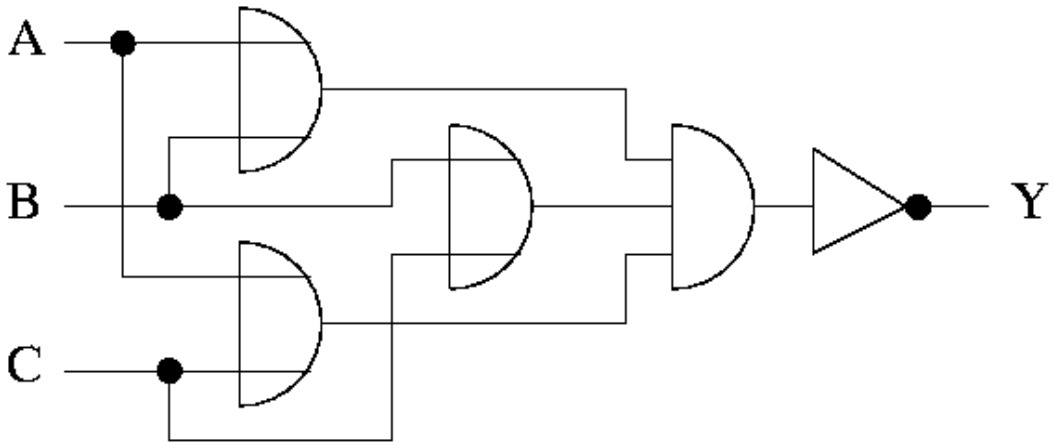
Beispiel einer logischer Schaltung

- zweite Lösungsvariante:

Wahrheitstabelle:

A	B	C	Y
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0

Realisierung:



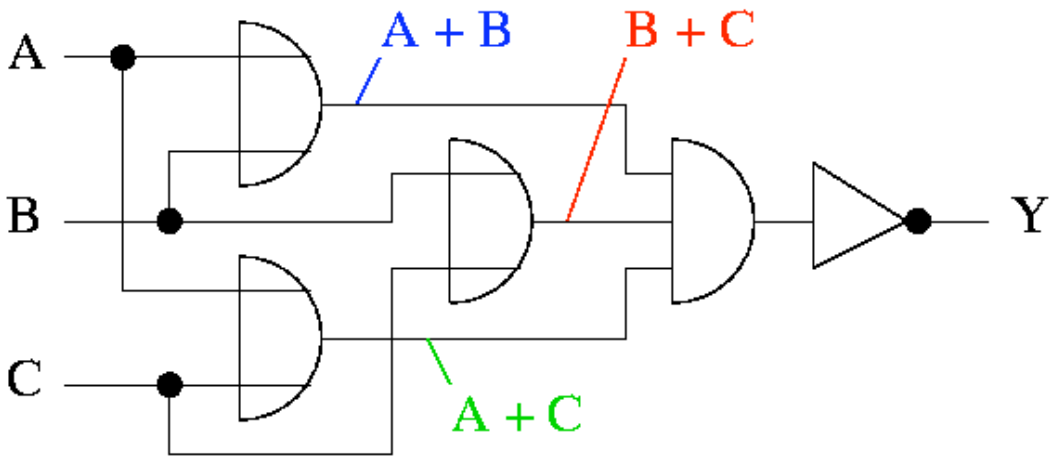
Beispiel einer logischer Schaltung

- zweite Lösungsvariante:

Wahrheitstabelle:

A	B	C	Y
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0

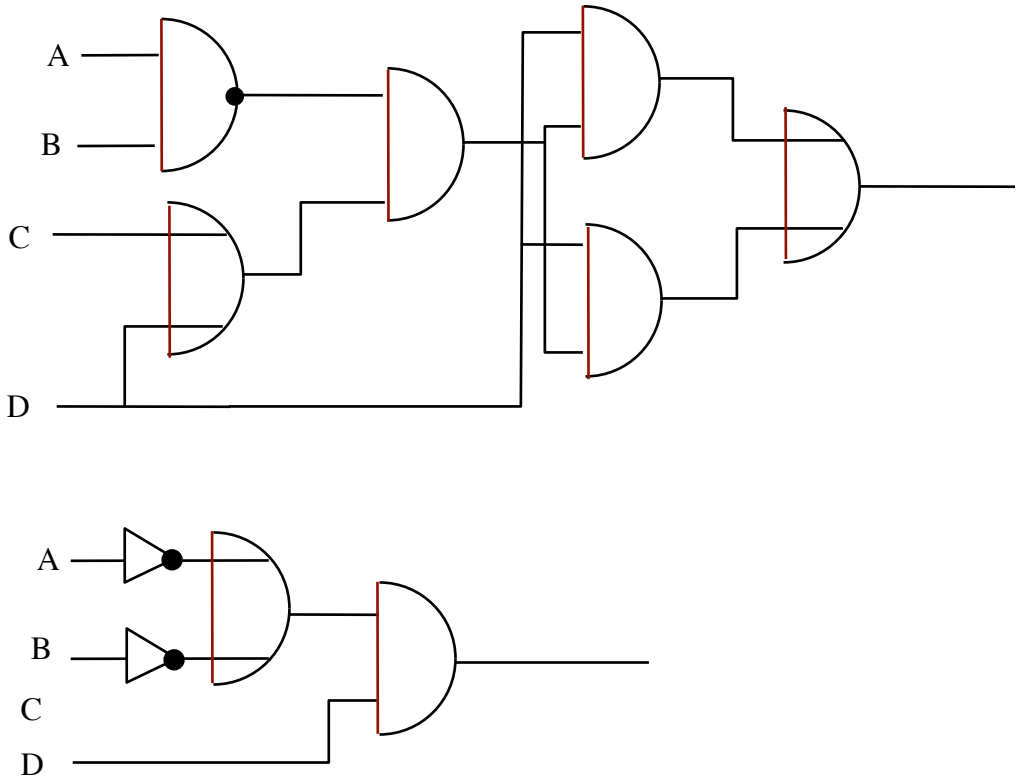
Realisierung:



$$Y = \overline{(A + B) \cdot (B + C) \cdot (A + C)}$$

- welche Variante ist besser ?

Äquivalenz von Schaltnetzen



??

Boolesche Algebra

- **Problem:** Gibt es ein Verfahren,
  - um die Äquivalenz zweier Schaltungen formal nachzuweisen ?
  - um Schaltungen auf einfache Weise zu transformieren ?
  - um minimale Schaltungen zu entwerfen ?

- **Lösung:** **Boolesche Algebra**
  - von G. Boole im Jahre 1854 entwickelt
  - zwei Werte: 0 und 1
  - drei Boolesche Operationen: $+$, $\cdot$ sowie " $\overline{\ }$ "
  - vier Axiome ...

Boolesche Algebra

Eine Algebra besteht aus

- **einer Tägermenge,**
- **Operationen über dieser Menge, für die bestimmte Axiome gelten**
- **neutralen Elementen für die Operationen.**

- **Sei B eine Menge und seien 0,1 Elemente aus B.**
- **Seien zweistellige Operationen • und + , sowie eine einstellige Operation $\bar{}$ auf B erklärt.**
- **Die neutralen Elemente seien: 1 für die Operation "•" und 0 für die Operation "+".**

Dann heißt (B, • ,+ , $\bar{}$,0 ,1) eine *Boolsche Algebra* , wenn für beliebige X, Y, Z ∈ B folgende Axiome gelten:

Axiome der Booleschen Algebra

- **Kommutativität:** $A+B = B + A$
  $A \cdot B = B \cdot A$
- **Assoziativität:** $A+(B+C) = (A+B)+C = A+B+C$
  $A \cdot (B \cdot C) = (A \cdot B) \cdot C = A \cdot B \cdot C$
- **Distributivität:** $A \cdot (B+C) = (A \cdot B)+(A \cdot C)$
  $A+(B \cdot C) = (A+B) \cdot (A+C)$
- **Neutrales Element:** $\mathbf{0} + A = A$
  $\mathbf{1} \cdot A = A$
- **Komplementarität:** $A + \overline{\mathbf{A}} = 1$
  $A \cdot \overline{\mathbf{A}} = 0$

Sätze der Booleschen Algebra

folgende Sätze können aus den vier Axiomen abgeleitet werden:

- **Idempotenzgesetze:** $A+A = A$
  $A \cdot A = A$
- **Substitutionsregeln:** $A+1 = 1$
  $A \cdot 0 = 0$
- **Absorptionsgesetze:** $A+(A \cdot B) = A$
  $A \cdot (A+B) = A$
- **Doppelnegation:** $\overline{\overline{A}} = A$

Sätze der Booleschen Algebra

- **Komplementäre Werte:** $\overline{0} = 1$

  $\overline{1} = 0$

- **Tertium non datur:** wenn $A \neq 0$, dann gilt $A = 1$.

  wenn $A \neq 1$, dann gilt $A = 0$.

- **Abgeschlossenheit:** Boolesche Operationen liefern nur boolesche Werte als Ergebnis.

Gesetze der Booleschen Algebra

$0 + A = A$	$1 \cdot A = A$	
$1 + A = 1$	$0 \cdot A = 0$	
$A + A = A$	$A \cdot A = A$	
$A + \overline{A} = 1$	$A \cdot \overline{A} = 0$	
$A = \overline{\overline{A}}$		
$A + B = B + A$	$A \cdot B = B \cdot A$	Kommutativges.
$A + (B + C) = (A + B) + C$	$A \cdot (B \cdot C) = (A \cdot B) \cdot C$	Ass. Ges.
$A \cdot (B + C) = A \cdot B + A \cdot C$	$A+(B \cdot C) = (A+B) \cdot (A+C)$	Distr. Ges.
$A + (A \cdot B) = A$	$A \cdot (A + B) = A$	Absorptionsges.
$A + (\overline{A} \cdot B) = A + B$		

Das Dualitätsprinzip der Booleschen Algebra

$0 + A = A$	$1 \cdot A = A$
$1 + A = 1$	$0 \cdot A = 0$
$A + A = A$	$A \cdot A = A$
$A + \overline{A} = 1$	$A \cdot \overline{A} = 0$

Zahlenalgebra:

$A \cdot B + A \cdot C = A \cdot (B + C)$

$(A+B)\cdot(A+C) = A\cdot A + A\cdot C + B\cdot A + B\cdot C$

$A+(B+C) = (A+B)+C = A+B+C$

$A \cdot (B \cdot C) = (A \cdot B) \cdot C = A\cdot B\cdot C$

$A+(B+A) = (A+B)+A = 2\cdot A + B$

$A\cdot (B+A) = (A\cdot B)+(A\cdot A) = A^2 + A\cdot B$

Boolesche Algebra:

$A \cdot B + A \cdot C = A \cdot (B + C)$ Distributivges.

$(A+B) \cdot (A+C) = A + (B \cdot C)$ Distributivges.

$A+(B+C) = (A+B)+C = A+B+C$ Assoziativges.

$A \cdot (B \cdot C) = (A \cdot B) \cdot C = A\cdot B\cdot C$

$A+(B+A) = (A+B)+A = A+B$

$A\cdot(B+A) = A+(B\cdot A) = A$ Absorption

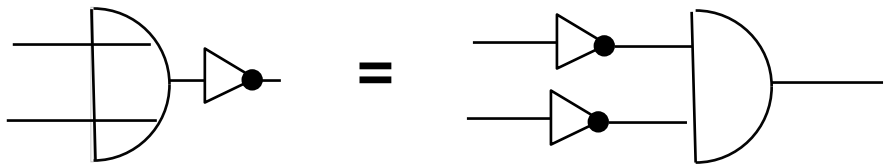
Sätze der Booleschen Algebra

Die de Morgan‘schen Regeln

A	B	A + B	$\overline{A + B}$
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

A	B	$\overline{A}$	$\overline{B}$	$\overline{A} \cdot \overline{B}$
0	0	1	1	1
0	1	1	0	0
1	0	0	1	0
1	1	0	0	0

$$\overline{A+B} = \overline{A} \cdot \overline{B}$$



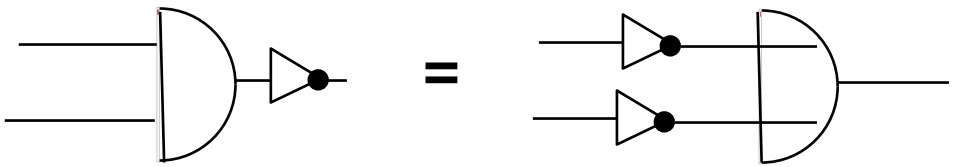
Sätze der Booleschen Algebra

Die de Morgan‘schen Regeln

A	B	$A \cdot B$	$\overline{A \cdot B}$
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

A	B	$\overline{A}$	$\overline{B}$	$\overline{A}+\overline{B}$
0	0	1	1	1
0	1	1	0	1
1	0	0	1	1
1	1	0	0	0

$$\overline{A \cdot B} = \overline{A} + \overline{B}$$



Sätze der Booleschen Algebra

- **de Morgansche Regeln:** $\overline{A+B} = \overline{A} \cdot \overline{B}$

  $\overline{A \cdot B} = \overline{A}+\overline{B}$

  Anmerkung: de Morgansche Regeln ermöglichen eine einfache Negation von Termen durch

  1) Tausch der Operationen + (ODER) und · (UND)

  2) Komplementierung aller Variablen

de Morgan‘s Regel erlaubt es, negierte Ausdrücke in einzelne negierte Ausdrücke aufzulösen und Operatoren wechselseitig zu ersetzen.

Regeln zur Umformung Boolscher Gleichungen:

Für zwei Gleichungen sagen wir, dass die erste aus der zweiten ableitbar ist, wenn wir die zweite durch Anwendung einer der folgenden Regeln auf die erste Gleichung erhalten:

Unabhängige Auswertung:

1. Jeder Ausdruck auf der linken oder rechten Seite einer Gleichung kann durch einen anderen ersetzt werden, der mit ihm identisch ist, d.h. man kann die Ausdrücke links und rechtes unabhängig voneinander vereinfachen.

Komplementbildung:

2: Die rechte und die linke Seite einer Gleichung können gleichzeitig durch ihre Komplemente ersetzt werden.

Erweiterung:

3. Jede Seite einer Gleichung kann mit demselben Ausdruck oder mit einem äquivalenten Ausdruck durch den UND-Operator verknüpft werden.
4. Dual dazu gilt, dass zu jeder Seite äquivalente Ausdrücke durch den ODER-Operator verknüpft werden können.

Ersetzung:

5. Eine Gleichung der Form A + B = 0 ist durch zwei Gleichungen A = 0, B = 0 ersetzbar und umgekehrt.
6. Dual dazu ist eine Gleichung der Form AB = 1 durch zwei Gleichungen A=1, B= 1 ersetzbar.

Schaltfunktionen

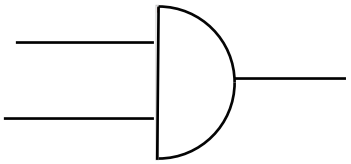
- Funktionen $f: \{0,1\}^n \rightarrow \{0,1\}^m$ mit $n,m \geq 1$ werden auch als Schaltfunktionen bezeichnet
- Eine Schaltfunktion $f: \{0,1\}^n \rightarrow \{0,1\}$ heißt eine n-stellige Boolesche Funktion
- Jede Schaltfunktion $f: \{0,1\}^n \rightarrow \{0,1\}^m$ kann durch m Boolesche Funktionen ausgedrückt werden
- Jede Boolesche Funktion läßt sich eindeutig beschreiben
  - durch eine Wahrheitstabelle (auch Wahrheitstafel genannt)
  - durch einen booleschen Ausdruck (gebildet durch Boolesche Variablen und Operationen aus der Booleschen Algebra)
- Es gibt 2^{2^n} verschiedene n-stellige Boolesche Funktionen (also 16 zweistellige, 256 dreistellige, 65536 vierstellige, ...)

Schaltfunktionen: 2-stellige Fkt. - 1

- zweistellige Boolesche Funktionen $f(x,y)$

Name			Nullfunktion	Konjunktion		
Wahr-heits-tafel	x	y	$f(x,y)$	$f(x,y)$	$f(x,y)$	$f(x,y)$
	0	0	0	0	0	0
	0	1	0	0	0	0
	1	0	0	0	1	1
	1	1	0	1	0	1
Boolescher Ausdruck			$f(x,y) = 0$	$f(x,y) = x \cdot y$	$f(x,y) = x \cdot \overline{y}$	$f(x,y) = x$

- Konjunktion: UND-Funktion

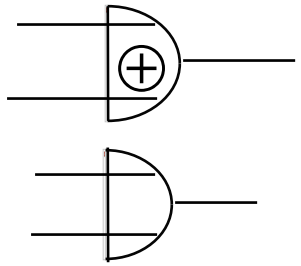


Schaltfunktionen: 2-stellige Fkt. - 2

- zweistellige Boolesche Funktionen $f(x,y)$

Name					exklusives ODER	Disjunktion incl. ODER
Wahr-heits-tafel	x	y	$f(x,y)$	$f(x,y)$	$f(x,y)$	$f(x,y)$
	0	0	0	0	0	0
	0	1	1	1	1	1
	1	0	0	0	1	1
	1	1	0	1	0	1
Boolescher Ausdruck			$f(x,y) = \overline{x} \cdot y$	$f(x,y) = y$	$f(x,y) = \overline{x} \cdot y + x \cdot \overline{y}$	$f(x,y) = x + y$

- exklusives ODER wird auch Antivalenz, XOR oder Addition modulo 2 bezeichnet; andere Notation: $f(x,y) = x \oplus y$
- Disjunktion: ODER-Funktion

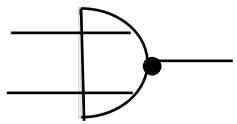


Schaltfunktionen: 2-stellige Fkt. - 3

- zweistellige Boolesche Funktionen *f(x,y)*

Name			negiertes ODER	Äquivalenz		
Wahr-heits-tafel	*x*	*y*	*f* (*x*,*y*)	*f* (*x*,*y*)	*f* (*x*,*y*)	*f* (*x*,*y*)
	0	0	1	1	1	1
	0	1	0	0	0	0
	1	0	0	0	1	1
	1	1	0	1	0	1
Boolescher Ausdruck			$f(x,y) = \overline{x + y}$	$f(x,y) = x \cdot y + \overline{x} \cdot \overline{y}$	$f(x,y) = \overline{y}$	$f(x,y) = x + \overline{y}$

- negiertes ODER wird auch als NOR oder Peirce-Funktion bezeichnet



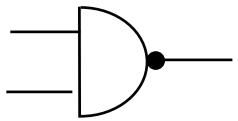
- andere Notation für Äquivalenz: $f(x,y) = x \equiv y$

Schaltfunktionen: 2-stellige Fkt. - 4

- zweistellige Boolesche Funktionen *f(x,y)*

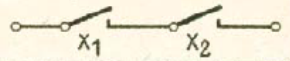
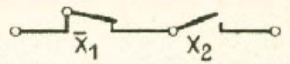
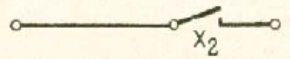
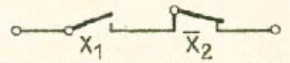
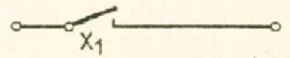
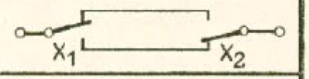
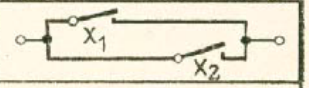
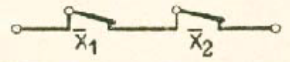
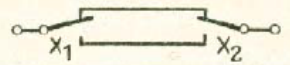
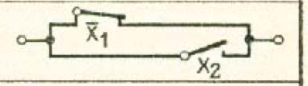
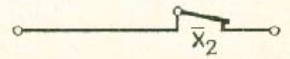
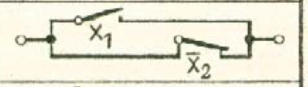
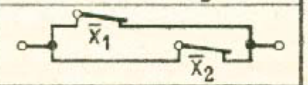
Name				Implikation	negiertes UND	Einsfunktion
Wahr-heits-tafel	*x*	*y*	*f(x,y)*	*f(x,y)*	*f(x,y)*	*f(x,y)*
	0	0	1	1	1	1
	0	1	1	1	1	1
	1	0	0	0	1	1
	1	1	0	1	0	1
Boolescher Ausdruck			$f(x,y) = \overline{x}$	$f(x,y) = \overline{x} + y$	$f(x,y) = \overline{x \cdot y}$	$f(x,y) = 1$

- andere Notation für Implikation: $f(x,y) = x \Rightarrow y$
- negiertes UND wird auch NAND oder Sheffer-Funktion genannt



„Schalt(er)-Algebra

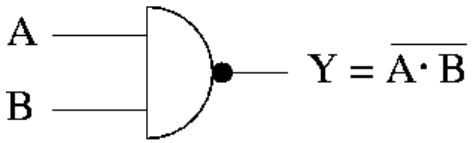
aus: Steinbuch: „Automat
und Mensch“, Springer, Berlin 1963

x_1 / x_2	0 / 0	1 / 0	0 / 1	1 / 1	Bezeichnung	Symbol	Relaisschaltungen
$y_0 = 0$	0	0	0	0		0	
$y_1 = x_1 \bullet x_2$	0	0	0	1	Konjunktion; Und; Sowohl als auch	$\bullet$	
$y_2 = \bar{x}_1 \bullet x_2$	0	0	1	0			
$y_3 = x_2$	0	0	1	1			
$y_4 = x_1 \bullet \bar{x}_2$	0	1	0	0			
$y_5 = x_1$	0	1	0	1			
$y_6 = (x_1 \bullet \bar{x}_2) \vee (\bar{x}_1 \bullet x_2) = x_1 \not\equiv x_2$	0	1	1	0	Antivalenz; exklusives Oder	$\not\equiv$	
$y_7 = x_1 \vee x_2$	0	1	1	1	Disjunktion; inklusives Oder	$\vee$	
$y_8 = \bar{x}_1 \bullet \bar{x}_2 = x_1 \downarrow x_2$	1	0	0	0	weder-noch PEIRCE'scher Pfeil	$\downarrow$	
$y_9 = x_1 \equiv x_2$	1	0	0	1	Äquivalenz	$\equiv$	
$y_{10} = \bar{x}_1$	1	0	1	0			
$y_{11} = \bar{x}_1 \vee x_2$	1	0	1	1	Implikation wenn-dann		
$y_{12} = \bar{x}_2$	1	1	0	0			
$y_{13} = x_1 \vee \bar{x}_2$	1	1	0	1	Implikation wenn-dann		
$y_{14} = \bar{x}_1 \vee \bar{x}_2 = x_1 / x_2$	1	1	1	0	SHEFFER'scher Strich	/	
$y_{15} = 1$	1	1	1	1		1	

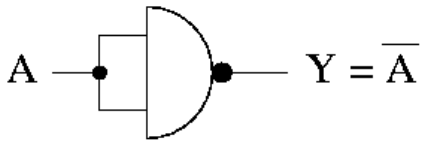
Schaltfunktionen

- mit NAND-Funktion (bzw. NAND-Gatter) können NICHT, UND und ODER einfach nachgebildet werden:

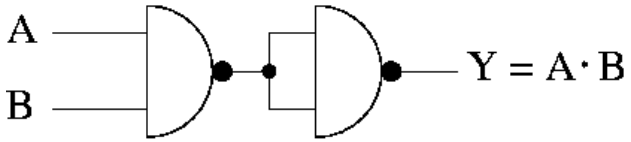
NAND–Gatter:



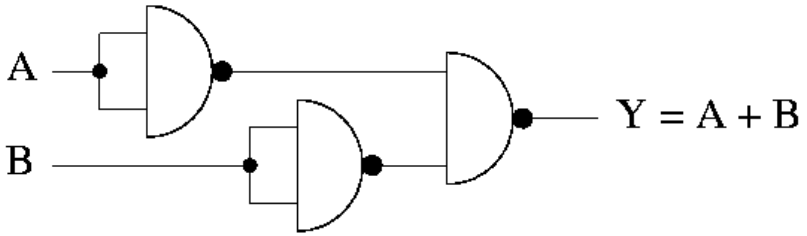
NICHT mit NAND:



UND mit NAND:



ODER mit NAND:



$\Rightarrow$ jede Schaltfunktion ist durch ausschließliche Verwendung von NAND-Gattern realisierbar!

- die gleiche Eigenschaft trifft für NOR-Gatter zu!
- NAND-Gatter waren sehr einfach in Hardware implementierbar

Das Dualitätsprinzip der Booleschen Algebra

- **Dualität:** Für jede aus den Axiomen ableitbare Aussage gibt es eine duale Aussage, die durch Tausch der Operationen + und · sowie durch Tausch der Werte 0 und 1 entsteht.

Sätze und Axiome können paarweise betrachtet werden:

$0 + A = A$	$1 \cdot A = A$	
$1 + A = 1$	$0 \cdot A = 0$	
$A + A = A$	$A \cdot A = A$	
$A + B = B + A$	$A \cdot B = B \cdot A$	Kommutativgesetze
$A + (B + C) = (A + B) + C$	$A \cdot (B \cdot C) = (A \cdot B) \cdot C$	Assoziativgesetze
$(A \cdot B) + (A \cdot C) = A \cdot (B + C)$	$(A+B) \cdot (A+C) = A+(B \cdot C)$	Distributivgesetze
$A + (A \cdot B) = A$	$A \cdot (A + B) = A$	Absorptionsgesetze

Das Dualitätsprinzip der Booleschen Algebra

Definition:

Sei *p* eine aus den Gesetzen der Boolschen Algebra abgeleitete Gleichung.
Die *duale* Gleichung *p'* erhält man aus *p* durch gleichzeitiges Vertauschen der Operatoren + und • sowie 0 und 1.

Satz (Dualitätsprinzip):

Sei *p* eine aus den Gesetzen der Boolschen Algebra abgeleitete Gleichung.
Gilt *p*, dann gilt auch die zu *p duale* Gleichung *p'* .

Schaltfunktionen

Claude Shannon hat im Rahmen seiner Masterarbeit gezeigt, dass man die Boolsche Algebra zur Beschreibung von Schaltkreisen anwenden kann

→ Schaltalgebra

Claude Shannon: „On the Symbolic Analysis of Relay and Switching Circuits“, 1940

Schaltfunktionen

- vier einstellige Boolesche Funktionen:

Name		Nullfunktion	Identität	Negation	Einsfunktion
Wahr-heits-tafel	x	$f(x)$	$f(x)$	$f(x)$	$f(x)$
	0	0	0	1	1
	1	0	1	0	1
Boolescher Ausdruck		$f(x) = 0$	$f(x) = x$	$f(x) = \overline{x}$	$f(x) = 1$

Schaltfunktionen: die 2-stellige Funktionen

a	b	f_0	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_A	f_B	f_C	f_D	f_E	f_F
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

Implikation

f_1: AND

f_6: EXOR

f_7: OR

f_E: NAND

f_8: NOR

Synthese von Schaltungen

- jede Boolesche Funktion kann durch
  - Wahrheitstabelle
  - Boolescher Ausdruck
  - Schaltung mit elementaren Gattern

  dargestellt werden

- eine Darstellung als Boolescher Ausdruck sowie als Schaltung sind jedoch nicht eindeutig!

  ⇒ es werden normierte Ausdrücke (Normalformen) zur Darstellung von Booleschen Funktionen benötigt

- zunächst einige wichtige Begriffe ...

ausgezeichnete Terme

- **Produktterm:**
  - einfache Variable (ggf. negiert)
  - Konjunktion mehrerer Variablen (ggf. negiert)
  - Beispiele: x, y, $\overline{x} \cdot y$, $x \cdot \overline{y} \cdot z$
- **Summenterm:**
  - wie Produktterm, jedoch Disjunktion statt Konjunktion
  - Beispiele: x, $\overline{y}$, $x + \overline{y}$, $x + \overline{y} + z$
- **Minterm:**
  - Produktterm, in dem jede Variable einer booleschen Funktion genau einmal vorkommt (einfach oder negiert)
  - Beispiel: $x \cdot \overline{y} \cdot z$ ist ein Minterm der Funktion $f(x,y,z)$
- **Maxterm:**
  - Summenterm, in dem jede Variable einer booleschen Funktion genau einmal vorkommt (einfach oder negiert)
  - Beispiel: $\overline{x} + y + z$ ist ein Maxterm der Funktion $f(x,y,z)$

Ableitung eines Terms (Summe von Produkten)

Die Spezifikation:

Eingaben		Ausgabe
X	Y	A
0	0	1
0	1	0
1	0	1
1	1	1

Ein- Ausgabeverhalten einer Schaltung

Erweiterung der Tabelle für Minterme:

Eingaben		Ausgabe	Minterme
X	Y	A	
0	0	1	$\overline{X}\,\overline{Y}$
0	1	0	$\overline{X}\,Y$
1	0	1	$X\overline{Y}$
1	1	1	XY

B.-Gleichung $\overline{X}\overline{Y} + X\overline{Y} + XY = A$

Ableitung:

$\overline{X}\,\overline{Y} + X\overline{Y} + XY = A$

$\overline{X}\overline{Y} + X(\overline{Y} + Y) = A$

$\overline{X}\overline{Y} + X = A$

$X + \overline{Y} = A$

Beweis der Äquivalenz

Wahrheitstafel:

X	Y	$\overline{Y}$	$X + \overline{Y}$
0	0	1	1
0	1	0	0
1	0	1	1
1	1	0	1

$X + \overline{Y}$

Gesetze der Booleschen Algebra

Zusammengefasst:

$0 + A = A$	$1 \bullet A = A$	
$1 + A = 1$	$0 \bullet A = 0$	
$A + A = A$	$A \bullet A = A$	
$A + \overline{A} = 1$	$A \bullet \overline{A} = 0$	
$A = \overline{\overline{A}}$		
$A + B = B + A$	$A \bullet B = B \bullet A$	Kommutativges.
$A + (B + C) = (A + B) + C$	$A \bullet (B \bullet C) = (A \bullet B) \bullet C$	Ass. Ges.
$A\bullet (B + C) = A \bullet B + A \bullet C$	$A+(B \bullet C) = (A+B) \bullet (A+C)$	Distr. Ges.
$A + (A \bullet B) = A$	$A \bullet (A + B) = A$	Absorptionsges.
$A + (A \bullet B) = A + B$		

Ableitung eines Terms (Summe von Produkten)

Wir haben:

1. ..aus dem Ein-Ausgabeverhalten einer Schaltung eine Spezifikation in Form einer Wahrheitstafel erstellt.

2. ..aus der Wahrheitstafel über die Minterme eine Boolsche Gleichung abgeleitet.

3. ..vereinfacht nach den Regeln der Boolschen Algebra

4. ..verifiziert durch Wahrheitstabelle

Ableitung eines Terms (Summe von Produkten)

Spezifikation mit 3 Eingaben:

Eingaben			Ausgabe
X	Y	Z	A
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

Erweiterung der Tabelle für die Minterme:

Eingaben			Ausgabe	Minterme
X	Y	Z	A	
0	0	0	0	$\overline{X}\,\overline{Y}\,\overline{Z}$
0	0	1	1	$\overline{X}\,\overline{Y}\,Z$
0	1	0	0	$\overline{X}\,Y\,\overline{Z}$
0	1	1	1	$\overline{X}\,Y\,Z$
1	0	0	0	$X\,\overline{Y}\,\overline{Z}$
1	0	1	1	$X\overline{Y}Z$
1	1	0	0	$X\,Y\,\overline{Z}$
1	1	1	1	XYZ

$$\overline{X}\,\overline{Y}Z + \overline{X}YZ + X\overline{Y}Z + XYZ = A$$

Gesetze der Booleschen Algebra

Zusammengefasst:

$0 + A = A$	$1 \bullet A = A$	
$1 + A = 1$	$0 \bullet A = 0$	
$A + A = A$	$A \bullet A = A$	
$A + \overline{A} = 1$	$A \bullet \overline{A} = 0$	
$A = \overline{\overline{A}}$		
$A + B = B + A$	$A \bullet B = B \bullet A$	Kommutativges.
$A + (B + C) = (A + B) + C$	$A \bullet (B \bullet C) = (A \bullet B) \bullet C$	Ass. Ges.
$A\bullet (B + C) = A \bullet B + A \bullet C$	$A+(B \bullet C) = (A+B) \bullet (A+C)$	Distr. Ges.
$A + (A \bullet B) = A$	$A \bullet (A + B) = A$	Absorptionsges.
$A + (A \bullet B) = A + B$		

Ableitung eines Terms (Summe von Produkten)

Spezifikation mit 3 Eingaben:

Eingaben			Ausgabe
X	Y	Z	A
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

Vereinfachen:

$\overline{X}\overline{Y}Z + \overline{X}YZ + X\overline{Y}Z + XYZ = A$

$\overline{X}Z(\overline{Y} + Y) + XZ(\overline{Y} + Y) = A$

$\overline{X}Z + XZ = A$

$Z(\overline{X} + X) = A$

$Z = A$

Erweiterung der Tabelle für die Minterme:

Eingaben			Ausgabe	Minterme
X	Y	Z	A	
0	0	0 ⟷	0	$\overline{X}\,\overline{Y}\,\overline{Z}$
0	0	1 ⟷	1	$\overline{X}\,\overline{Y}\,Z$
0	1	0 ⟷	0	$\overline{X}\,Y\,\overline{Z}$
0	1	1 ⟷	1	$\overline{X}\,Y\,Z$
1	0	0 ⟷	0	$X\,\overline{Y}\,\overline{Z}$
1	0	1 ⟷	1	$X\overline{Y}Z$
1	1	0 ⟷	0	$X\,Y\,\overline{Z}$
1	1	1 ⟷	1	XYZ

$\overline{X}\overline{Y}Z + \overline{X}YZ + X\overline{Y}Z + XYZ = A$

Ableitung eines Terms (Produkt von Summen)

Dual zur „Summe von Produkten“

Eingaben		Ausgabe
X	Y	A
0	0	1
0	1	0
1	0	1
1	1	1

Eingaben		Ausgabe	Maxterme
X	Y	A	
0	0	1	$X + Y$
0	1	0	$X + \overline{Y}$
1	0	1	$\overline{X} + Y$
1	1	1	$\overline{X} + \overline{Y}$

Abgeleitete Gleichung: $A = X + \overline{Y}$

Entsprechend der Dualität werden die Terme ausgewählt, die „0“ ergeben.

Ableitung eines Terms (Produkt von Summen)

Dual zur „Summe von Produkten“

Eingaben Ausgabe

X	Y	A
0	0	1
0	1	0
1	0	1
1	1	1

Eingaben		Ausgabe	Maxterme
X	Y	A	
0	0	1	$X + Y$
0	1	0	$X + \overline{Y}$
1	0	1	$\overline{X} + Y$
1	1	1	$\overline{X} + \overline{Y}$

Abgeleitete Gleichung: $A = X + \overline{Y}$

Vergleich mit Minterm-Darstellung

$$
\begin{aligned}
A &= \bar{Y}\bar{X} + \bar{Y}X + XY \\
&= (\bar{X}+X)\bar{Y} + XY \\
&= \bar{Y} + XY \\
&= \bar{Y} + X
\end{aligned}
$$

Ausgezeichnete Terme

- **Disjunktive Normalform (DNF, Summe von Produkten):**
  - Disjunktion von Produktttermen
  - Beispiel: $(x \cdot \overline{y}) + (x \cdot \overline{y} \cdot z)$
- **Konjunktive Normalform (KNF, Produkt von Summen):**
  - Konjunktion von Summentermen
  - Beispiel: $w \cdot (\overline{x} + y) \cdot (x + \overline{y} + z)$
- **Kanonische Disjunktive Normalform (KDNF):**
  - eindeutige Darstellung einer booleschen Funktion f als Disjunktion von Mintermen
  - Beispiel: $(\overline{x} \cdot \overline{y} \cdot \overline{z}) + (x \cdot \overline{y} \cdot z) + (x \cdot y \cdot \overline{z})$ ist KDNF von $f(x,y,z)$
- **Kanonische Konjunktive Normalform (KKNF):**
  - eindeutige Darstellung einer booleschen Funktion f als Konjunktion von Maxtermen
  - Beispiel: $(\overline{x} + \overline{y}) \cdot (\overline{x} + y) \cdot (x + \overline{y})$ ist KKNF von $f(x,y)$

Sätze der Schaltalgebra

- Satz: Jede Boolsche Funktion läßt sich als genau eine KDNF darstellen
- Bildung der KDNF für *n* -stellige Boolsche Funktion *f* :
  - für jede Zeile der Wahrheitstabelle mit $f(x_1, x_2, \ldots, x_n) = 1$ wird ein Minterm aufgestellt
  - hierin wird jede Variable x_i negiert, wenn in der entsprechenden Zeile der Wert der Variablen 0 ist
  - Beispiel:

x_1	x_2	x_3	x_4	$f(x_1,x_2,x_3,x_4)$
			...	
0	1	1	0	1
			...	

$\rightarrow \overline{x_1} \cdot x_2 \cdot x_3 \cdot \overline{x_4}$

- Darstellung als KDNF ist abgesehen von Reihenfolge eindeutig!

Sätze der Schaltalgebra

- Satz: Jede Boolosche Funktion läßt sich als genau eine KKNF darstellen
- Bildung der KKNF für *n* -stellige Boolsche Funktion *f* :
  - für jede Zeile der Wahrheitstabelle mit $f(x_1, x_2, \ldots, x_n) = 0$ wird ein Maxterm aufgestellt
  - hierin wird jede Variable x_i negiert, wenn in der entsprechenden Zeile der Wert der Variablen 1 ist
  - Beispiel:

x_1	x_2	x_3	x_4	$f(x_1,x_2,x_3,x_4)$
			...	
1	0	1	0	0
			...	

$\rightarrow \quad \overline{x_1} + x_2 + \overline{x_3} + x_4$

- Darstellung als KKNF ist abgesehen von Reihenfolge eindeutig!

Sätze der Schaltalgebra

- Satz: Jede KDNF kann in eine KKNF umgewandelt, jede KKNF kann in eine KDNF umgewandelt werden
- Zwei Darstellungen Boolescher Funktionen sind äquivalent, wenn sie durch Umformungen gemäß den Sätzen der Booleschen Algebra auf die gleiche KDNF bzw. KKNF zurückgeführt werden können

Synthese

- Eine Schaltung zur Realisierung einer Booleschen Funktion *f* kann erzeugt werden durch:
  1) Aufstellen der Wahrheitstafel von *f*
  2) Bilden der KDNF (oder KKNF) von *f*
  3) Realisierung der KDNF (oder KKNF) mit Gattern

KDNF oder KKNF?

Synthese

- eine KDNF ist günstiger als eine KKNF, wenn nur für wenige Kombinationen der Eingabewerte $f(x_1, x_2, ..., x_n) = 1$ gilt.
  Umgekehrt, wenn nur für wenige Kombinationen der Eingabewerte $f(x_1, x_2, ..., x_n) = 0$ gilt, ist die KKNF vorzuziehen.

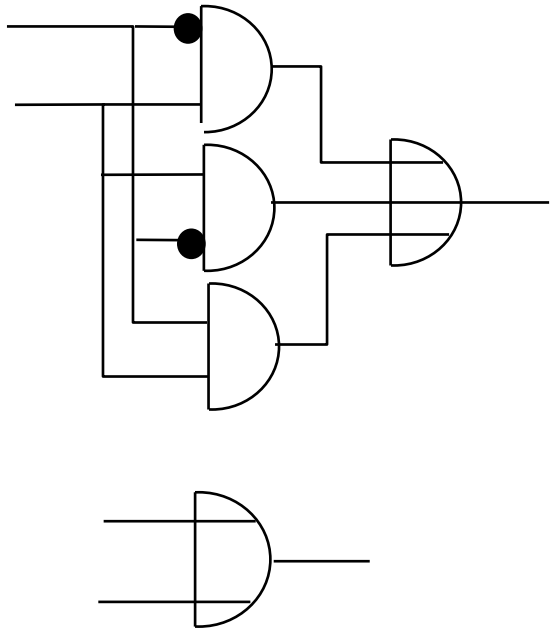
- Beispiel : ODER-Funktion

  Wahrheitstafel

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

KDNF: $\bar{x}\cdot y + x\cdot\bar{y} + x\cdot y$

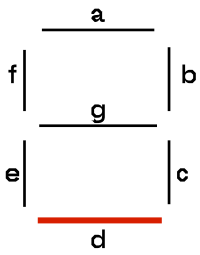
KKNF: $x + y$

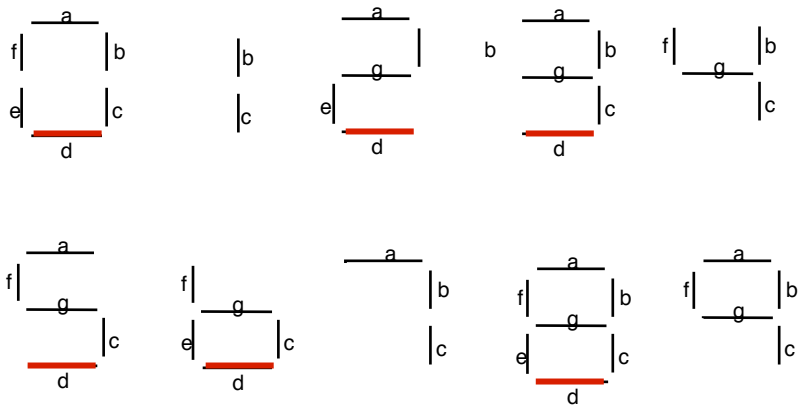


Synthese mit Don‘t Care Einträge

Beispiel :

Siebensegment-Anzeige:





Die Belegungen die auf Don't Care abgebildet werden, müssen nicht in der KDNF oder KKNF berücksichtigt werden!

Wahrheitstafel für Segment d

	Eingaben W	X	Y	Z	Ausgabe d
0	0	0	0	0	1
1	0	0	0	1	0
2	0	0	1	0	1
3	0	0	1	1	1
4	0	1	0	0	0
5	0	1	0	1	1
6	0	1	1	0	1
7	0	1	1	1	0
8	1	0	0	0	1
9	1	0	0	1	0
10	1	0	1	0	D
11	1	0	1	1	D
12	1	1	0	0	D
13	1	1	0	1	D
14	1	1	1	0	D
15	1	1	1	1	D

D: Don‘t care

Minimierung von Schaltfunktionen

- kanonische Normalformen stellen zwar eine eindeutige, jedoch nicht minimale Darstellung einer Schaltfunktion dar
- Wann heißt eine Darstellung minimal ?
  - kleinste Anzahl an Gattern ?
  - kleinste Anzahl an Verbindungsleitungen ?
  - kleinste Anzahl an Produkttermen bzw. Summentermen ?

Unser Ziel: **kleinste Anzahl Boolescher Operationen !**

- Lösungsmöglichkeiten:
  1) geeignetes Umformen mit Gesetzen der Booleschen Algebra
  2) Einsatz eines graphischen Verfahrens (z.B. ein Karnaugh-Veith-Diagramm), nur möglich bei Schaltfunktionen mit wenigen Variablen
  3) algorithmisches Minimieren (z.B. Verfahren nach Quine-Mc-Cluskey), geeignet auch für Schaltfunktionen mit vielen Variablen

Minimierung Boolscher Ausdrücke

Verfahren: Anwendung der Gesetze der Boolschen Algebra

Beispiel:

$$f(x_1, x_2, x_3) = x_1 \cdot x_2 \cdot \overline{x_3} + x_1 \cdot x_2 \cdot x_3 + x_1 \cdot \overline{x_2} \cdot x_3 \quad \text{(Idempotenzgesetz)}$$

$$= x_1 \cdot x_2 \cdot \overline{x_3} + x_1 \cdot x_2 \cdot x_3 + x_1 \cdot x_2 \cdot x_3 + x_1 \cdot \overline{x_2} \cdot x_3 \quad \text{(Distributivgesetz)}$$

$$= x_1 \cdot x_2 \cdot (\overline{x_3} + x_3) + x_1 \cdot x_2 \cdot x_3 + x_1 \cdot \overline{x_2} \cdot x_3 \quad \text{(Kommutativgesetz)}$$

$$= x_1 \cdot x_2 \cdot (x_3 + \overline{x_3}) + x_1 \cdot x_3 \cdot x_2 + x_1 \cdot x_3 \cdot \overline{x_2} \quad \text{(Distributivgesetz)}$$

$$= x_1 \cdot x_2 \cdot (x_3 + \overline{x_3}) + x_1 \cdot x_3 \cdot (x_2 + \overline{x_2}) \quad \text{(Komplementäres Element)}$$

$$= x_1 \cdot x_2 \cdot 1 + x_1 \cdot x_3 \cdot 1 \quad \text{(Neutrales Element)}$$

$$= x_1 \cdot x_2 + x_1 \cdot x_3$$

Minimierung Boolscher Ausdrücke

Idee: Anwendung der Resolutionregel: $x \cdot y + x \cdot \overline{y} = x$

Wenn sich in einer disjunktiven Normalform zwei Summanden nur in genau einer komplementären Variablen unterscheiden, so können beide Summanden durch ihren gemeinsamen Teil ersetzt werden

Karnaugh-Veitch-Diagramme

Ausgangspunkt: $x \cdot y + x \cdot \bar{y} = x$

jede zweistellige Boolsche Fkt.
lässt sich darstellen als:

$x \cdot y + \bar{x} \cdot y + x \cdot \bar{y} + \bar{x} \cdot \bar{y}$

Terme in Spalten unterscheiden sich jeweils genau um das Vorzeichen einer Komponente

Terme in Zeilen unterscheiden sich jeweils um das „Vorzeichen“ einer Komponente.

grafische Darstellung:

	$\bar{x}$	x
$\bar{y}$	$\bar{x} \cdot \bar{y}$	$x \cdot \bar{y}$
y	$\bar{x} \cdot y$	$x \cdot y$

		x = 0	x = 1
		$\bar{x}$	x
y = 0	$\bar{y}$	0	1
y = 1	y	0	1

jedes Element repräsentiert genau einen Minterm

Karnaugh-Veitch-Diagramme

Ausgangspunkt: $x \cdot y + x \cdot \bar{y} = x$

Beispiel: Exclusives ODER:

$$x \cdot \bar{y} + \bar{x} \cdot y = F$$

x	y	F	Minterm
0	0	0	$\bar{x} \cdot \bar{y}$
0	1	1	$\bar{x} \cdot y$
1	0	1	$x \cdot \bar{y}$
1	1	0	$x \cdot y$

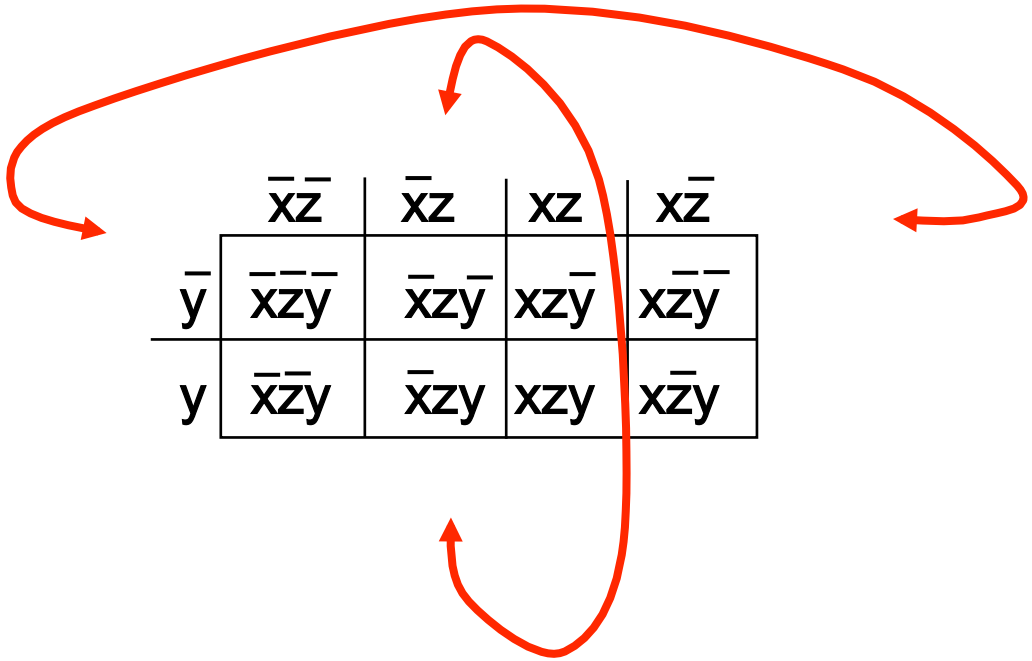
grafische Darstellung:

	$\bar{x}$	x
$\bar{y}$	$\bar{x} \cdot \bar{y}$	$x \cdot \bar{y}$
y	$\bar{x} \cdot y$	$x \cdot y$

		x = 0	x = 1
		$\bar{x}$	x
y = 0	$\bar{y}$	0	1
y = 1	y	1	0

jedes Element repräsentiert genau einen Minterm

Karnaugh-Veitch-Diagramme für 3 Variablen



Karnaugh-Veitch_Diagramme werden geometrisch als Torus interpretiert !

Karnaugh-Veitch-Diagramme

Alternative Notationsformen:

	$\bar{x}\bar{z}$	$\bar{x}z$	xz	$x\bar{z}$
$\bar{y}$	$\bar{x}\bar{z}\bar{y}$	$\bar{x}z\bar{y}$	$xz\bar{y}$	$x\bar{z}\bar{y}$
y	$\bar{x}\bar{z}y$	$\bar{x}zy$	xzy	$x\bar{z}y$

y \ xz	00	01	11	01
0	$\bar{x}\bar{z}\bar{y}$	$\bar{x}z\bar{y}$	$xz\bar{y}$	$x\bar{z}\bar{y}$
1	$\bar{x}\bar{z}y$	$\bar{x}zy$	xzy	$x\bar{z}y$

	$\bar{x}$		x	
$\bar{y}$	$\bar{x}\bar{z}\bar{y}$	$\bar{x}z\bar{y}$	$xz\bar{y}$	$x\bar{z}\bar{y}$
y	$\bar{x}\bar{z}y$	$\bar{x}zy$	xzy	$x\bar{z}y$
	$\bar{z}$	z		$\bar{z}$

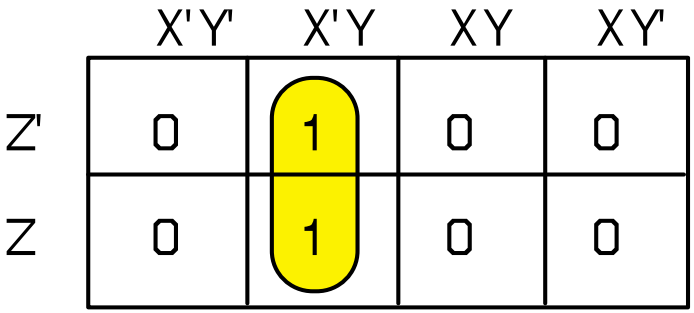
Karnaugh-Veitch-Diagramme

- zur Minimierung der KDNF einer n-stelligen Funktion f

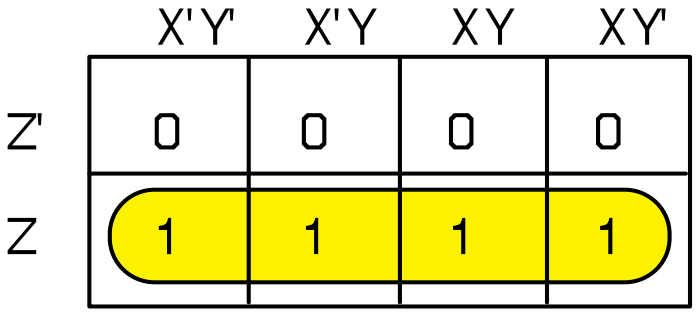
1. wird für jeden Minterm mit $f(x_1,\ldots,x_n) = 1$ eine 1 im Diagramm eingetragen;
2. wird für jeden Minterm mit $f(x_1,\ldots,x_n) = 0$ eine 0 im Diagramm eingetragen;
3. werden möglichst wenige und möglichst große zusammenhängende Bereiche (Schleifen) aus Einsen markiert, wobei
   1) jeder Bereich aus 2^k Elementen (mit $0 \leq k \leq n$) besteht;
   2) alle Einsen überdeckt werden müssen;
4. die markierten Bereiche nach der Resolutionsregel zu Produkttermen zusammengefasst werden, die summiert werden.

Karnaugh-Veitch-Diagramme

Einfache Beispiele:



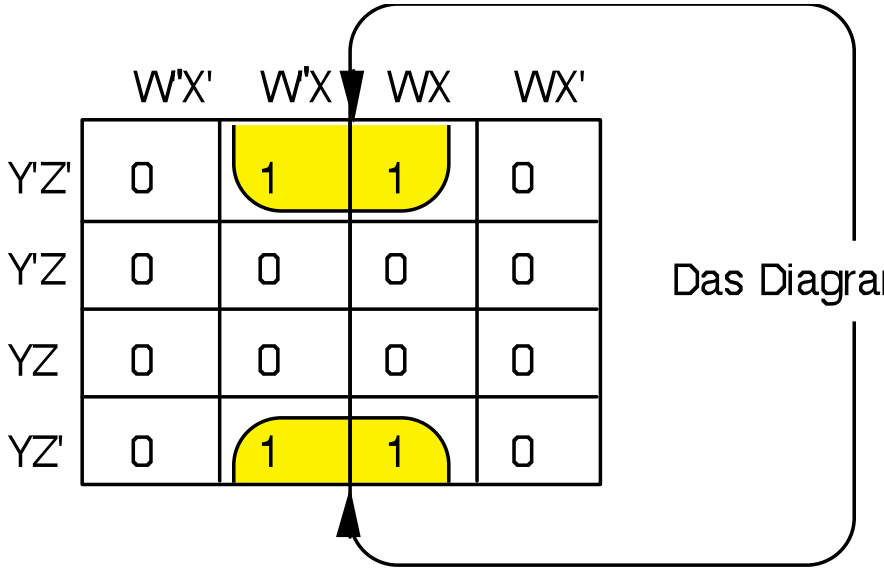
x‘yz‘ + x‘yz = (x‘y)z‘ + (x‘y) z
= x‘y



x‘y‘z + x‘yz + xyz + xy‘z
= (x‘z)y‘ + (x‘z) y + (xz)y + (xz) y‘
= x‘z + xz
= z

Karnaugh-Veitch-Diagramme mit 4 Variablen

Einfache Beispiele:



Das Diagramm ist gerollt

w‘xy‘z‘ + wxy‘z‘ + w‘xyz‘ + wxyz‘
= w‘(xy‘z‘) + w(xy‘z‘) + w‘(xyz‘) + w(xyz‘)
= xy‘z‘ + xyz‘
= xz‘

Karnaugh-Veitch-Diagramme

Wahrheitstabelle:

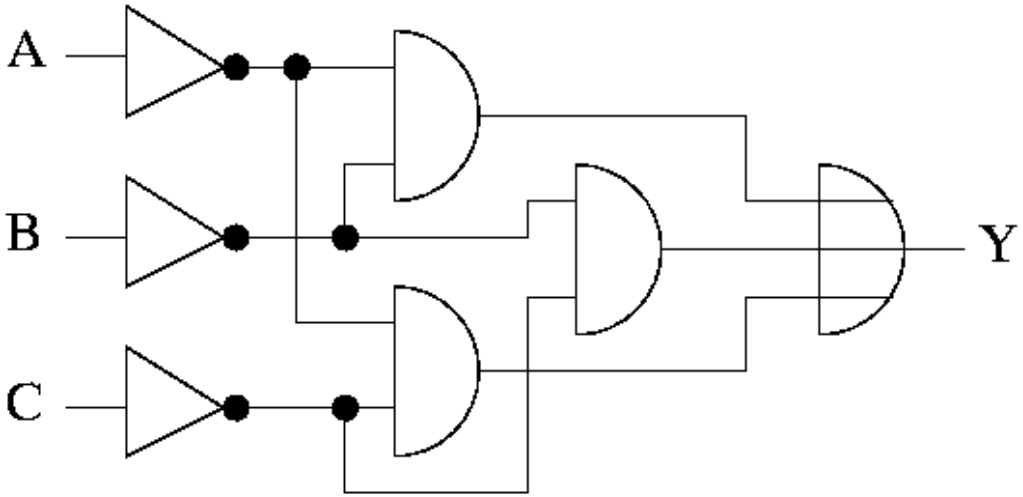
A	B	C	Y
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0

Karnaugh-Veitch-Diagramme

Wahrheitstabelle:

A	B	C	Y
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0

Realisierung:

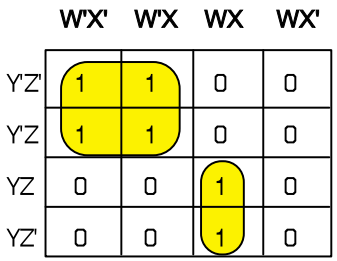


Y = A'C' + C'B' + A'B'

Karnaugh-Veitch-Diagramme

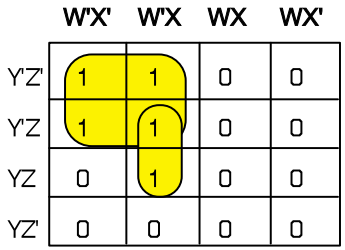
Nach welchen Kriterien werden Schleifen gezogen ?

Disjunkt:

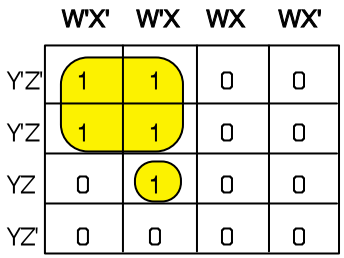


W‘Y‘ + WXY

Überlappend:



(W‘Y‘) + (W‘XY)

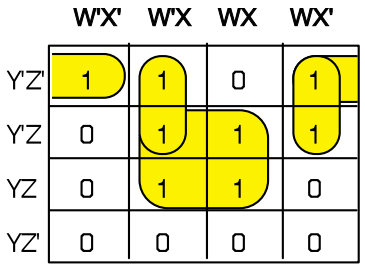


(W‘Y‘) + (W‘XYZ)

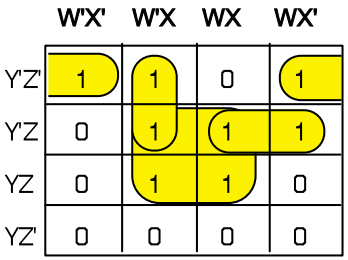
immer die größtmöglichen Schleifen wählen !

Karnaugh-Veitch-Diagramme

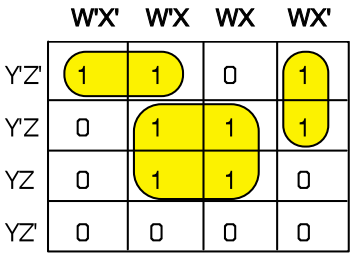
	W'X'	W'X	WX	WX'
Y'Z'	1	1	0	1
Y'Z	0	1	1	1
YZ	0	1	1	0
YZ'	0	0	0	0



	W'X'	W'X	WX	WX'
Y'Z'	1	1	0	1
Y'Z	0	1	1	1
YZ	0	1	1	0
YZ'	0	0	0	0



	W'X'	W'X	WX	WX'
Y'Z'	1	1	0	1
Y'Z	0	1	1	1
YZ	0	1	1	0
YZ'	0	0	0	0



	W'X'	W'X	WX	WX'
Y'Z'	1	1	0	1
Y'Z	0	1	1	1
YZ	0	1	1	0
YZ'	0	0	0	0

Lösung 1:
x‘y‘z‘ + w‘xy‘ + xz + wx‘y‘

Lösung 2:
x‘y‘z‘ + w‘xy‘ + xz + wy‘z

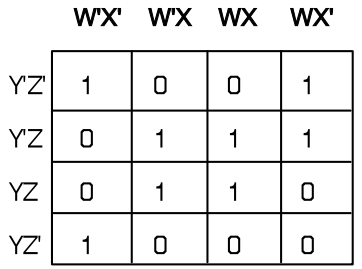
Lösung 3:
w‘y‘z‘ + xz + wx‘y‘

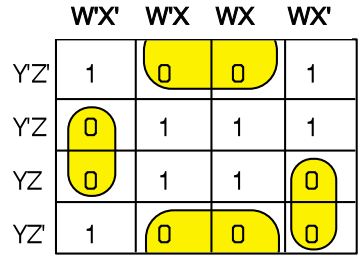
Überlappung lohnt sich nur, wenn dadurch größere Schleifen entstehen!
z.B. eine isolierte „1“ vermieden wird.

Karnaugh-Veitch-Diagramme

Karnaugh-Veitch Diagramme von „Produkten von Summen“:

Eingaben				Ausgabe
W	X	Y	Z	A
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	0
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	1
1	1	1	0	0
1	1	1	1	1





Ergebnis: XZ' + W'X'Z + WX'Y = A'

Komplement: (X'+Z)(W+X+Z')(W'+X+Y') = A

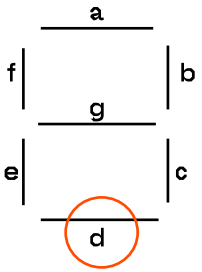
Karnaugh-Veitch-Diagramme

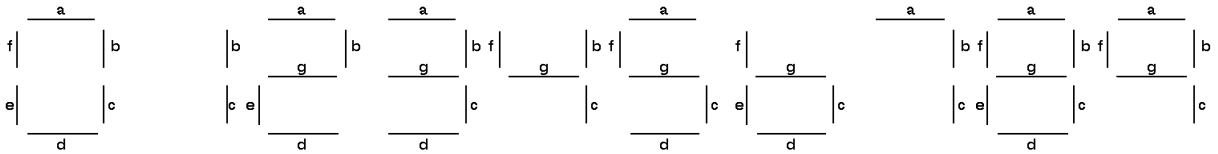
Überprüfung:

$$(X'+Z)(W+X+Z')(W'+X+Y') = A$$

Eingaben				invertiert				Terme des Produkts der Summen			Ausgabe
W	X	Y	Z	W'	X'	Y'	Z'	X'+Z	W+X+Z'	W'+X+Y'	d
0	0	0	0	1	1	1	1	1	1	1	1
0	0	0	1	1	1	1	0	1	0	1	0
0	0	1	0	1	1	0	1	1	1	1	1
0	0	1	1	1	1	0	0	1	0	1	0
0	1	0	0	1	0	1	1	0	1	1	0
0	1	0	1	1	0	1	0	1	1	1	1
0	1	1	0	1	0	0	1	0	1	1	0
0	1	1	1	1	0	0	0	1	1	1	1
1	0	0	0	0	1	1	1	1	1	1	1
1	0	0	1	0	1	1	0	1	1	1	1
1	0	1	0	0	1	0	1	1	1	0	0
1	0	1	1	0	1	0	0	1	1	0	0
1	1	0	0	0	0	1	1	0	1	1	0
1	1	0	1	0	0	1	0	1	1	1	1
1	1	1	0	0	0	0	1	0	1	1	0
1	1	1	1	0	0	0	0	1	1	1	1

Karnaugh-Veitch-Diagramme





Eingaben				Ausgabe
W	X	Y	Z	d
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	0
1	0	0	0	1
1	0	0	1	0
1	0	1	0	D
1	0	1	1	D
1	1	0	0	D
1	1	0	1	D
1	1	1	0	D
1	1	1	1	D

	W'X'	W'X	WX	WX'
Y'Z'	1	0	D	1
Y'Z	0	1	D	0
YZ	1	0	D	D
YZ'	1	1	D	D

	W'X'	W'X	WX	WX'
Y'Z'	1	0	D	1
Y'Z	0	1	D	0
YZ	1	0	D	D
YZ'	1	1	D	D

$$d = X'Z' + X'Y + YZ' + XY'Z$$

Karnaugh-Veitch-Diagramme

Regeln zur Bildung der Schleifen:

- Fange mit isolierten Zellen an. Die entsprechenden Minterme können nicht mehr vereinfacht werden.
- Falls keine isolierten Zellen existieren, fange bei denen mit den wenigsten gleichwertigen Nachbarzellen an.
- Suche die Schleifen mit der größten Überdeckung von Zellen. Die Schleifen umfassen jeweils 2^n (n= 0,1,2,...) benachbarte Zellen. Fange mit den kleinsten Schleifen an.
- Alle Minterme mit dem Wert "1" müssen abgedeckt sein.
- Überlappungen führen nur dann zu minimaleren Ausdrücken, wenn dadurch größere Schleifen gebildet werden können.
- Die minimale Funktion besteht aus der kleinsten Schleifenmenge, die alle individuell möglichst groß sind.

Karnaugh-Veitch-Diagramme

mit mehr als 4 Variablen

?

Karnaught-Veitch Diagramme mit 5 Variablen

$\bar{E}$

	$\bar{A}\bar{B}$	$\bar{A}B$	AB	$A\bar{B}$
$\bar{C}\bar{D}$	0	0	0	0
$\bar{C}D$	0	1	0	0
CD	0	0	0	0
$C\bar{D}$	0	0	0	0

E

	$\bar{A}\bar{B}$	$\bar{A}B$	AB	$A\bar{B}$
$\bar{C}\bar{D}$	0	0	0	0
$\bar{C}D$	0	1	0	0
CD	0	0	0	1
$C\bar{D}$	0	0	0	1

$$\bar{A}B\bar{C}D\bar{E} + \bar{A}B\bar{C}DE + A\bar{B}CDE + A\bar{B}C\bar{D}E$$

Karnaught-Veitch Diagramme mit 6 Variablen

$$\bar{A}\bar{B}\bar{C}\bar{D}\bar{E}\bar{F} + \bar{A}\bar{B}\bar{C}\bar{D}\bar{E}F + \bar{A}\bar{B}\bar{C}\bar{D}E\bar{F} + \bar{A}\bar{B}\bar{C}DE\bar{F}$$

$$+ \bar{A}\bar{B}\bar{C}DEF + \bar{A}\bar{B}C\bar{D}\bar{E}\bar{F} + \bar{A}\bar{B}C\bar{D}\bar{E}F + \bar{A}\bar{B}C\bar{D}E\bar{F}$$

$$+ \bar{A}\bar{B}CDEF + \bar{A}BC\bar{D}\bar{E}\bar{F} + \bar{A}BC\bar{D}\bar{E}F + \bar{A}BC\bar{D}E\bar{F}$$

$$+ \bar{A}BCDEF + A\bar{B}C\bar{D}\bar{E}\bar{F} + A\bar{B}C\bar{D}\bar{E}F + A\bar{B}C\bar{D}E\bar{F}$$

$$+ A\bar{B}CDEF + ABC\bar{D}\bar{E}\bar{F} + ABC\bar{D}\bar{E}F + ABC\bar{D}E\bar{F}$$

$$+ ABCDEF$$

Karnaught-Veitch Diagramme mit 6 Variablen

$$\overline{A}\overline{B}\overline{C}\overline{D}\overline{E}\overline{F} + \overline{A}\overline{B}\overline{C}\overline{D}\overline{E}F + \overline{A}\overline{B}\overline{C}\overline{D}EF + \overline{A}\overline{B}\overline{C}DEF$$

$$+ \overline{A}\overline{B}\overline{C}\overline{D}EF + \overline{A}\overline{B}\overline{C}\overline{D}\overline{E}F + \overline{A}\overline{B}\overline{C}\overline{D}\overline{E}\overline{F} + \overline{A}\overline{B}\overline{C}\overline{D}\overline{E}\overline{F}$$

$$+ \overline{A}\overline{B}\overline{C}DEF + \overline{A}\overline{B}\overline{C}D\overline{E}F + \overline{A}\overline{B}\overline{C}D\overline{E}\overline{F} + \overline{A}\overline{B}\overline{C}\overline{D}\overline{E}\overline{F}$$

$$+ \overline{A}\overline{B}CDEF + \overline{A}\overline{B}CD\overline{E}F + \overline{A}\overline{B}CD\overline{E}\overline{F} + \overline{A}\overline{B}C\overline{D}\overline{E}\overline{F}$$

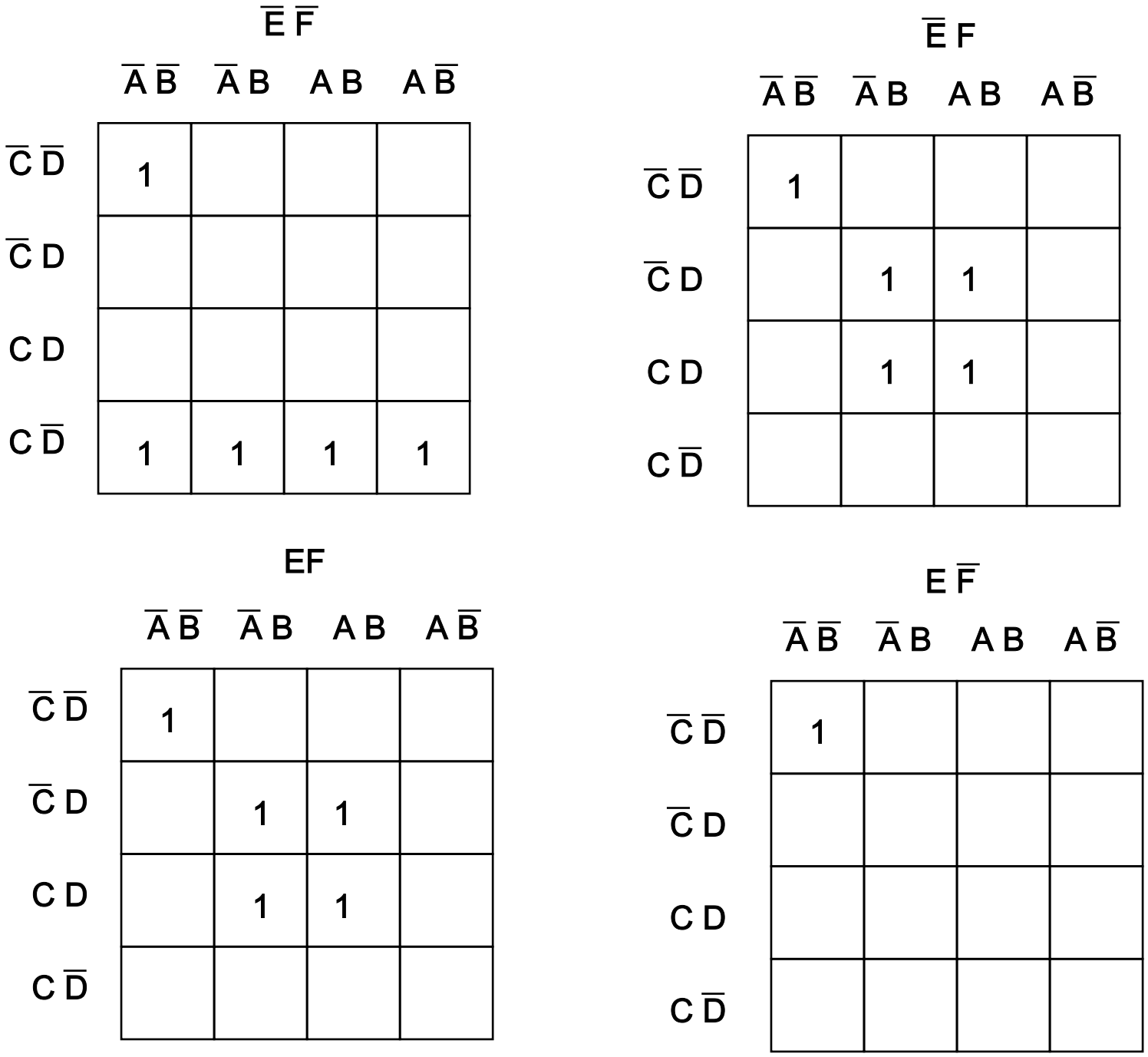
$$+ \overline{A}BCDEF + \overline{A}BCD\overline{E}F + \overline{A}BCD\overline{E}\overline{F} + \overline{A}BC\overline{D}\overline{E}\overline{F}$$

$$+ A\overline{B}CDEF + A\overline{B}CD\overline{E}F + A\overline{B}CD\overline{E}\overline{F} + A\overline{B}C\overline{D}\overline{E}\overline{F}$$

$$+ AB\overline{C}DEF + AB\overline{C}D\overline{E}F + AB\overline{C}D\overline{E}\overline{F} + AB\overline{C}\overline{D}\overline{E}\overline{F}$$

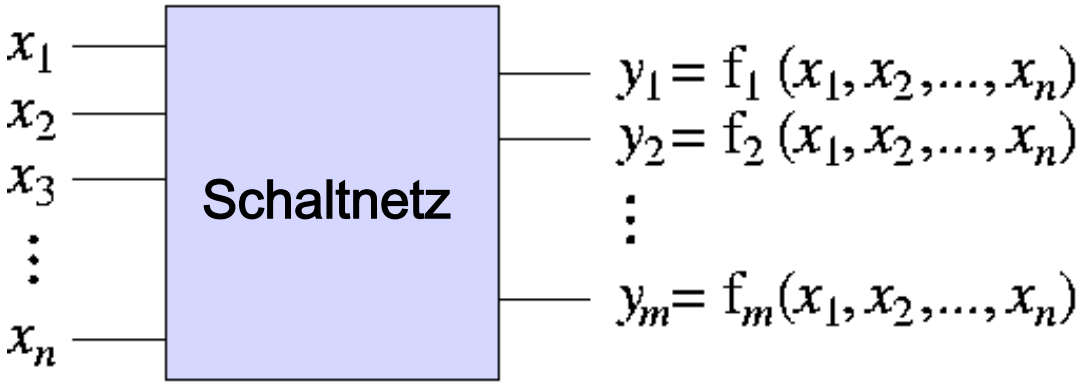
$$+ ABCDEF + ABCD\overline{E}F + ABCD\overline{E}\overline{F} + ABC\overline{D}\overline{E}\overline{F}$$

$$\overline{A}\overline{B}\overline{C}\overline{D} + \overline{C}\overline{D}\overline{E}\overline{F} + BDF$$


$\overline{E}\,\overline{F}$
$\overline{A}\,\overline{B}$ $\overline{A}\,B$ $A\,B$ $A\,\overline{B}$
$\overline{C}\,\overline{D}$ 1
$\overline{C}\,D$
$C\,D$
$C\,\overline{D}$ 1 1 1 1
$\overline{E}\,F$
$\overline{A}\,\overline{B}$ $\overline{A}\,B$ $A\,B$ $A\,\overline{B}$
$\overline{C}\,\overline{D}$ 1
$\overline{C}\,D$ 1 1
$C\,D$ 1 1
$C\,\overline{D}$
EF
$\overline{A}\,\overline{B}$ $\overline{A}\,B$ $A\,B$ $A\,\overline{B}$
$\overline{C}\,\overline{D}$ 1
$\overline{C}\,D$ 1 1
$C\,D$ 1 1
$C\,\overline{D}$
$E\,\overline{F}$
$\overline{A}\,\overline{B}$ $\overline{A}\,B$ $A\,B$ $A\,\overline{B}$
$\overline{C}\,\overline{D}$ 1
$\overline{C}\,D$
$C\,D$
$C\,\overline{D}$

Schaltnetze

- Ein Schaltnetz ist eine schaltungstechnische Realisierung einer Schaltfunktion
  $f: \{0,1\}^n \rightarrow \{0,1\}^m$ mit $n, m \geq 1$
- f ist zerlegbar in m Boolesche Funktionen mit den gleichen n Eingangsvariablen :
  $f_1(x_1, x_2, ..., x_n)$, $f_2(x_1, x_2, ..., x_n)$, ... , $f_m(x_1, x_2, ..., x_n)$



- ein Schaltnetz bezeichnet man auch als kombinatorische Logik

Schaltnetze

- Schaltnetze können einstufig (eine Gatterebene), zweistufig (zwei Gatterebenen) oder mehrstufig sein;
- jedes Schaltnetz ist als gerichteter, ayzklischer Graph darstellbar:
  - Gatter, Ein- und Ausgänge sind die Knoten
  - Verbindungsleitungen entsprechen den gerichteten Kanten
  - Zyklen (Rückkopplungen) sind nicht zulässig!
- aus der Darstellung als kanonische Normalform resultiert, daß jede Schaltfunktion $f(x_1, x_2, ..., x_n)$ durch ein zweistufiges Schaltnetz realisierbar ist, wenn
  - alle Eingangssignale x_i sowohl einfach als auch negiert vorliegen
  - Gatter mit einer ausreichenden Größe (d.h. Anzahl an Eingängen) zur Verfügung stehen

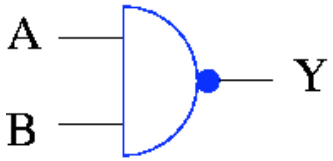
Schaltnetze

- bei Realisierung einer KDNF werden benötigt:
  - je Minterm ein UND-Gatter
  - ein ODER-Gatter zur Disjunktion aller Minterme
- bei Realisierung einer KKNF werden benötigt:
  - je Maxterm ein ODER-Gatter
  - ein UND-Gatter zur Konjunktion aller Maxterme
- Anzahl der benötigten Gatter zur Realisierung der KDNF (bzw. KKNF) einer *n*-stelligen Schaltfunktion:
  - je Boolescher Funktion max. 2^n UND-Gatter (bzw. ODER-Gatter) mit jeweils max. *n* Eingängen
  - je Boolescher Funktion ein ODER-Gatter (bzw. UND-Gatter) mit max. 2^n Eingängen
- Minimierung der Booleschen Funktionen reduziert Anzahl und Größe der benötigten Gatter

Schaltnetze

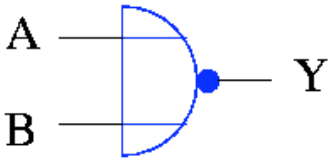
- Jedes Schaltnetz kann nur mit UND-, ODER- und NICHT-Gattern aufgebaut werden.
- Es ist allerdings kompakter, auch andere Gatterbausteine zu verwenden. Gängige Bausteine sind:

NAND

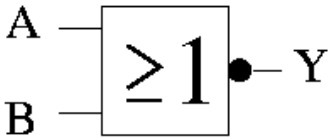


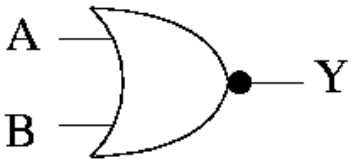
andere Darstellungen:

NOR

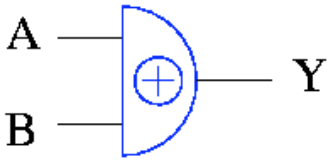


andere Darstellungen:

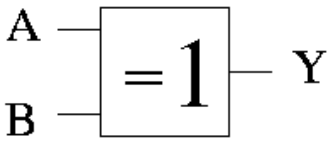


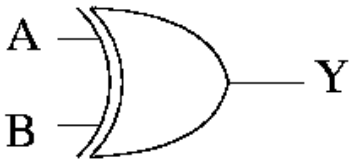


XOR



andere Darstellungen:

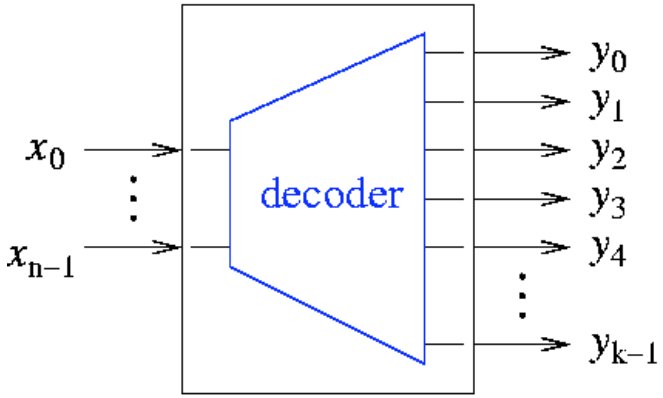




Typische Schaltnetze

- n-zu-k Dekodierer:
  - n Eingänge $x_0, x_1, \ldots, x_{n-1}$
  - $k = 2^n$ Ausgänge $y_0, y_1, \ldots, y_{k-1}$
  - wenn $(x_{n-1}, \ldots, x_1, x_0)_2 = i$, dann ist $y_i = 1$ und $y_{j \neq i} = 0$

- benötigt z.B. zur Dekodierung von Adressen oder Instruktionen



für jede Eingangskombination wird genau 1 Ausgang aktiviert.

Beispiel: Realisierung eines 3-zu-8 Dekodierers:

Eingaben			Ausgaben							
x2	x1	x0	y0	y1	y2	y3	y4	y5	y6	y7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

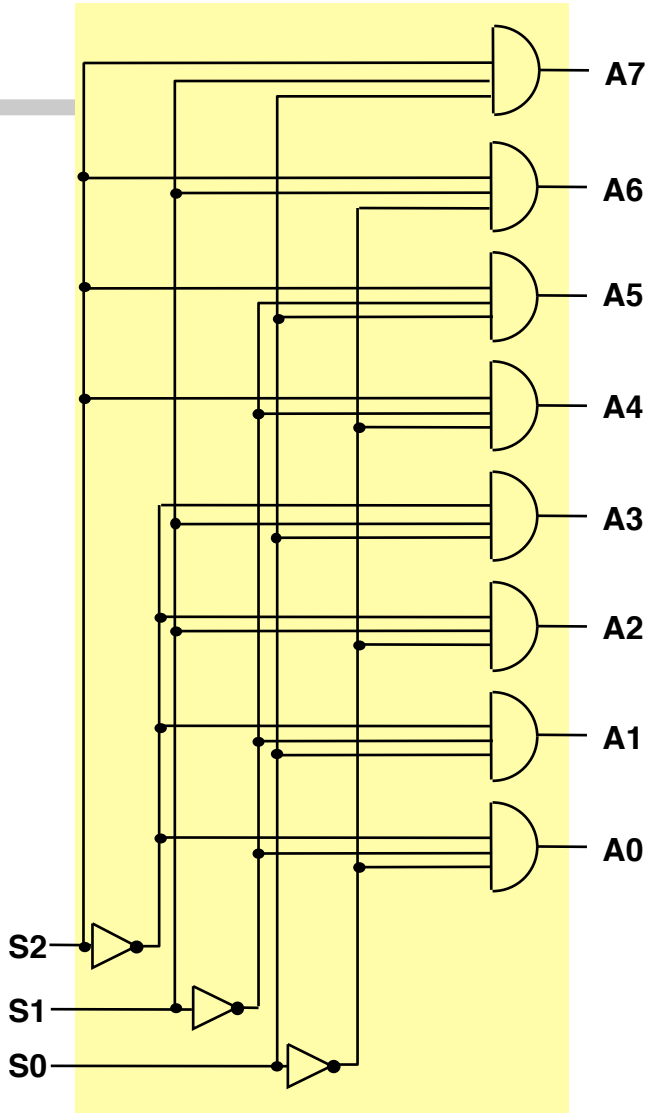
$$y_0 = \overline{x}_2\,\overline{x}_1\overline{x}_0 \qquad y_4 = x_2\,\overline{x}_1\,\overline{x}_0$$

$$y_1 = \overline{x}_2\,\overline{x}_1 x_0 \qquad y_5 = x_2\,\overline{x}_1\,x_0$$

$$y_2 = \overline{x}_2\,x_1\overline{x}_0 \qquad y_6 = x_2\,x_1\,\overline{x}_0$$

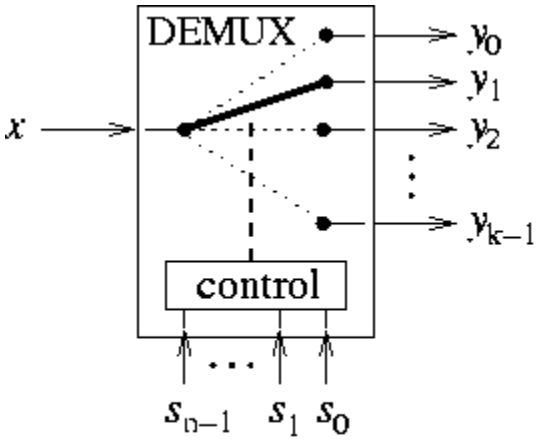
$$y_3 = \overline{x}_2\,x_1 x_0 \qquad y_7 = x_2\,x_1\,x_0$$

Dekodierer



Typische Schaltnetze

- 1-zu-k Demultiplexer:
  - n Steuerleitungen $s_{n-1}, \ldots, s_1, s_0$
  - ein Eingang x
  - $k = 2^n$ Ausgänge $y_0, y_1, \ldots, y_{k-1}$
  - $y_i = x$ für $(s_{n-1}, \ldots, s_1, s_0)_2 = i$



- Beispiel: Realisierung eines 1-zu-4 Demultiplexers:

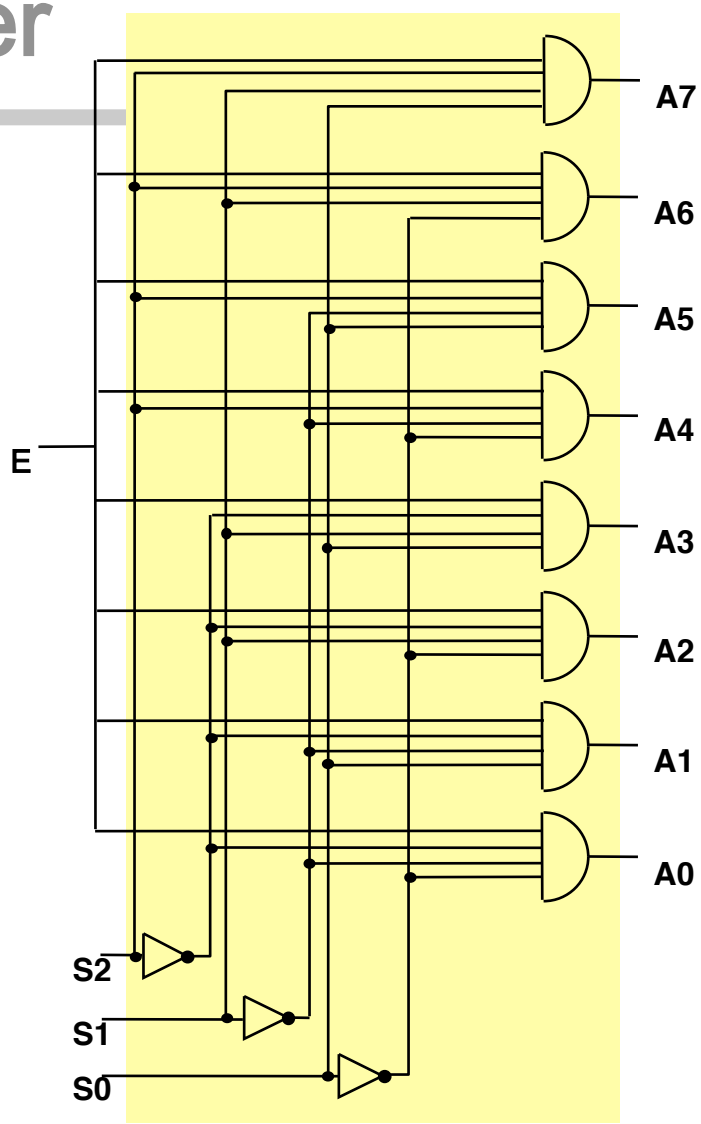
  $y_0 = \overline{s}_1 \overline{s}_0 x, \quad y_1 = \overline{s}_1 s_0 x,$

  $y_2 = s_1 \overline{s}_0 x, \quad y_3 = s_1 s_0 x.$

Die Belegung des Eingangs wird zu genau einem Ausgang propagiert. Der Ausgang wird über die Selektionseingänge ausgewählt.

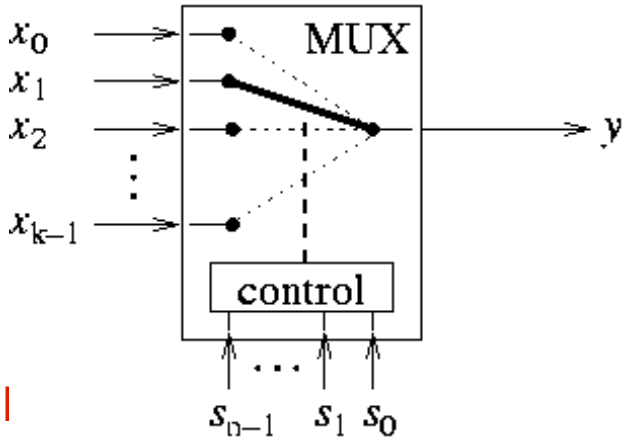
- benötigt z.B. zur Auswahl einer Datensenke (E/A-Gerät, Speicherzeile)

Demultiplexer



Typische Schaltnetze

- 1-aus-k Multiplexer:
  - n Steuerleitungen: $s_{n-1}, \ldots, s_1, s_0$
  - $k = 2^n$ Eingänge: $x_0, x_1, \ldots, x_{k-1}$
  - ein Ausgang: y
  - $y = x_i$ für $(s_{n-1}, \ldots, s_1, s_0)_2 = i$ ← Binärzahl



- Beispiel: Realisierung eines 1-aus-4 Multiplexers

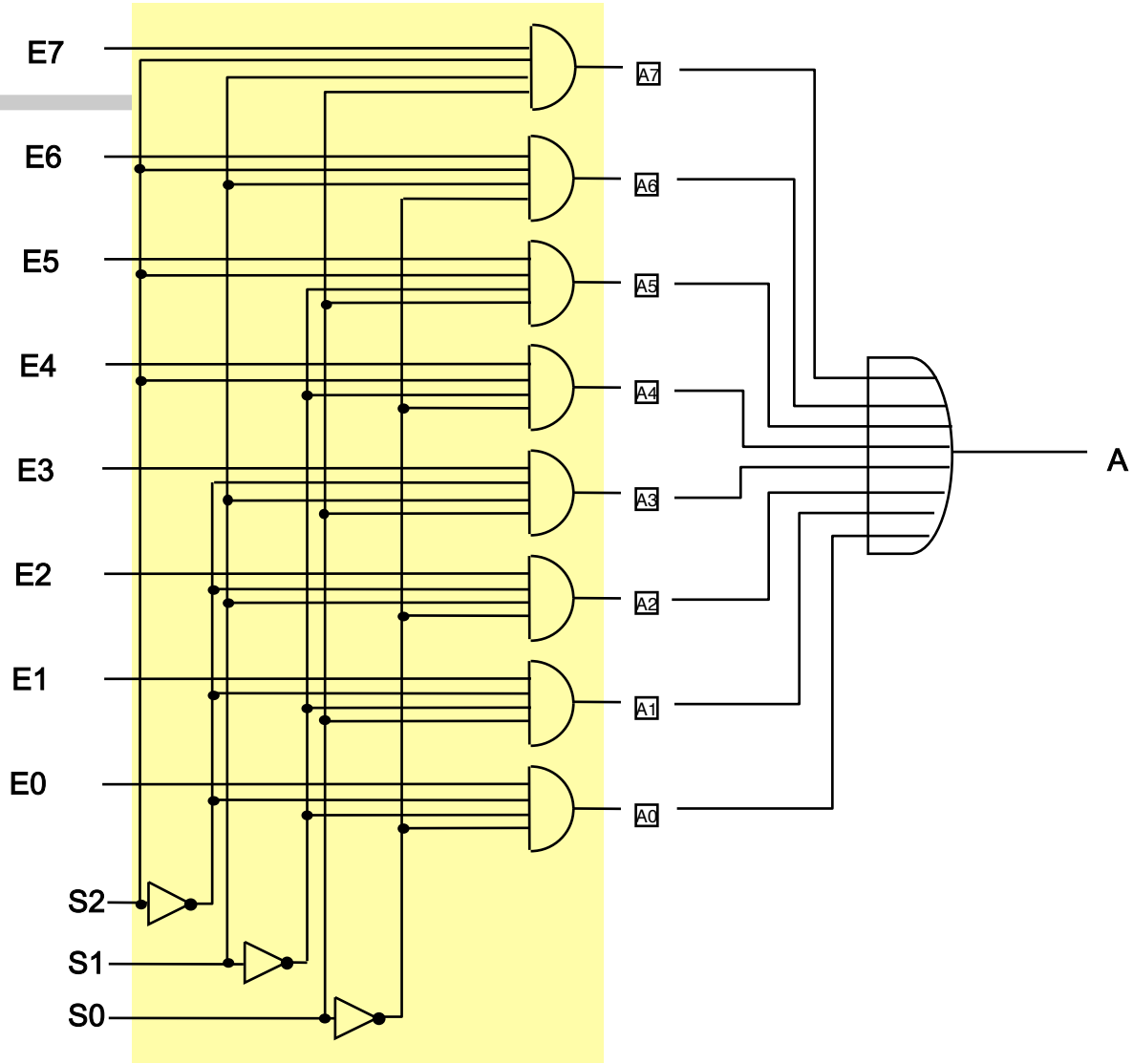
  KDNF:

$$y = \overline{s_1}\,\overline{s_0}x_0 + \overline{s_1}s_0x_1 + s_1\overline{s_0}x_2 + s_1s_0x_3$$

Genau ein durch die Belegung der Selektionseingänge gewählter Eingang wird zum Ausgang propagiert.

- benötigt z.B. zur Auswahl einer Datenquelle

Multiplexer

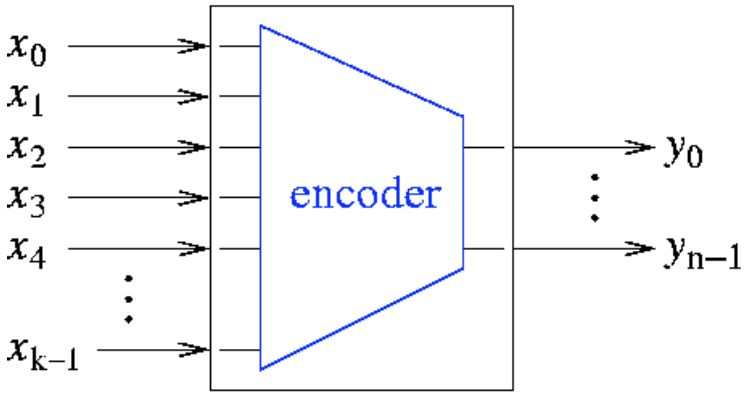


Typische Schaltnetze

- k-zu-n Kodierer:
  - n Ausgänge $y_0, y_1, \ldots, y_{n-1}$
  - $k = 2^n$ Eingänge $x_0, x_1, \ldots, x_{k-1}$
  - nur genau eine Eingangsleitung darf auf 1 sein:

    $x_i = 1,\ x_{j \neq i} = 0$

$$(y_{n-1}, \ldots, y_1, y_0)_2 = i$$



Jeder Eingangsleitung ist genau eine Kombination der möglichen Belegungen der Ausgangsleitungen zugeordet, z.B. ihre binäre Repräsentation.

Beispiel: Realisierung eines 8-zu-3 Kodierers:

Eingaben								Ausgaben		
x0	x1	x2	x3	x4	x5	x6	x7	y2	y1	y0
1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	1	0	0	1	0	1
0	0	0	0	0	0	1	0	1	1	0
0	0	0	0	0	0	0	1	1	1	1

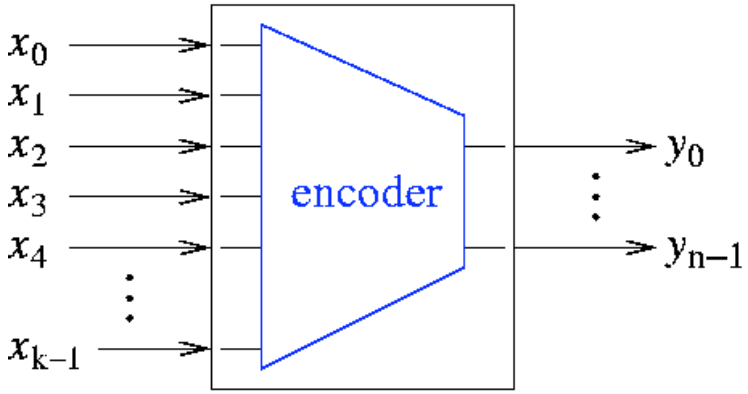
Unvollständige Wahrheitstafel !

$$y_0 = x_1 + x_3 + x_5 + x_7$$

$$y_1 = x_2 + x_3 + x_6 + x_7$$

$$y_2 = x_4 + x_5 + x_6 + x_7$$

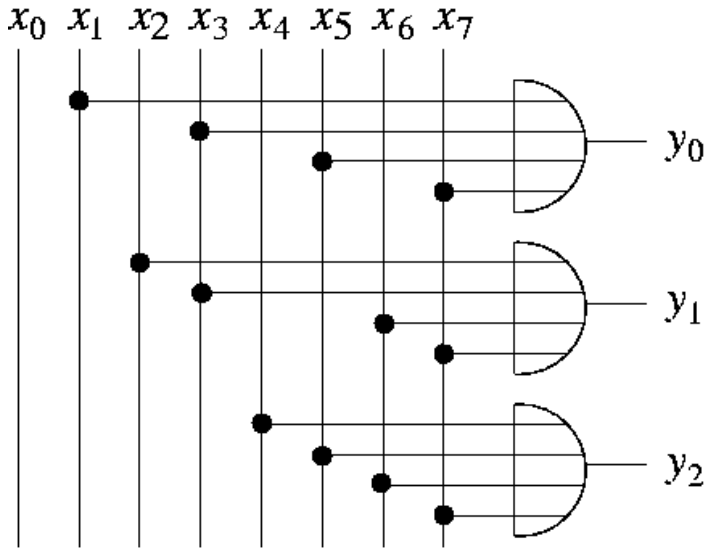
Beispiel: Realisierung eines 8-zu-3 Kodierers:



$y_0 = x_1 + x_3 + x_5 + x_7$

$y_1 = x_2 + x_3 + x_6 + x_7$

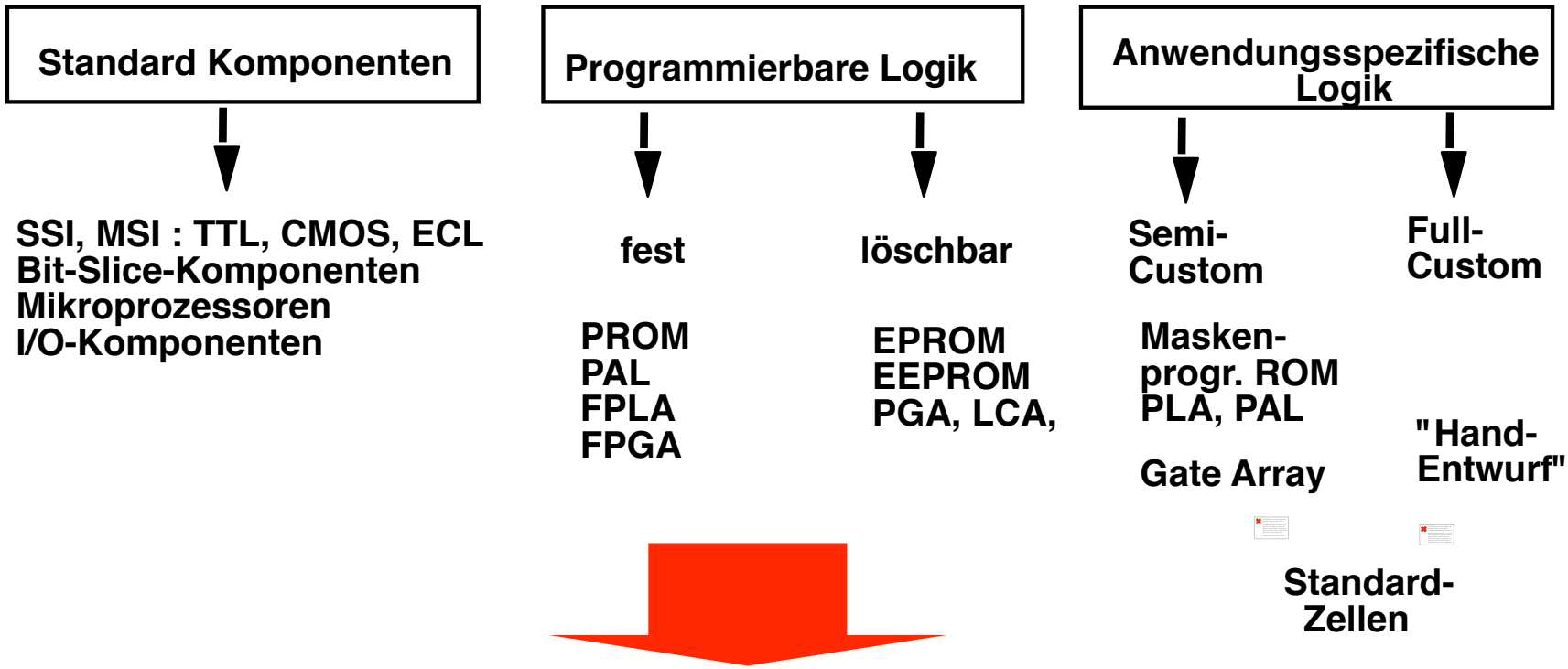
$y_2 = x_4 + x_5 + x_6 + x_7$



Realisierung eines Schaltnetzes

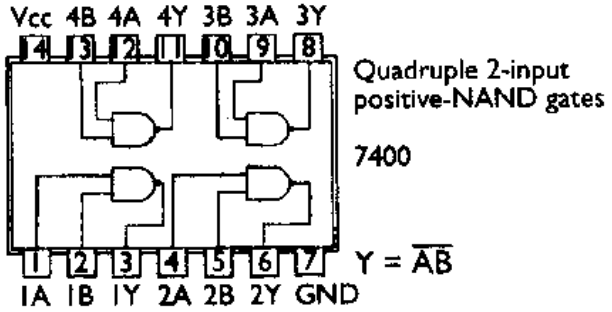
- Ein Schaltnetz wird heute nicht mehr mit diskreten Gattern aufgebaut, statt dessen verwendet man:
  - integrierte Bausteine (ICs)
    - Standardfamilien in TTL (Transistor-Transistor-Logik) oder CMOS („*Complementary Metal Oxide Semiconductor*“):
    - bei SSI (Small Scale Integration) mehrere Gatter des gleichen Typs je Baustein
  - programmierbare Logikbausteine
    - PROM
    - PLA
    - PAL/GAL
    - FPGAs („*Field-Programmable Gate Arrays*“)
  - hochintegrierte Schaltkreise (ASICs = „Application-Specific Integrated Circuits“) in VLSI-Technologie (VLSI = „Very Large Scale Integration“)

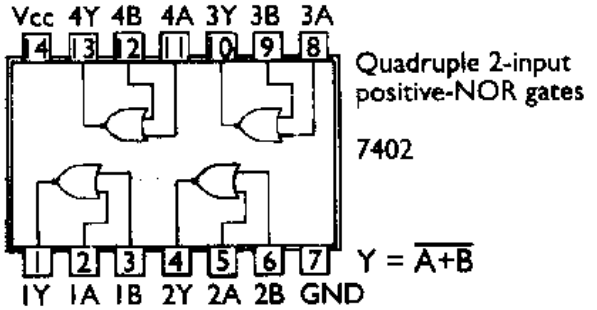
Digitale Logik Familien

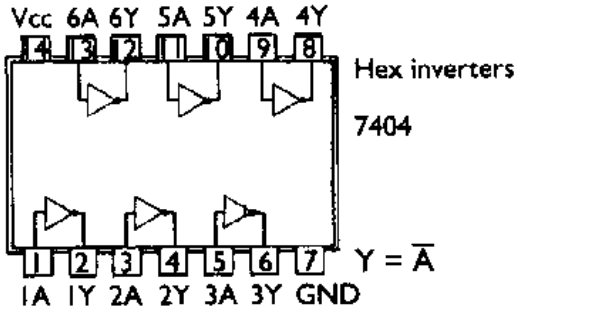


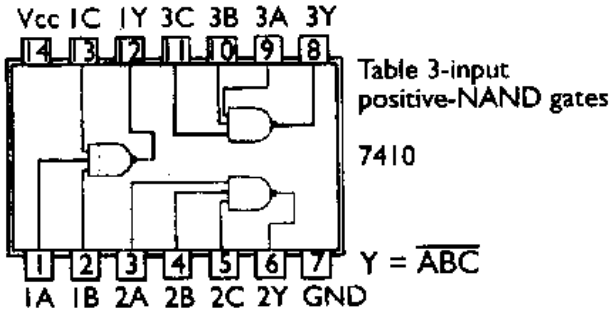
Realisierung eines Schaltnetzes

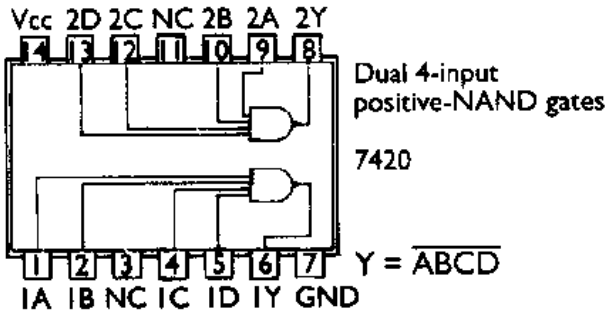
- einige Integrierte Bausteine (TTL-ICs) mit Gattern:

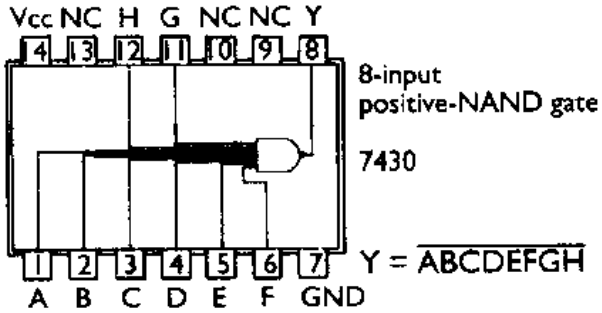












Ziel: Das universelle programmierbare Gatter

Die 16 zweistelligen Boolschen Funktionen

a	b	f_0	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_A	f_B	f_C	f_D	f_E	f_F
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

Implikation

f_1: AND

f_6: EXOR

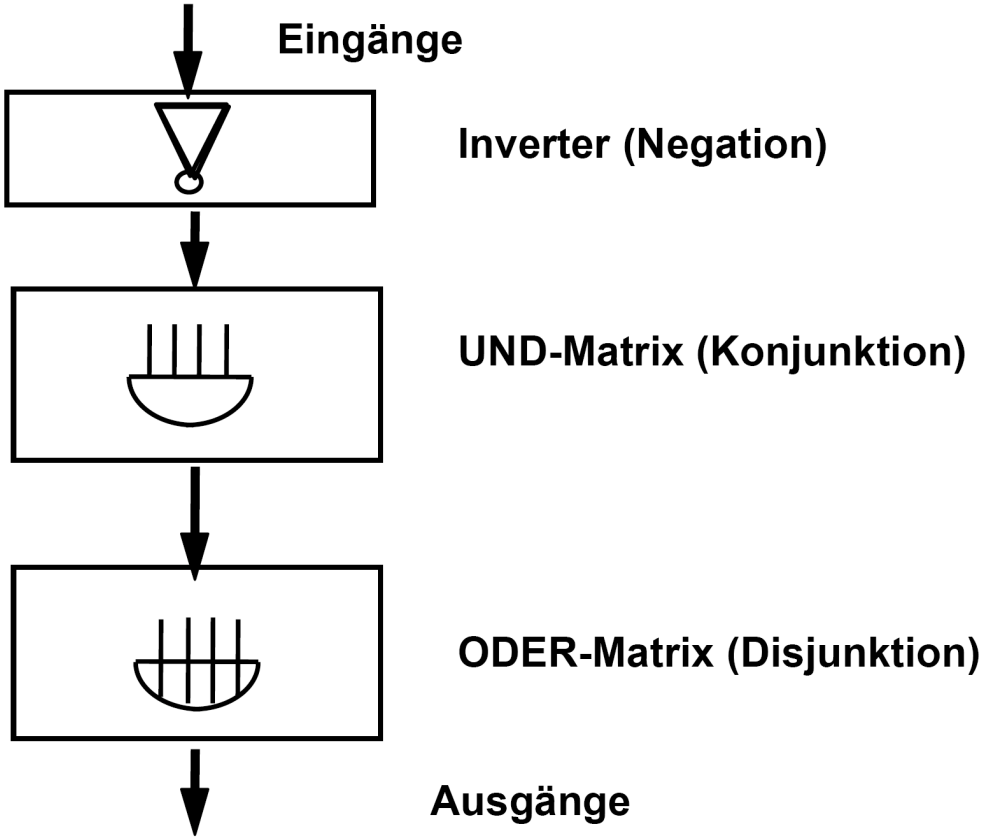
f_7: OR

f_E: NAND

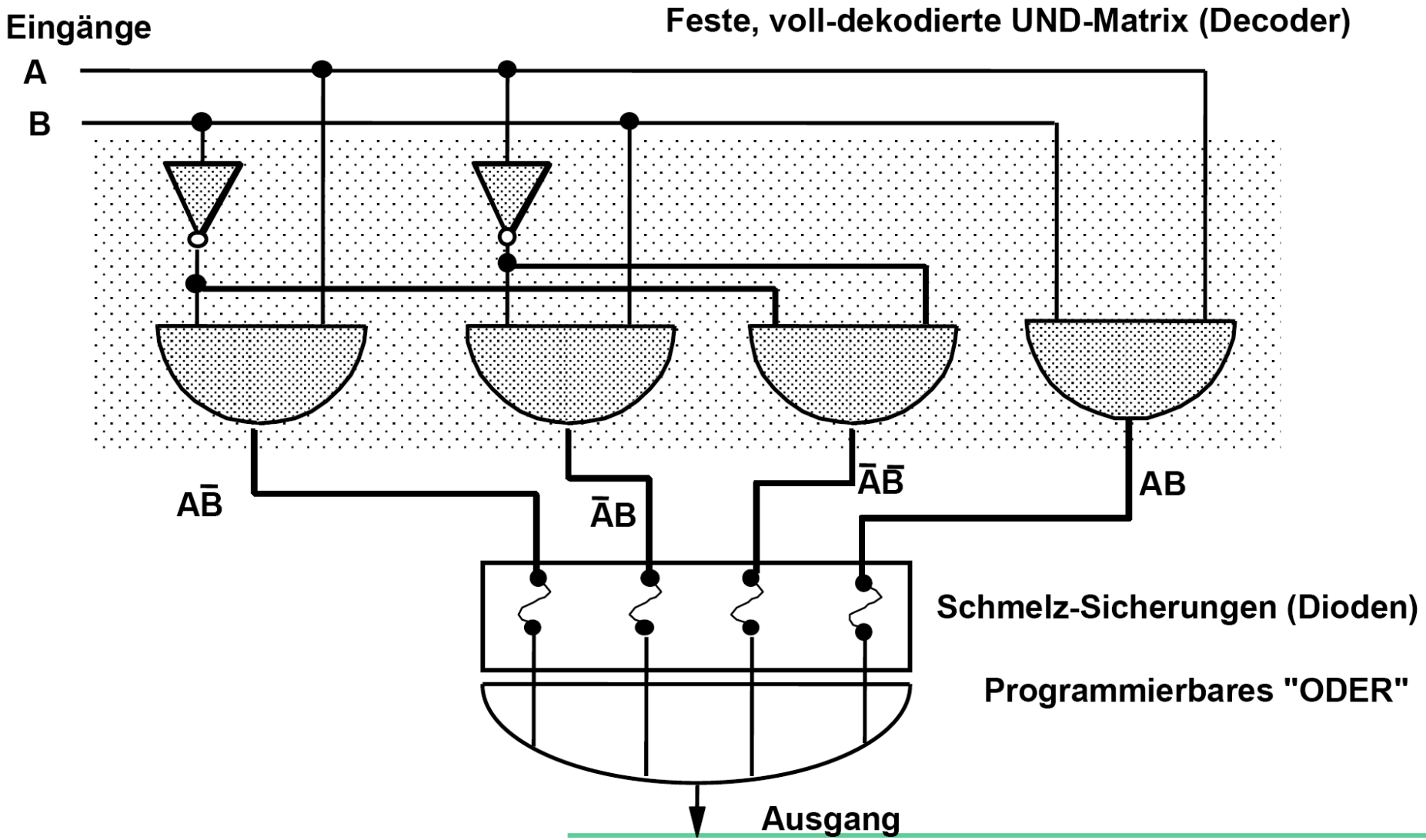
f_8: NOR

Bausteine zur Realisierung beliebiger Logikfunktionen (Grundstruktur)

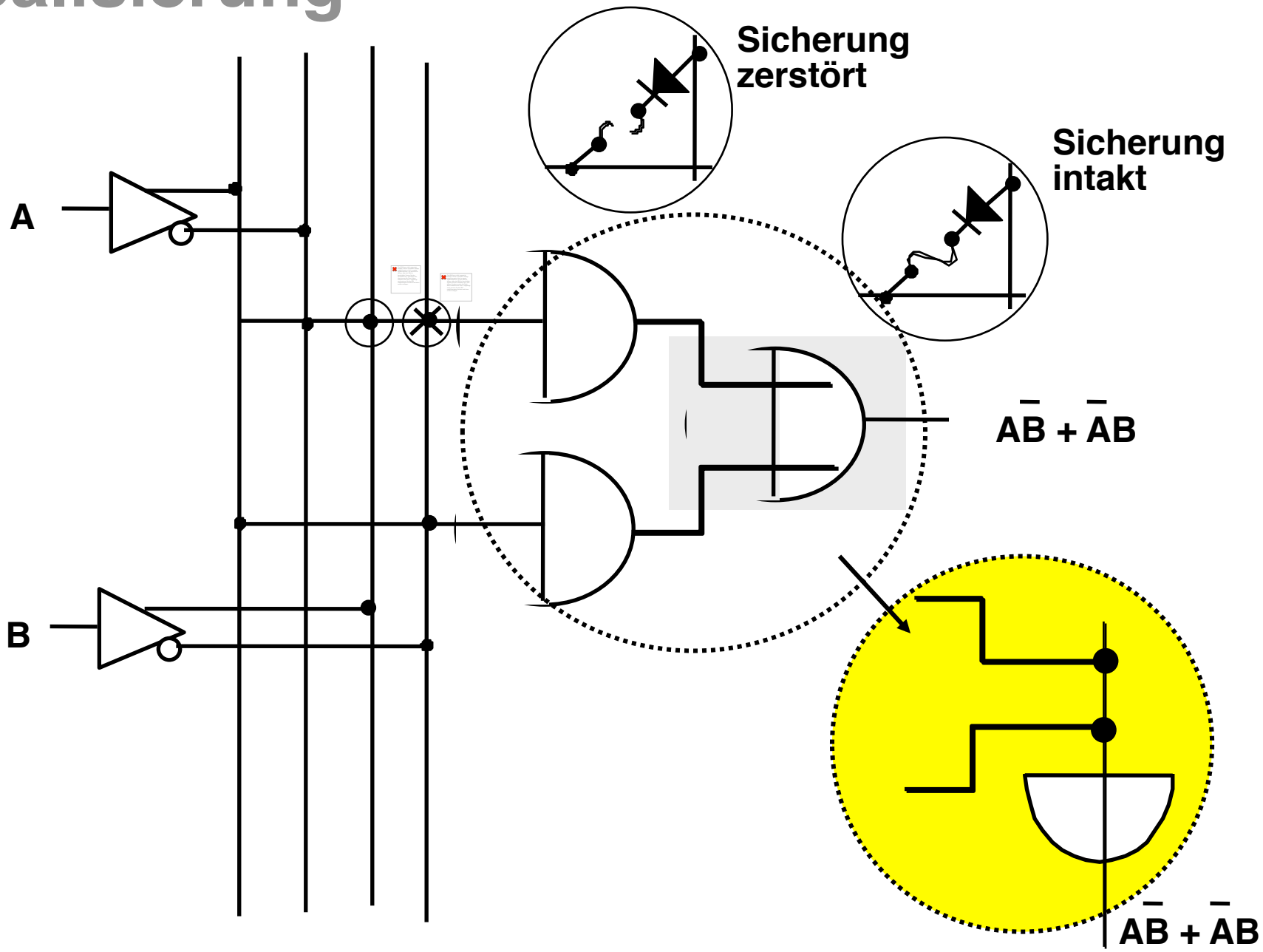
Voraussetzung: Jede Boolsche Funktion kann in Disjunktiver Normalform (Summe von Produkten) dargestellt werden.



Universalgatter (PROM-Ansatz)

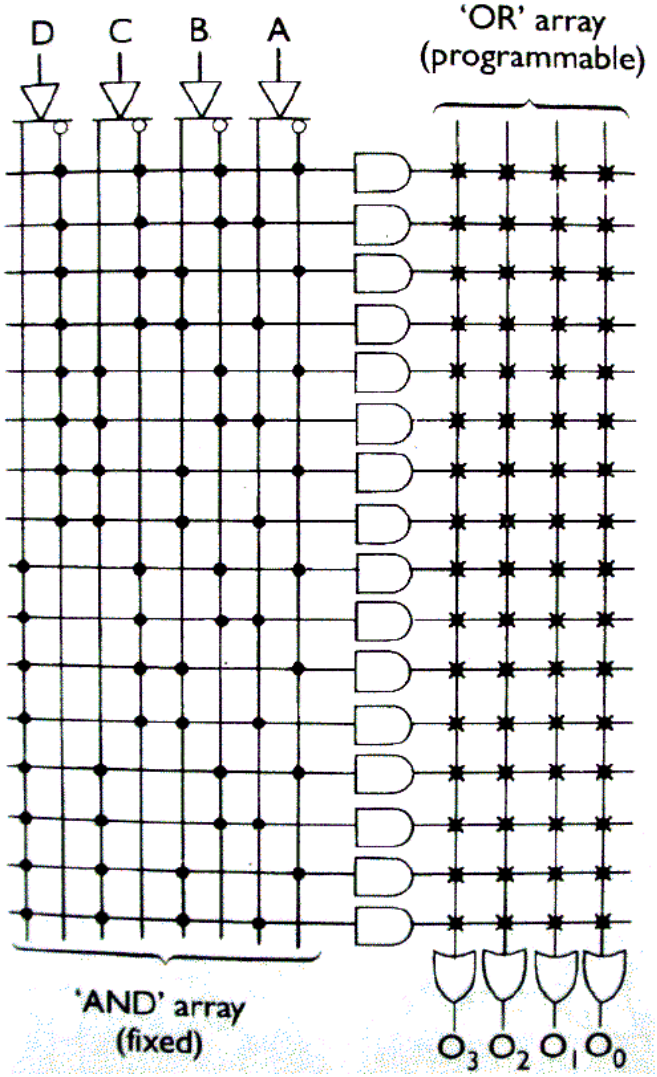


Realisierung

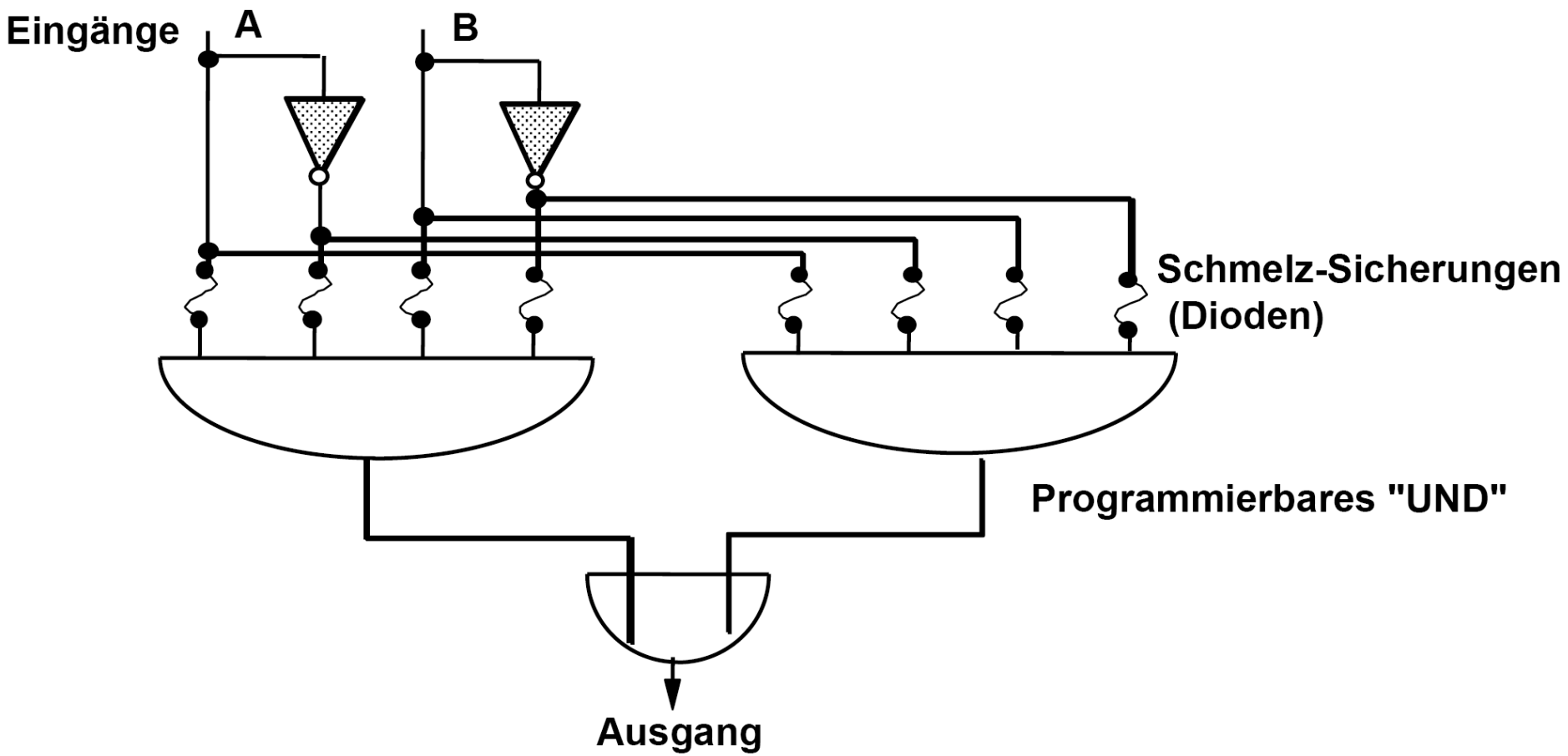


Realisierung eines Schaltnetzes

- Realisierung mit PROM („*Programmable Read-Only Memory*“)
  - Schematische Darstellung (hier ein PROM mit 16 Worten zu 4 Bit):
    - Adressdekoder entspricht einer festen UND-Matrix
    - Koppelelemente bilden eine programmierbare ODER-Matrix
  - realisiert unmittelbar die Wahrheitstabelle in Hardware!
  - ein PROM mit 2^m n-Bit Worten kann jede beliebige Schaltfunktion $f: \{0,1\}^m \rightarrow \{0,1\}^n$ ohne Minimierung implementieren

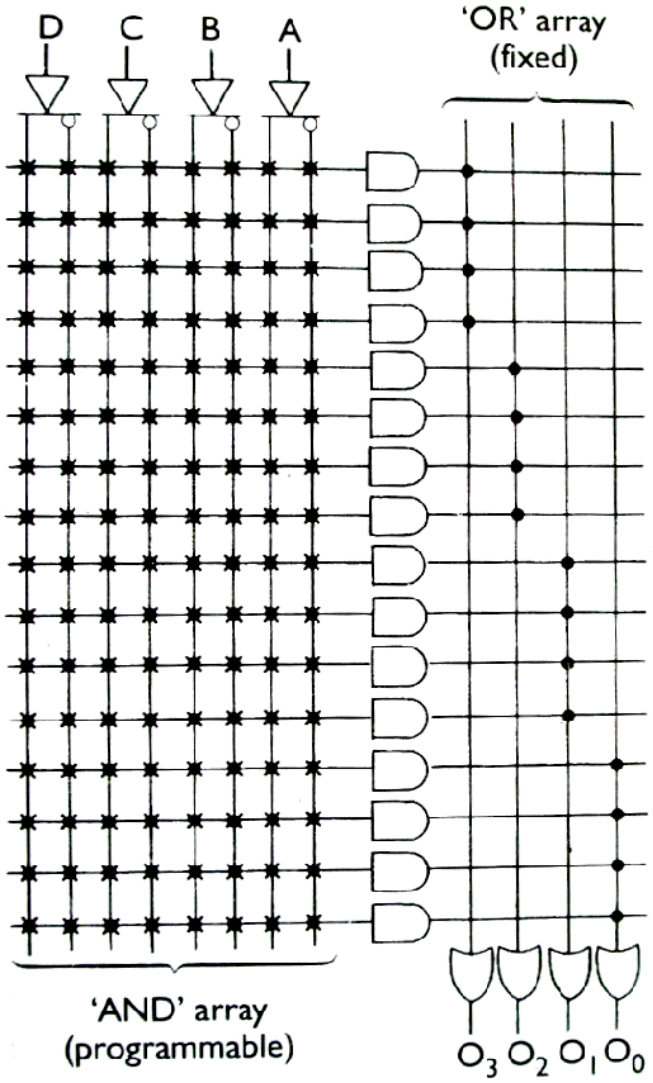


Universalgatter (PAL-Ansatz)

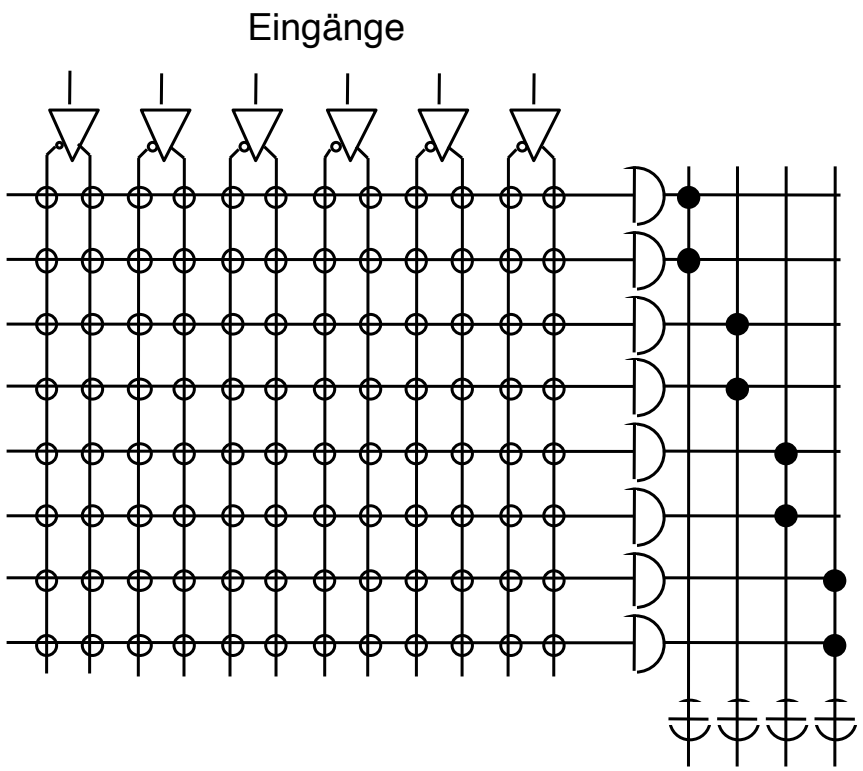


Realisierung eines Schaltnetzes

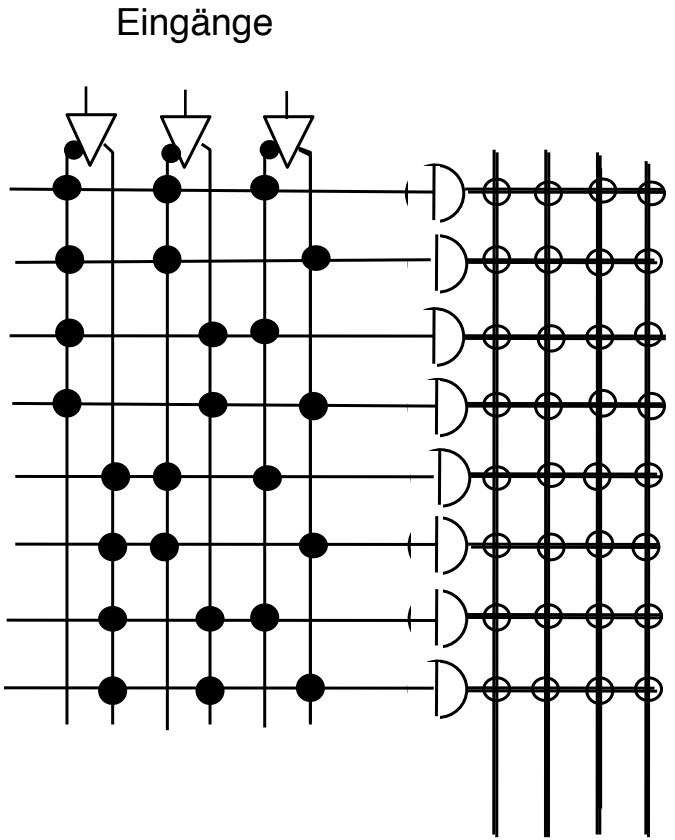
- Realisierung mit PAL / GAL („*Programmable* / *Generic Array Logic*“)
  - Schematische Darstellung eines PAL-Bausteins (hier mit 4 Ein- und Ausgängen und 4 Produkttermen je Ausgang):
  - Aufbau:
    - programmierbare UND-Matrix
    - Produktterme werden mit fester ODER-Matrix verknüpft
  - GAL ist wiederprogrammierbar
  - kann jede (ggf. minimierte) DNF realisieren, wenn Zahl der Produktterme je ODER ausreicht



Strukturvergleich



PAL-Struktur



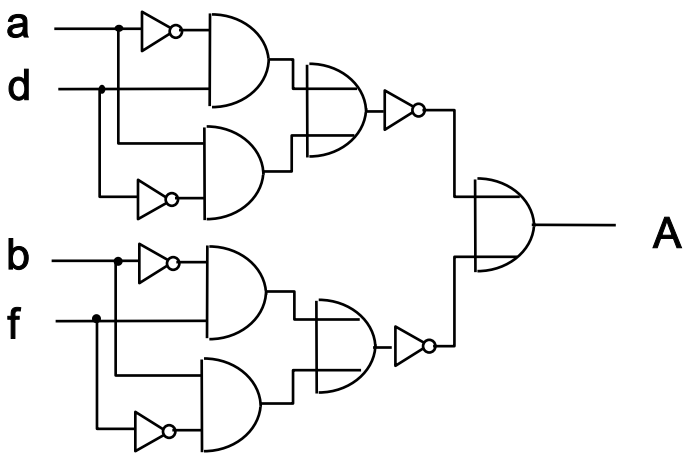
PROM-Struktur

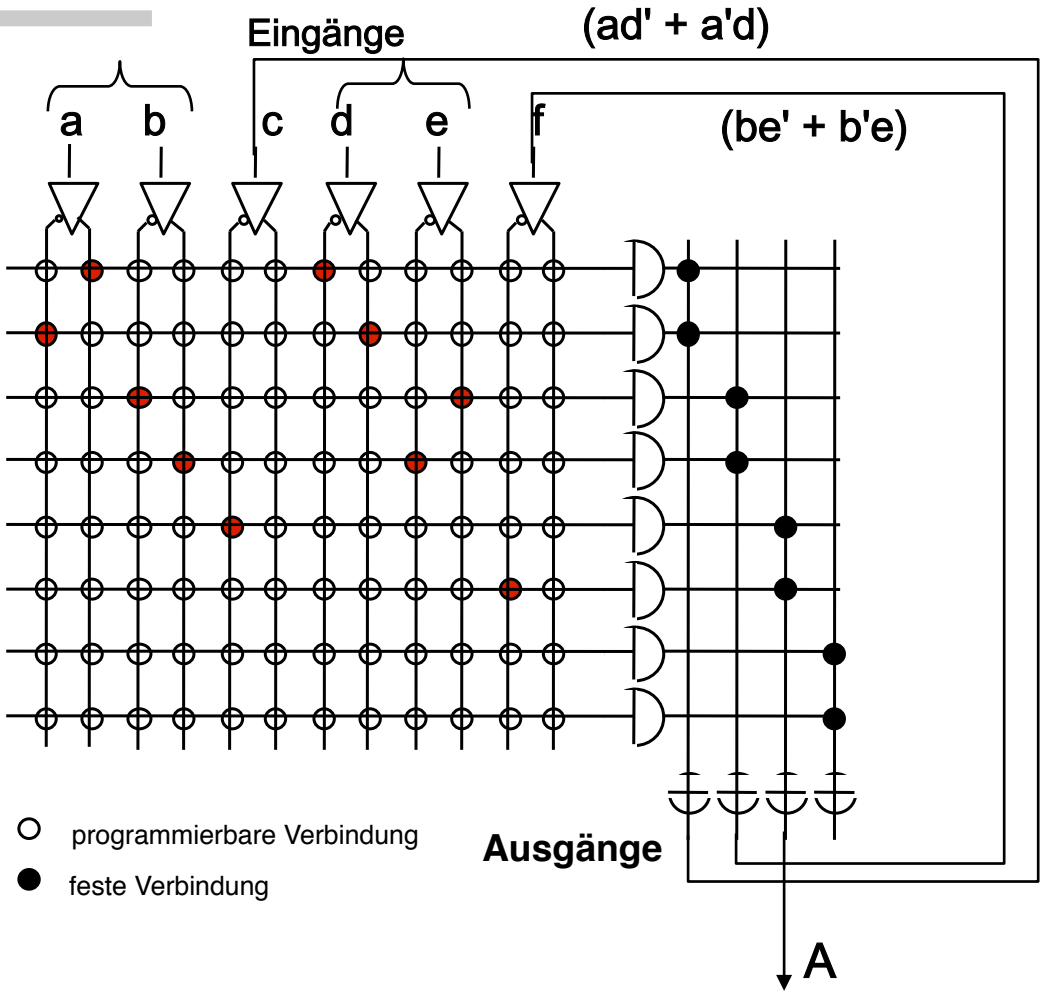
○ programmierbare Verbindung

● feste Verbindung

Beispiel: 2-Bit Komparator

A = (ad' + a'd)' + (be' + b'e)'

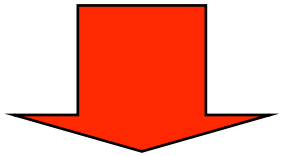




(ad' + a'd)' + (be' + b'e)'

Einige charakteristische Merkmale für PALs:

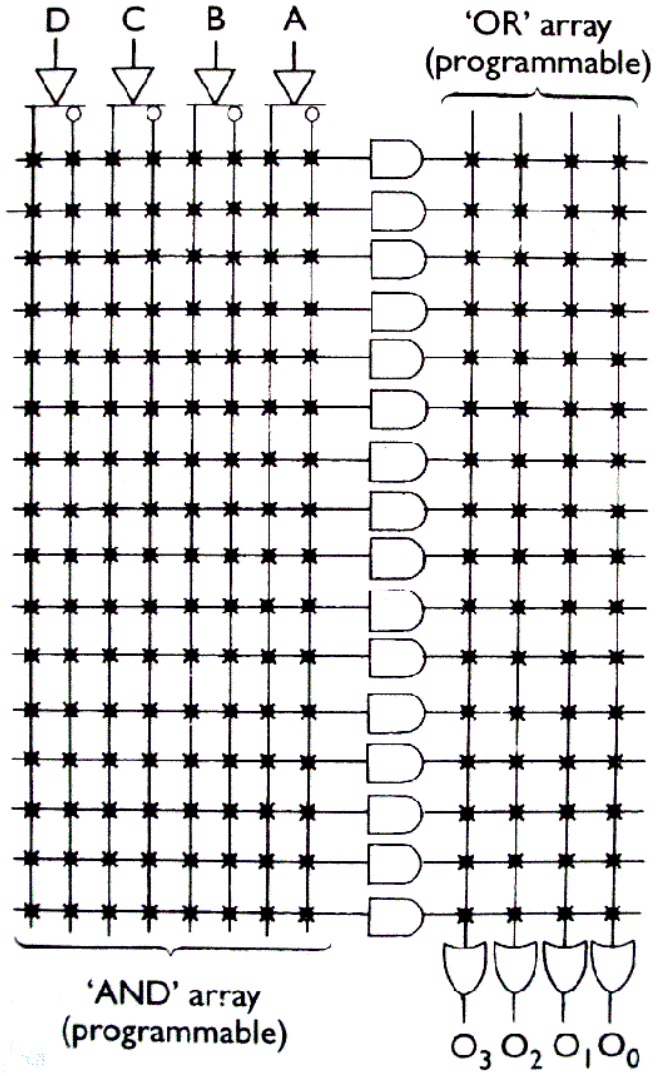
- Die Anzahl der Eingänge bestimmt die Anzahl der möglichen Variablen.
- Die Anzahl der ODER-Verknüpfungen bestimmt die Anzahl der möglichen Terme in der disjunktiven Normalform.



- PAL, wenn viele Variablen und relative wenige Terme. Viele Eingangsbelegungen werden auf dieselbe Ausgangs-belegung abgebildet.
- ROM, wenn jede Eingangsbelegung auf eine individuelle Ausgangsbelegung abgebildet werden muss.

Realisierung eines Schaltnetzes

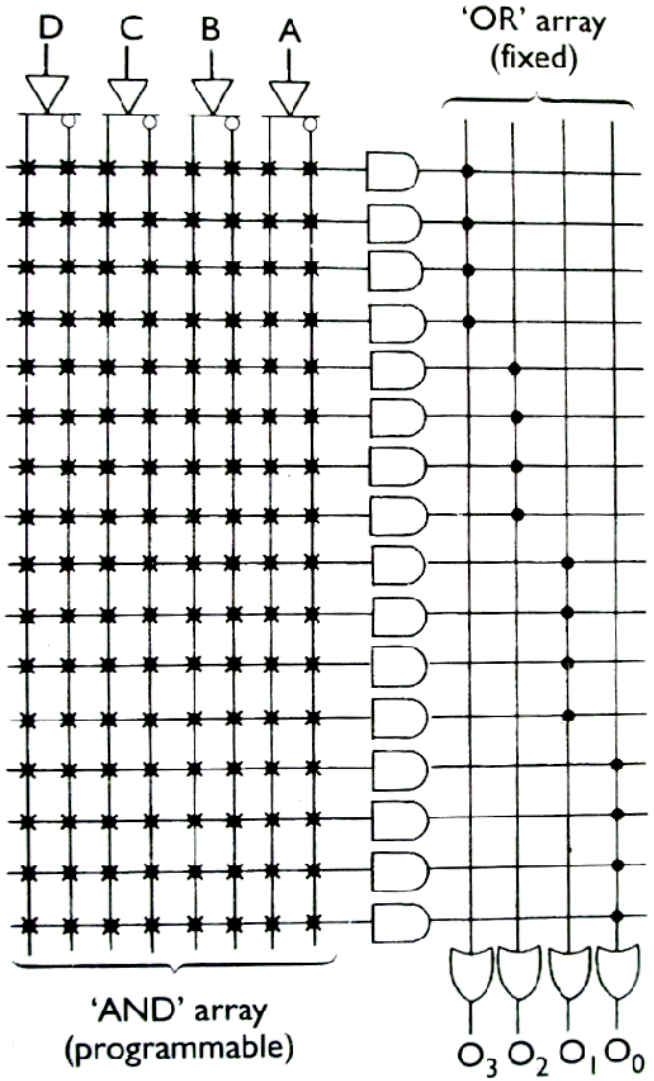
- Ultimative Flexibilität: PLA („*Programmable Logic Array*“)
  - Schematische Darstellung eines PLA-Bausteins (hier mit 4 Ein- und Ausgängen und 16 Produkttermen):
  - Aufbau:
    - programmierbare UND-Matrix
    - programmierbare ODER-Matrix
  - kann jede DNF für n Eingänge realisieren, wenn die Gesamtzahl der Produktterme im PLA ausreicht



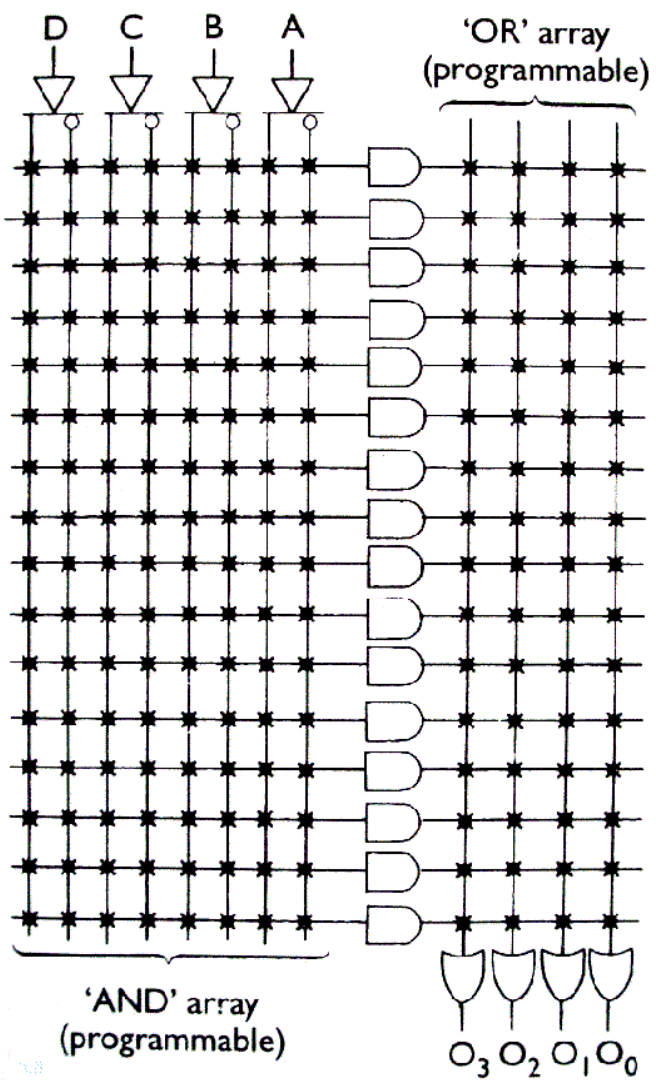
- **zu kompliziert**
- **wenige Anwendungen**

Strukturvergleich

PAL/GAL



PLA



Lernziele

- **Begriffe:** Schaltfunktion, Schaltnetz, Minterm, Maxterm, DNF, KNF, KDNF, KKNF, (De-)Multiplexer, (De-)Kodierer, ...
- **Transformation Boolescher Ausdrücke** gemäß den Axiomen und Sätzen der Booleschen Algebra, z.B. zum Nachweis der Äquivalenz zweier Schaltnetze
- **Systematischer Entwurf eines Schaltnetzes** aus einer Problembeschreibung
  - Aufstellen der Wahrheitstabelle
  - Bildung der KDNF oder KKNF
  - Minimierung der KDNF oder KKNF mit Karnaugh-Veitch-Diagrammen
  - Realisierung des Schaltnetzes mit Gattern aus einer vorgegeben Menge oder mit programmierbaren Logikbausteinen