Einführung
in die technische Informatik

Christopher Kruegel chris@auto.tuwien.ac.at

http://www.auto.tuwien.ac.at/~chris

Logische Schaltungen

- System mit
  - Eingängen
  - Ausgängen
  - interne Logik die Eingänge auf Ausgänge abbildet

- Einfache Schaltungen
  - kein innerer Zustand
  - bestimmter Eingangszustand wird immer auf den selben Ausgangszustand abgebildet
  - Beispiel
    - Multiplexer, Addierer

Logische Schaltungen

- Komplexe (sequentielle) Schaltungen
  - haben inneren Zustand
  - bestimmter Eingangszustand kann auf unterschiedliche Ausgangszustände abgebildet werden
  - Beispiel
    - Zähler

- Logische Schaltungen können auf unterschiedlichen Abstraktionsebenen betrachtet werden

➢ In dieser Vorlesung sind logische Grundschaltungen (Grundgatter) die Basisbausteine
  - (N)AND Gatter, (N)OR Gatter, NOT Gatter
  - keine Transistoren werden mehr berücksichtigt

Logische Schaltungen

- Einfache logische Schaltungen
  - können durch Boolsche Funktionen beschrieben werden
  - Eingangszustand wird durch Eingangsvariablen beschrieben
  - Ausgangszustand wird durch Ausgangsvariablen beschrieben
  - jeder Ausgangsvariable kann eine Funktion zugeordnet werden, die von allen Eingangsvariablen abhängen kann

Boolsche Funktionen

- Boolsche Funktion

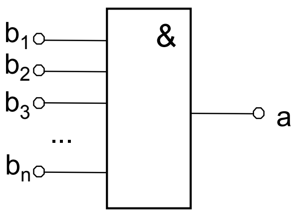
$$f(b_1, b_2, ..., b_n) \rightarrow \{true, false\}$$

- Wahrheitstabelle
  - 2^n Einträge, einen für jede mögliche Belegung

b_1	b_2	...	b_n	f(...)
0	0	0	0	0/1
0	0	0	1	0/1
..	..	..	..	..
1	1	1	1	0/1

Logische Schaltungen

- AND Grundgatter

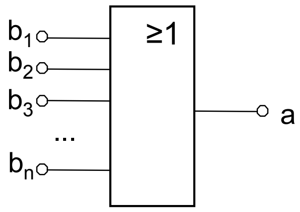


$$b_1 \wedge b_2 \wedge ... \wedge b_n \rightarrow \begin{cases} a = true & if\,(\forall b_i)\; b_i = true \\ a = false & else \end{cases}$$

b_1	b_2	b_3 .. $b_{(n-1)}$	b_n	a
0	0	0 .. 0	0	0
1	0	0 .. 0	0	0
0	1	0 .. 0	0	0
..	..	..	..	..
1	1	1 .. 1	1	1

Logische Schaltungen

- OR Grundgatter

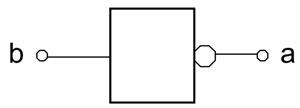


$$b_1 \vee b_2 \vee \ldots \vee b_n \rightarrow \begin{cases} a = true & if\ (\exists b_i)\ b_i = true \\ a = false & else \end{cases}$$

b_1	b_2	b_3 .. $b_{(n-1)}$	b_n	a
0	0	0 .. 0	0	0
1	0	0 .. 0	0	1
0	1	0 .. 0	0	1
..	..	..	..	..
1	1	1 .. 1	1	1

Logische Schaltungen

- NOT Grundgatter



$$\neg b \rightarrow \begin{cases} a = true & if\ (b = false) \\ a = false & else \end{cases}$$

b	a
0	1
1	0

Beispiel - Wahrheitstabelle

Die Wahrheitstabelle kann zu jeder Funktion erstellt werden.

Beispiel:

$$(A \wedge B \wedge C \wedge D) \vee$$
$$(A \wedge B \wedge \neg C \wedge D) \vee$$
$$(A \wedge B \wedge C \wedge \neg D) \vee$$
$$(A \wedge B \wedge \neg C \wedge \neg D) \vee$$
$$(C \wedge D)$$

Beispiel - Wahrheitstabelle

Funktion

$$(A \wedge B \wedge C \wedge D) \vee$$
$$(A \wedge B \wedge \neg C \wedge D) \vee$$
$$(A \wedge B \wedge C \wedge \neg D) \vee$$
$$(A \wedge B \wedge \neg C \wedge \neg D) \vee$$
$$(C \wedge D)$$

A	B	C	D	f(..)
0	0	0	0	
0	0	0	1	
0	0	1	0	
0	0	1	1	
0	1	0	0	
0	1	0	1	
0	1	1	0	
0	1	1	1	
1	0	0	0	
1	0	0	1	
1	0	1	0	
1	0	1	1	
1	1	0	0	
1	1	0	1	
1	1	1	0	
1	1	1	1	

Beispiel - Wahrheitstabelle

Funktion

$(A \wedge B \wedge C \wedge D) \vee$

$(A \wedge B \wedge \neg C \wedge D) \vee$

$(A \wedge B \wedge C \wedge \neg D) \vee$

$(A \wedge B \wedge \neg C \wedge \neg D) \vee$

$(C \wedge D)$

A	B	C	D	f(..)
0	0	0	0	
0	0	0	1	
0	0	1	0	
0	0	1	1	
0	1	0	0	
0	1	0	1	
0	1	1	0	
0	1	1	1	
1	0	0	0	
1	0	0	1	
1	0	1	0	
1	0	1	1	
1	1	0	0	
1	1	0	1	
1	1	1	0	
1	1	1	1	1

Beispiel - Wahrheitstabelle

Funktion

$(A \wedge B \wedge C \wedge D) \vee$

$(A \wedge B \wedge \neg C \wedge D) \vee$

$(A \wedge B \wedge C \wedge \neg D) \vee$

$(A \wedge B \wedge \neg C \wedge \neg D) \vee$

$(C \wedge D)$

A	B	C	D	f(..)
0	0	0	0	
0	0	0	1	
0	0	1	0	
0	0	1	1	
0	1	0	0	
0	1	0	1	
0	1	1	0	
0	1	1	1	
1	0	0	0	
1	0	0	1	
1	0	1	0	
1	0	1	1	
1	1	0	0	
1	1	0	1	1
1	1	1	0	
1	1	1	1	1

Beispiel - Wahrheitstabelle

Funktion

$$(A \wedge B \wedge C \wedge D) \vee$$
$$(A \wedge B \wedge \neg C \wedge D) \vee$$
$$(A \wedge B \wedge C \wedge \neg D) \vee$$
$$(A \wedge B \wedge \neg C \wedge \neg D) \vee$$
$$(C \wedge D)$$

A	B	C	D	f(..)
0	0	0	0	
0	0	0	1	
0	0	1	0	
0	0	1	1	
0	1	0	0	
0	1	0	1	
0	1	1	0	
0	1	1	1	
1	0	0	0	
1	0	0	1	
1	0	1	0	
1	0	1	1	
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

Beispiel - Wahrheitstabelle

Funktion

$$(A \wedge B \wedge C \wedge D) \vee$$
$$(A \wedge B \wedge \neg C \wedge D) \vee$$
$$(A \wedge B \wedge C \wedge \neg D) \vee$$
$$(A \wedge B \wedge \neg C \wedge \neg D) \vee$$
$$(C \wedge D)$$

A	B	C	D	f(..)
0	0	0	0	
0	0	0	1	
0	0	1	0	
0	0	1	1	1
0	1	0	0	
0	1	0	1	
0	1	1	0	
0	1	1	1	1
1	0	0	0	
1	0	0	1	
1	0	1	0	
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

Beispiel - Wahrheitstabelle

Funktion

$$(A \wedge B \wedge C \wedge D) \vee$$
$$(A \wedge B \wedge \neg C \wedge D) \vee$$
$$(A \wedge B \wedge C \wedge \neg D) \vee$$
$$(A \wedge B \wedge \neg C \wedge \neg D) \vee$$
$$(C \wedge D)$$

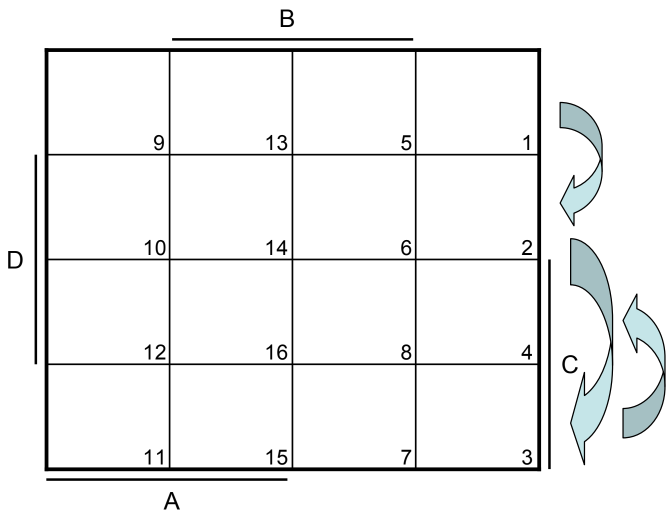
A	B	C	D	f(..)
0	0	0	0	**0**
0	0	0	1	**0**
0	0	1	0	**0**
0	0	1	1	1
0	1	0	0	**0**
0	1	0	1	**0**
0	1	1	0	**0**
0	1	1	1	1
1	0	0	0	**0**
1	0	0	1	**0**
1	0	1	0	**0**
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

Minimierung

- Boolsche Funktionen
  - unterschiedliche, aber gleichwertige Darstellungen möglich
  - disjunktive Normalform, konjunktive Normalform
  - Frage nach minimaler Form
- Minimale disjunktive Normalform
  - Verfahren nach Quine und McClusky
  - Verfahren nach Karnaugh und Veitch (KV-Diagramme)
- Vorteil
  - einfacher in Hardware zu realisieren

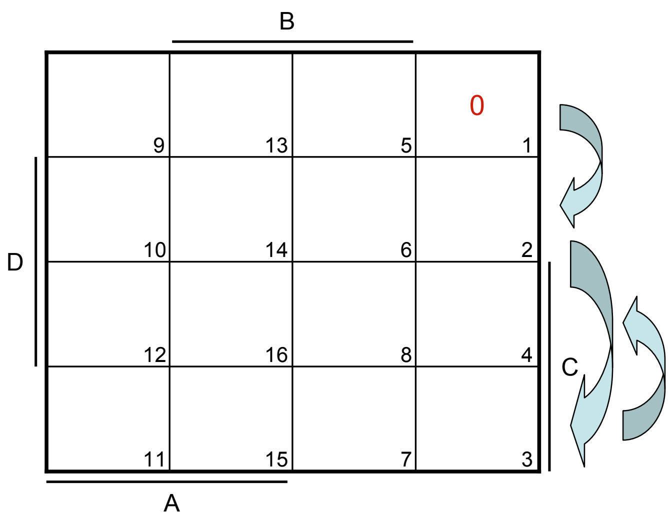
Beispiel - Minimierung

A	B	C	D	f(..)
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1



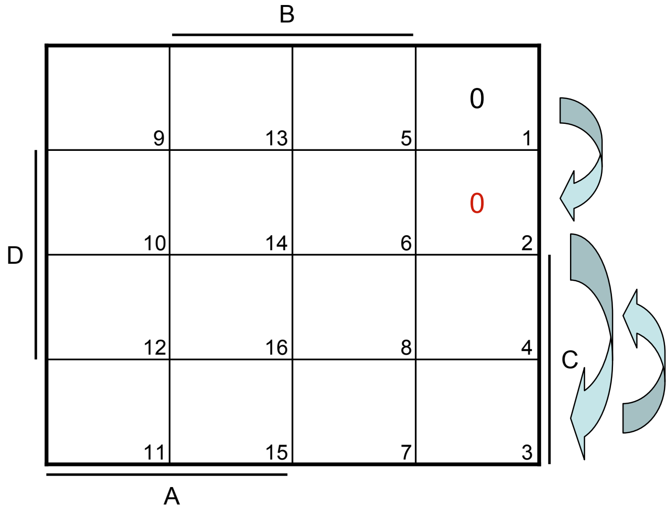
Beispiel - Minimierung

A	B	C	D	f(..)
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1



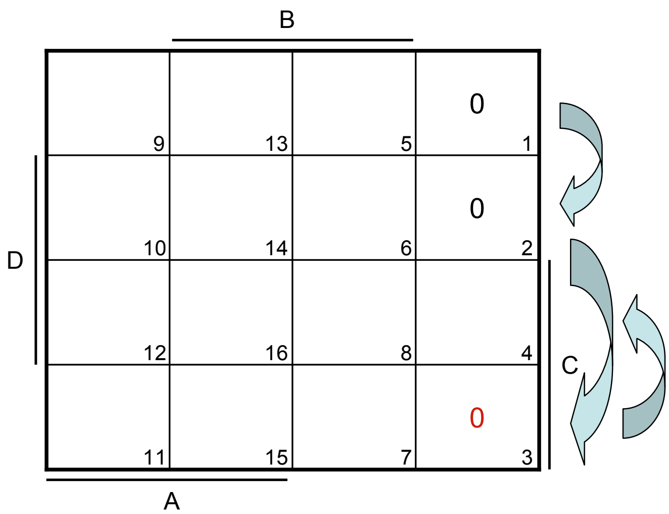
Beispiel - Minimierung

A	B	C	D	f(..)
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1



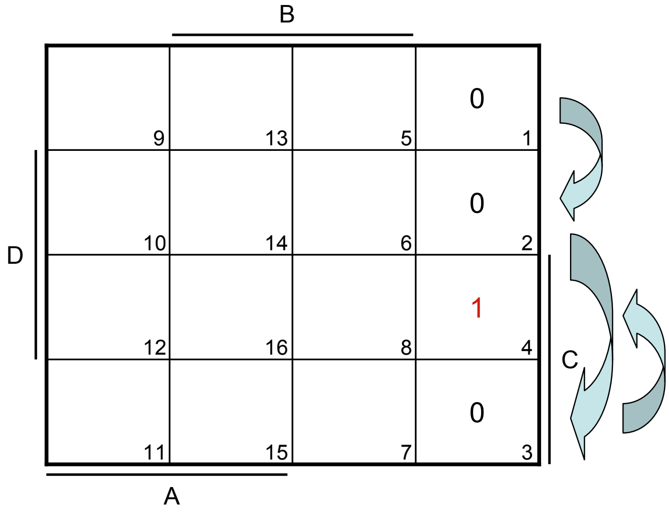
Beispiel - Minimierung

A	B	C	D	f(..)
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1



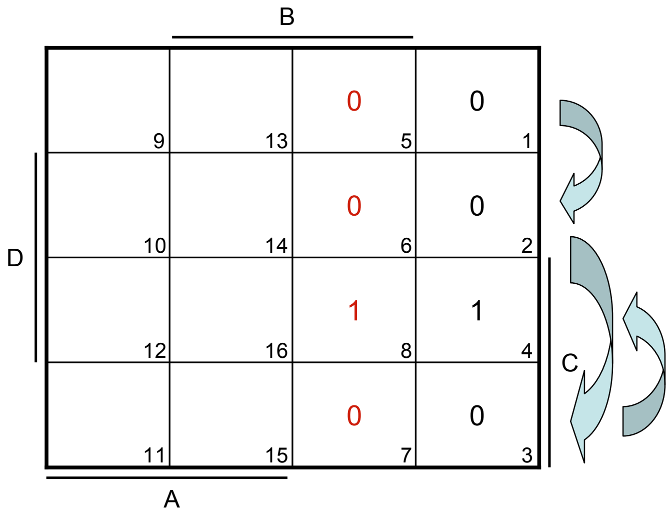
Beispiel - Minimierung

A	B	C	D	f(..)
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1



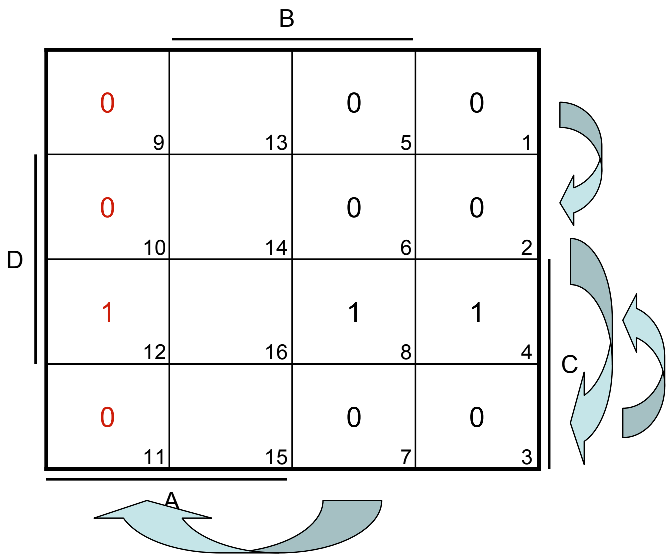
Beispiel - Minimierung

A	B	C	D	f(..)
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1



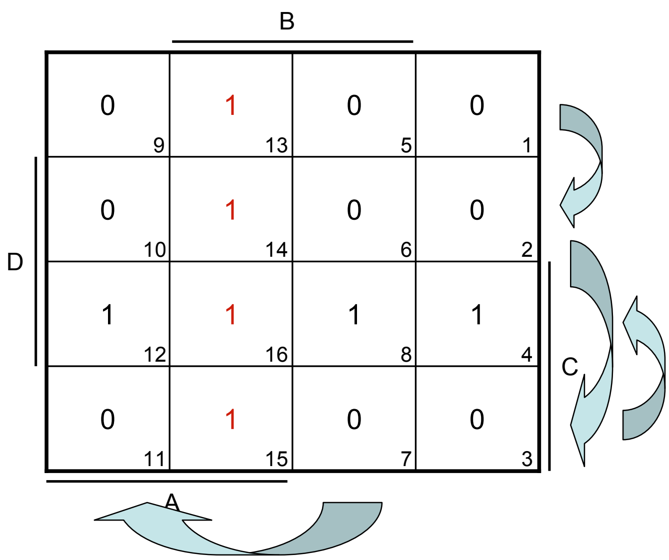
Beispiel - Minimierung

A	B	C	D	f(..)
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1



Beispiel - Minimierung

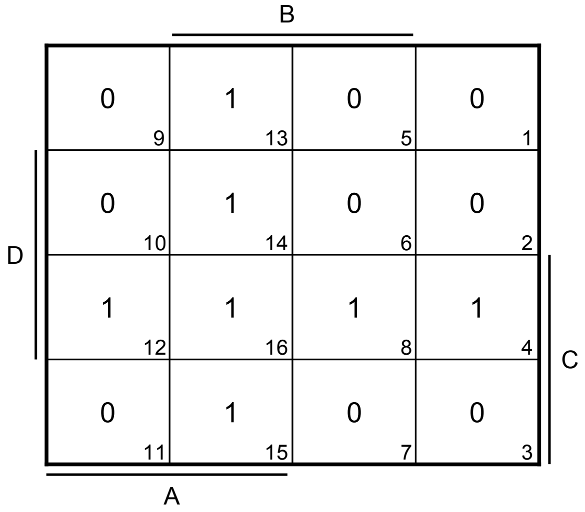
A	B	C	D	f(..)
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1



Beispiel - Minimierung

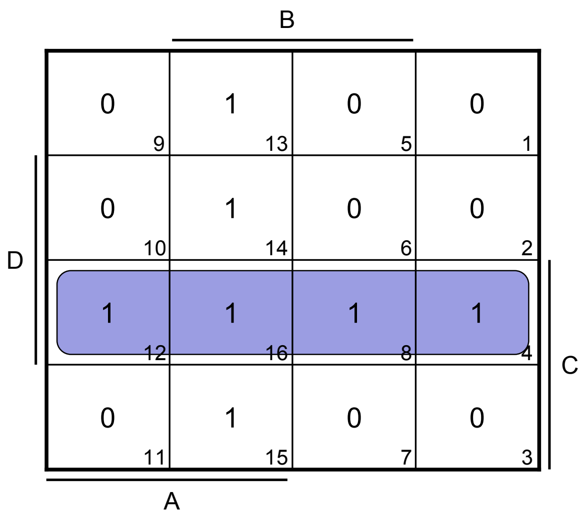
- Blöcke finden
  - größtmöglich
  - alle True-Werte abdecken

  - Randblöcke nicht übersehen
  - Überlappungen sind in Ordnung



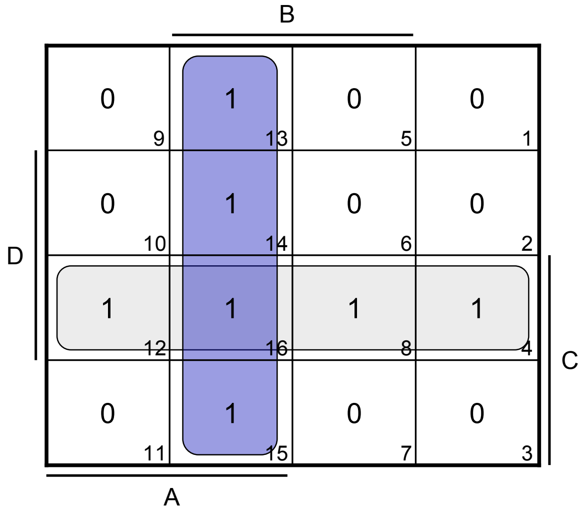
Beispiel - Minimierung

$(C \wedge D)$



Beispiel - Minimierung

$(C \wedge D) \vee$
$(A \wedge B)$



Rechenbeispiele

- Transformation
  - Ändern von vorgegebenen Schaltungen
- Schaltungen
  - Boolsche Funktionen
  - Wahrheitstabellen
  - Minimierung
  - Gatter / PLA

Transformation

- Schaltung nur aus bestimmten Gattern aufbauen
  - z.B., nur NAND
  - z.B., nur NOR
  - alles kann aus diesen Gattern aufgebaut werden

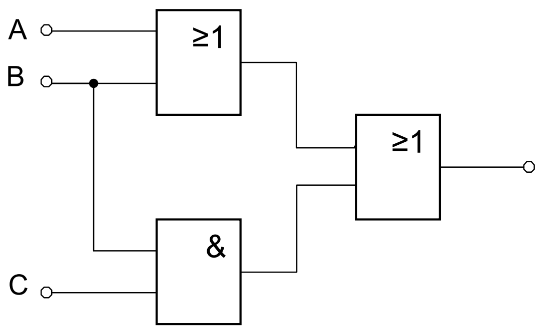
- De Morgan'sche Regeln

$$A \wedge B = \neg(\neg A \vee \neg B)$$

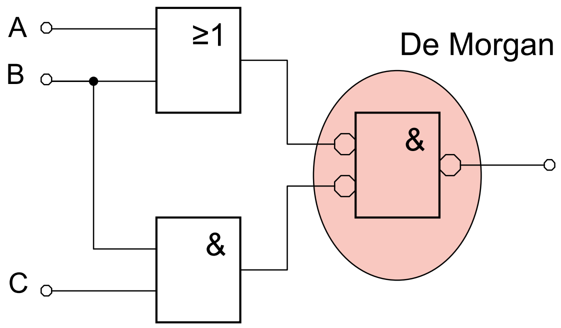
$$A \vee B = \neg(\neg A \wedge \neg B)$$

Beispiel - Transformation

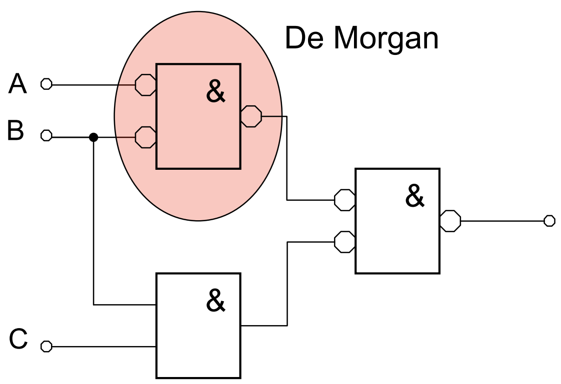
Die folgende Schaltung soll nur aus NAND Gattern aufgebaut werden



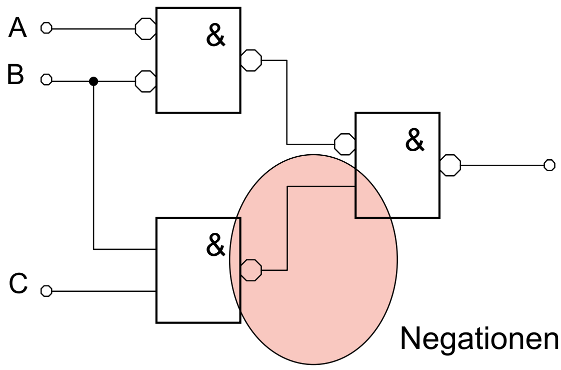
Beispiel - Transformation



Beispiel - Transformation



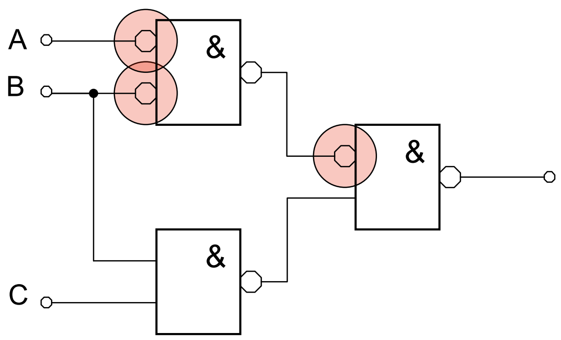
Beispiel - Transformation



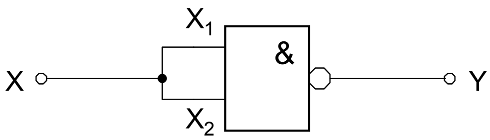
Negationen verschieben

Beispiel - Transformation

Alle drei Gatter sind NAND Gatter,
aber Negationen bei den Eingängen müssen noch ersetzt werden

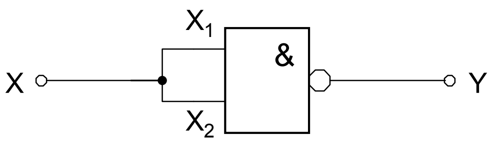


Beispiel - Transformation



X	X_1	X_2	X_1 and X_2	Y
0	0	0	0	1
1	1	1	1	0

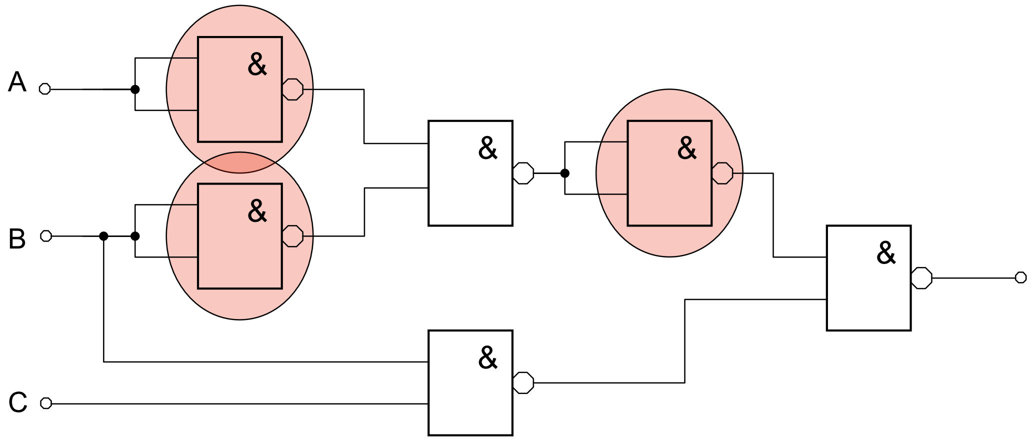
Beispiel - Transformation



X	X_1	X_2	X_1 and X_2	Y
0	0	0	0	1
1	1	1	1	0

$\rightarrow Y = \neg X$

Beispiel - Transformation



Rechenbeispiele

- Transformation
  - Ändern von vorgegebenen Schaltungen

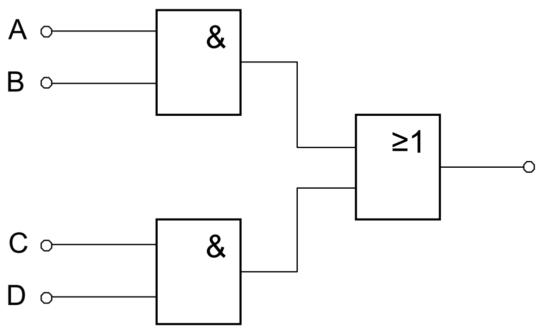
- Schaltungen
  - Boolsche Funktionen
  - Wahrheitstabellen
  - Minimierung
  - Gatter / PLA

Schaltung

- Implementieren der Funktion
- Einzelbauteile
  - AND, OR, NOT Gatter
- PLA (Programmable Logic Array)
  - Setzen der entsprechenden Verbindungen

Schaltung - Beispiel

- Realisieren der Funktion $(C \wedge D) \vee (A \wedge B)$ mit einzelnen Bauteilen



Wahrheitstabellen

- Vollständige Beschreibung des Systems
- Erstellt aus Boolscher Funktion
- Erstellt aus textueller Beschreibung des Systems

Beispiel

Angabe

Eine Schaltung mit 4 Eingängen soll entworfen werden, die genau dann 1 (true) liefert, wenn der Wert der Eingänge (als Binärzahl interpretiert) kleiner als 7 ist.

Beispiel - Schaltung

- 4 Eingänge
  - 4 Variable
  - Zahlen von 0 - 15
  - Tabelle mit 2^4 Zeilen
- Funktion
  - 1, wenn Wert ≤ 6
  - 0, wenn Wert ≥ 7

B_3	B_2	B_1	B_0	f(..)
0	0	0	0	
0	0	0	1	
0	0	1	0	
0	0	1	1	
0	1	0	0	
0	1	0	1	
0	1	1	0	
0	1	1	1	
1	0	0	0	
1	0	0	1	
1	0	1	0	
1	0	1	1	
1	1	0	0	
1	1	0	1	
1	1	1	0	
1	1	1	1	

Beispiel - Schaltung

- Funktion
  - 1, wenn Wert ≤ 6
  - 0, wenn Wert ≥ 7

B_3	B_2	B_1	B_0	Wert	f(..)
0	0	0	0	0	
0	0	0	1	1	
0	0	1	0	2	
0	0	1	1	3	
0	1	0	0	4	
0	1	0	1	5	
0	1	1	0	6	
0	1	1	1	7	
1	0	0	0	8	
1	0	0	1	9	
1	0	1	0	10	
1	0	1	1	11	
1	1	0	0	12	
1	1	0	1	13	
1	1	1	0	14	
1	1	1	1	15	

Beispiel - Schaltung

- Funktion
  – 1, wenn Wert ≤ 6

B_3	B_2	B_1	B_0	Wert	f(..)
0	0	0	0	0	**1**
0	0	0	1	1	**1**
0	0	1	0	2	**1**
0	0	1	1	3	**1**
0	1	0	0	4	**1**
0	1	0	1	5	**1**
0	1	1	0	6	**1**
0	1	1	1	7	
1	0	0	0	8	
1	0	0	1	9	
1	0	1	0	10	
1	0	1	1	11	
1	1	0	0	12	
1	1	0	1	13	
1	1	1	0	14	
1	1	1	1	15	

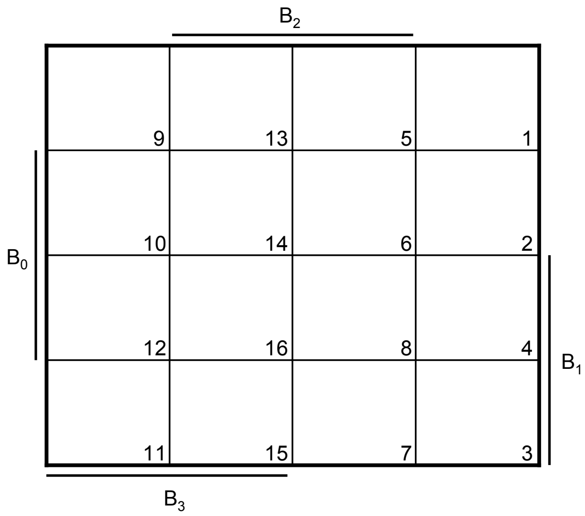
Beispiel - Schaltung

- Funktion
  – 0, wenn Wert ≥ 7

B_3	B_2	B_1	B_0	Wert	f(..)
0	0	0	0	0	1
0	0	0	1	1	1
0	0	1	0	2	1
0	0	1	1	3	1
0	1	0	0	4	1
0	1	0	1	5	1
0	1	1	0	6	1
0	1	1	1	7	**0**
1	0	0	0	8	**0**
1	0	0	1	9	**0**
1	0	1	0	10	**0**
1	0	1	1	11	**0**
1	1	0	0	12	**0**
1	1	0	1	13	**0**
1	1	1	0	14	**0**
1	1	1	1	15	**0**

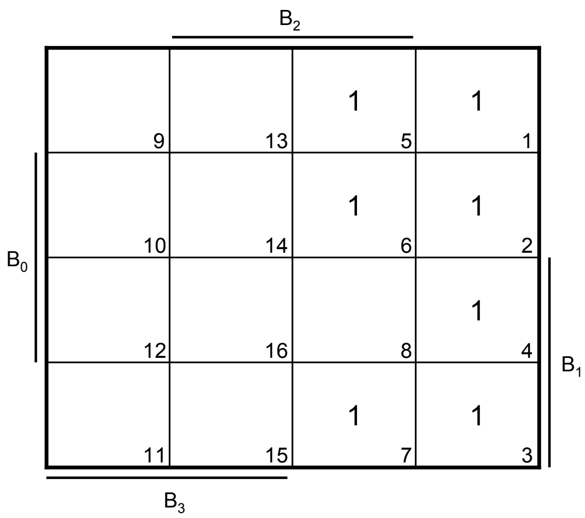
Beispiel - Schaltung

B_3	B_2	B_1	B_0	f(..)
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0



Beispiel - Schaltung

B_3	B_2	B_1	B_0	f(..)
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0



Beispiel - Schaltung

B_3	B_2	B_1	B_0	f(..)
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

	B_3	B_3, B_2	B_2	
	0 (9)	0 (13)	1 (5)	1 (1)
B_0	0 (10)	0 (14)	1 (6)	1 (2)
B_0, B_1	0 (12)	0 (16)	0 (8)	1 (4)
B_1	0 (11)	0 (15)	1 (7)	1 (3)

Beispiel - Schaltung

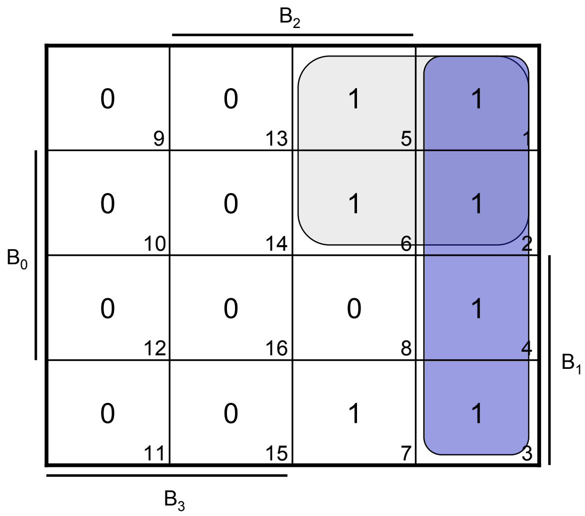
$$(\neg B_3 \wedge \neg B_1)$$

	B_3	B_3, B_2	B_2	
	0 (9)	0 (13)	1 (5)	1 (1)
B_0	0 (10)	0 (14)	1 (6)	1 (2)
B_0, B_1	0 (12)	0 (16)	0 (8)	1 (4)
B_1	0 (11)	0 (15)	1 (7)	1 (3)

Beispiel - Schaltung

$(\neg B_3 \wedge \neg B_1) \vee$

$(\neg B_3 \wedge \neg B_2)$

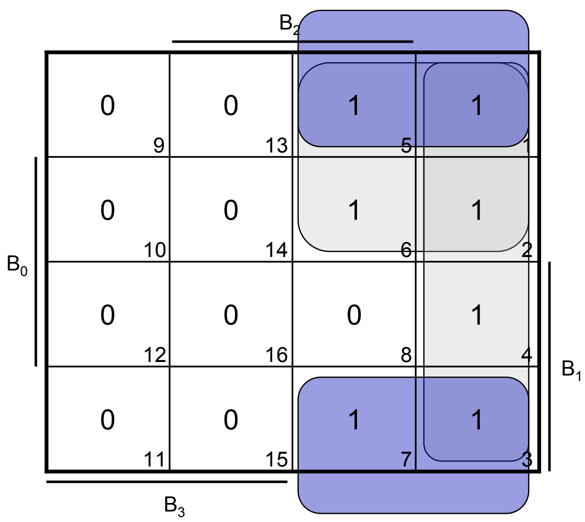


Beispiel - Schaltung

$(\neg B_3 \wedge \neg B_1) \vee$

$(\neg B_3 \wedge \neg B_2) \vee$

$(\neg B_3 \wedge \neg B_0)$

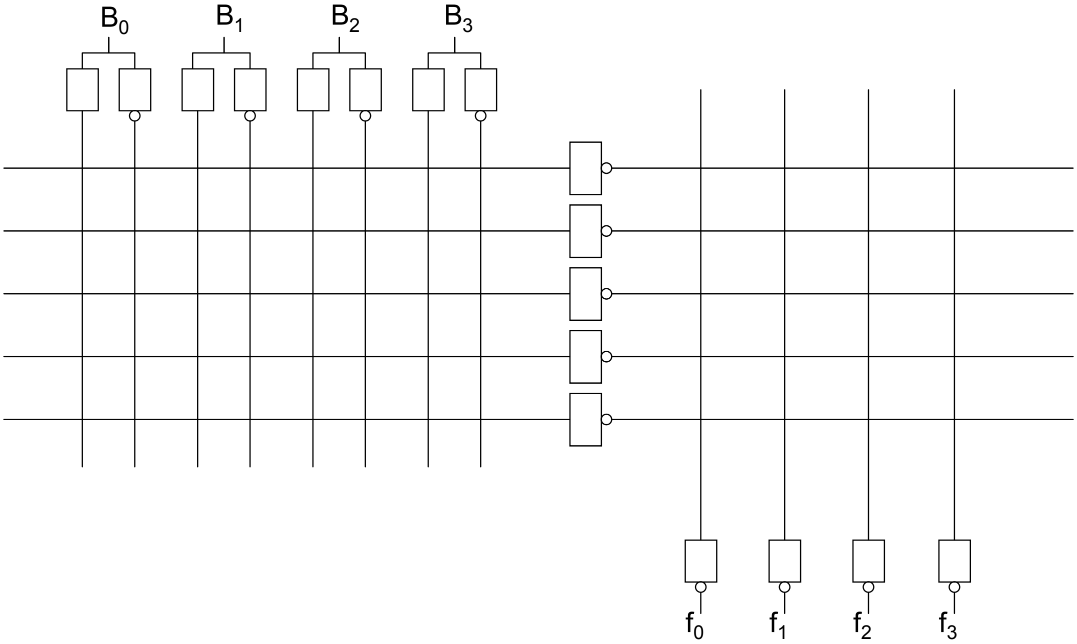


Beispiel - Schaltung

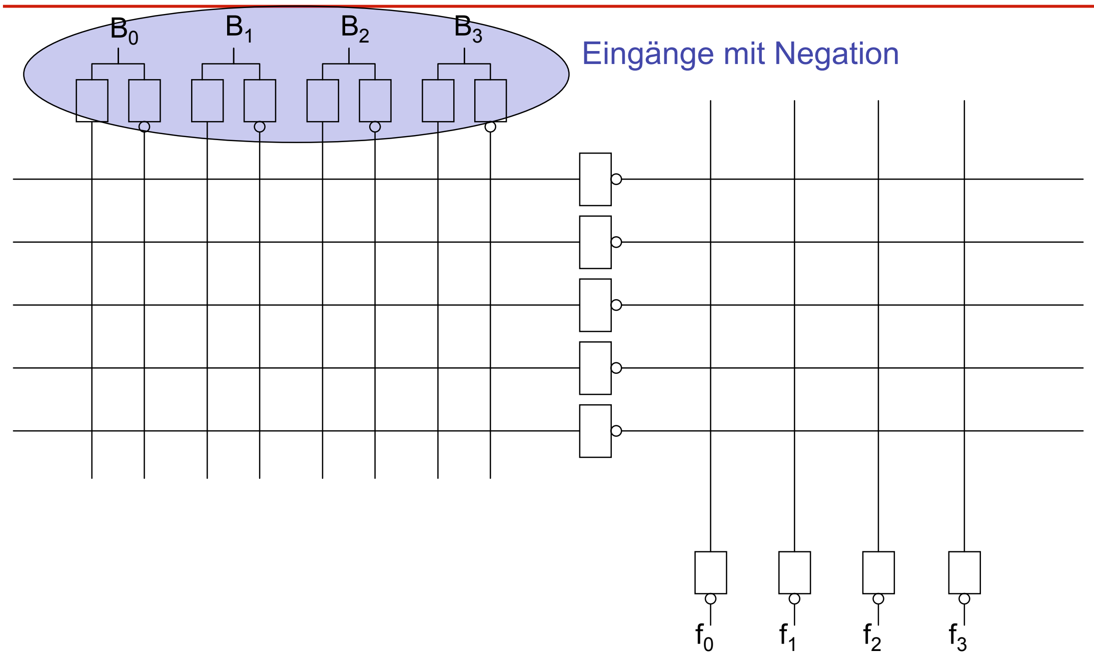
- Realisierung als PLA
  - jede Variable an einem Eingang
  - zu jeder Variable ist Negation verfügbar
  - AND Matrix und OR Matrix

  - ideal für disjunktive Normalform

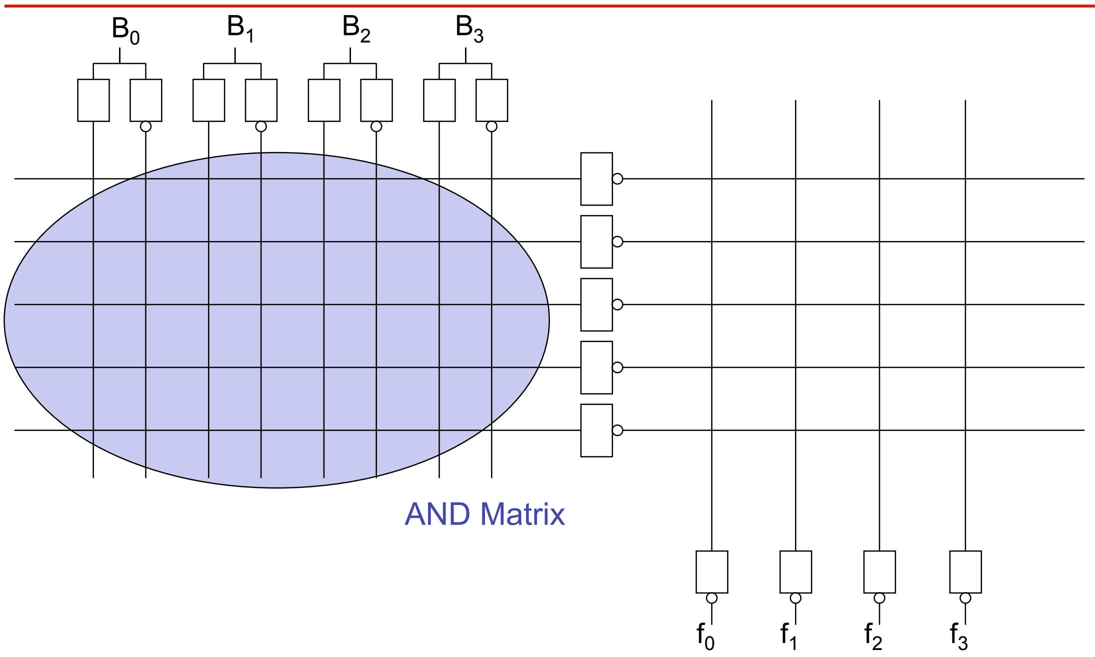
Beispiel - Schaltung



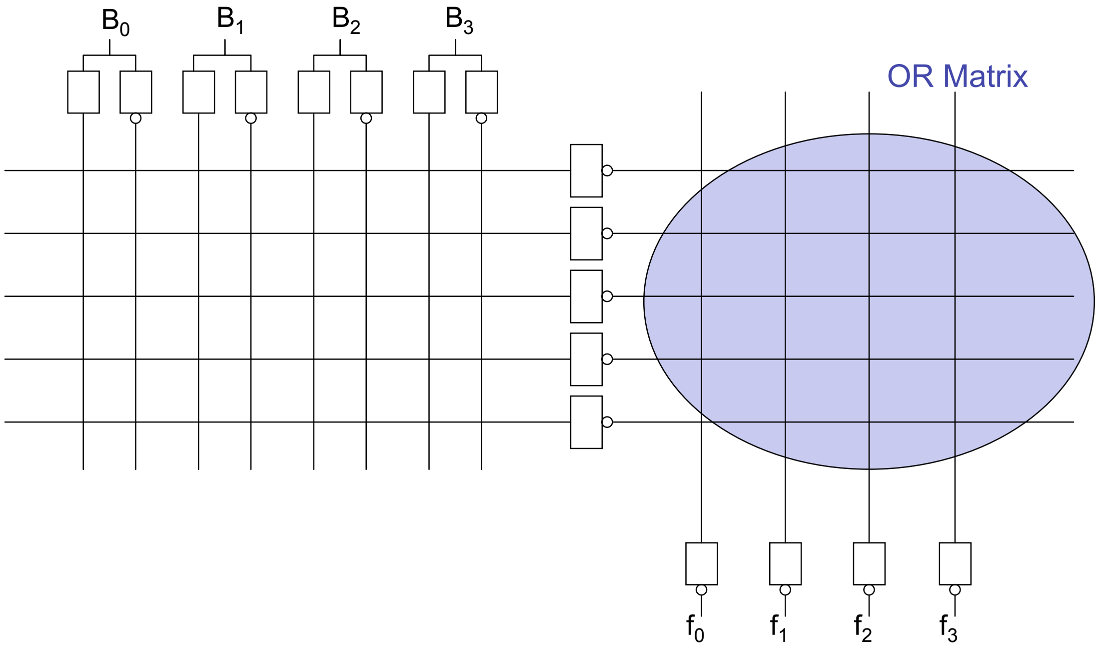
Beispiel - Schaltung



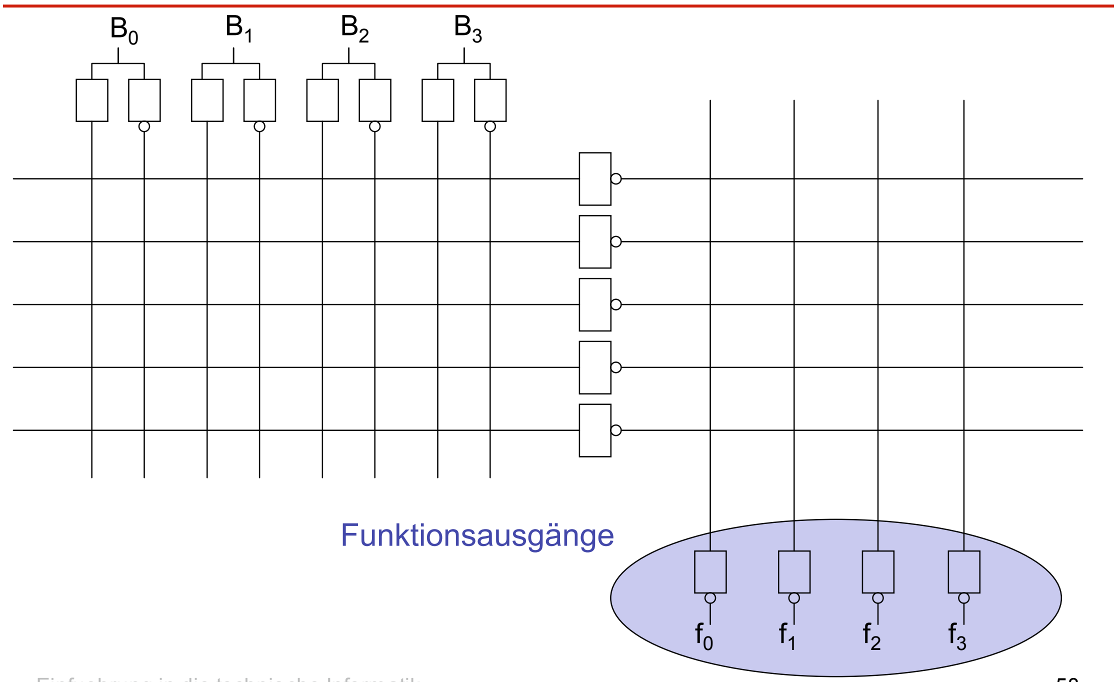
Beispiel - Schaltung



Beispiel - Schaltung

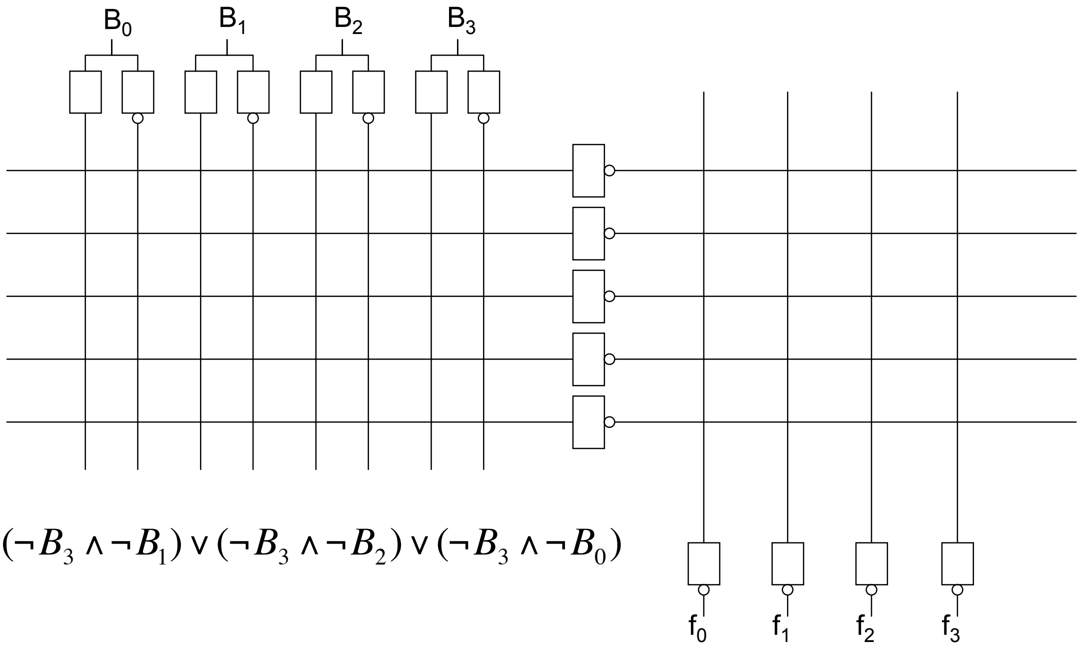


Beispiel - Schaltung



Beispiel - Schaltung

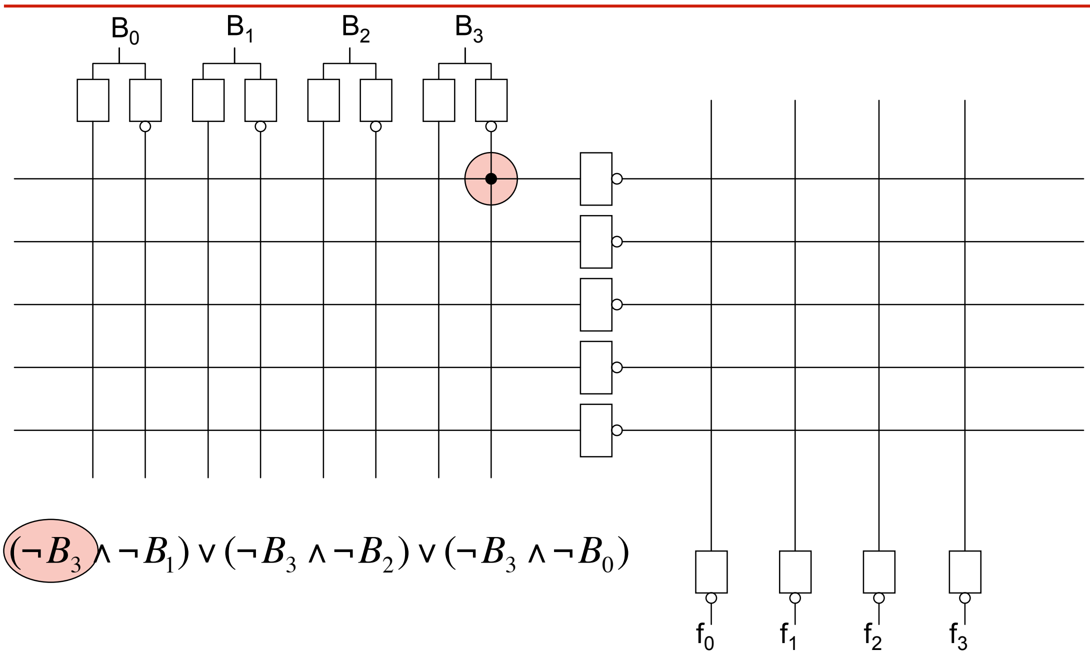
Automation Systems Group



$$(\neg B_3 \wedge \neg B_1) \vee (\neg B_3 \wedge \neg B_2) \vee (\neg B_3 \wedge \neg B_0)$$

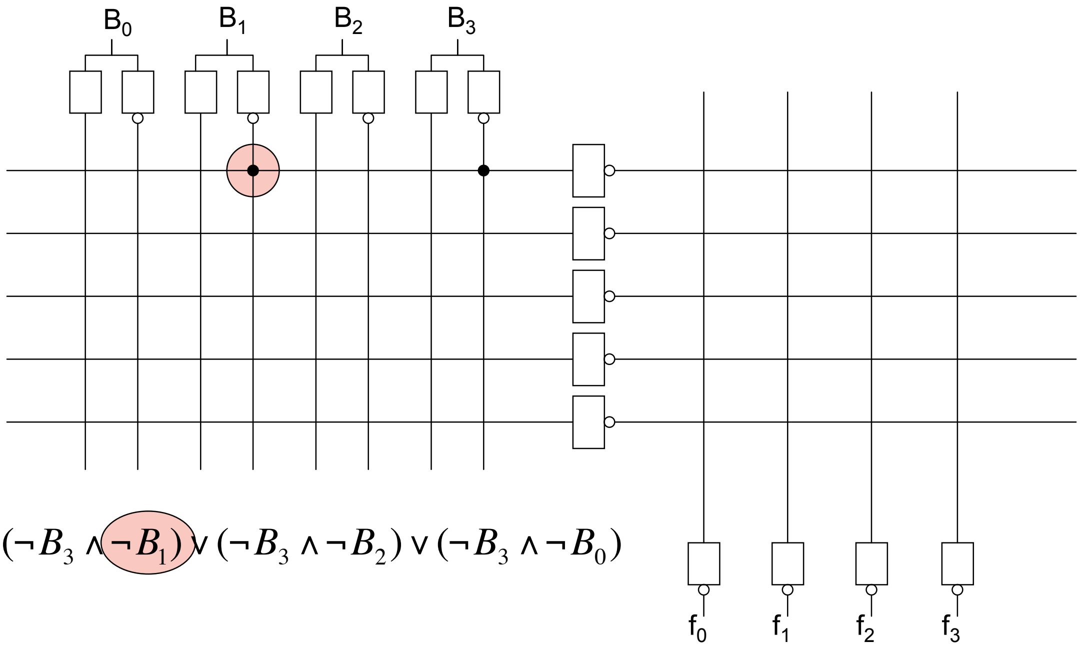
Beispiel - Schaltung

Automation Systems Group



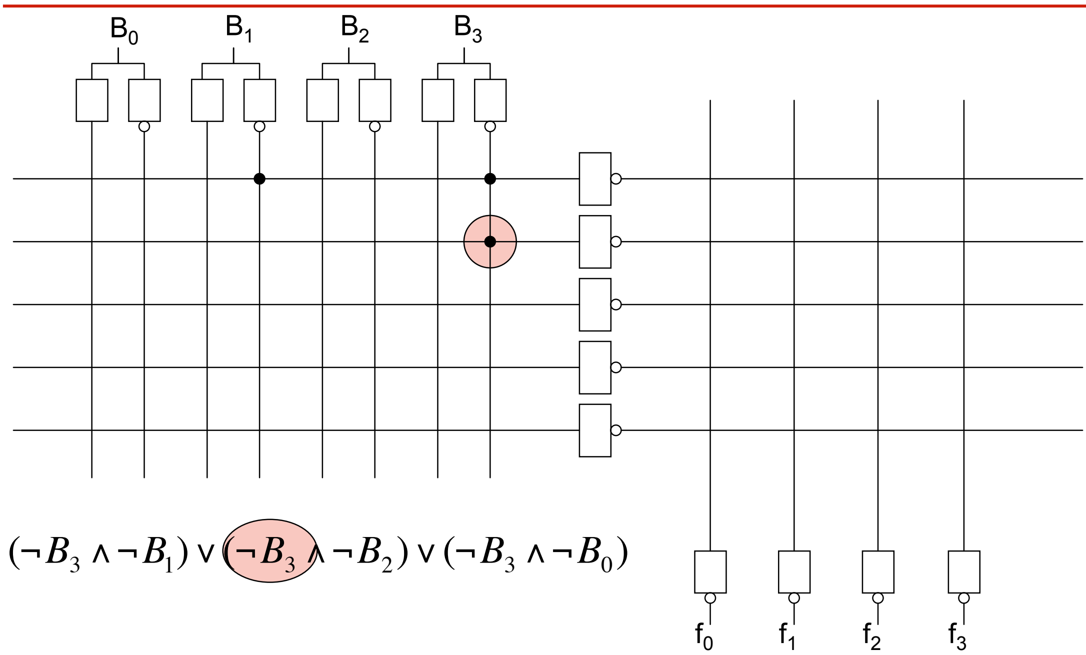
$$(\neg B_3 \wedge \neg B_1) \vee (\neg B_3 \wedge \neg B_2) \vee (\neg B_3 \wedge \neg B_0)$$

Beispiel - Schaltung



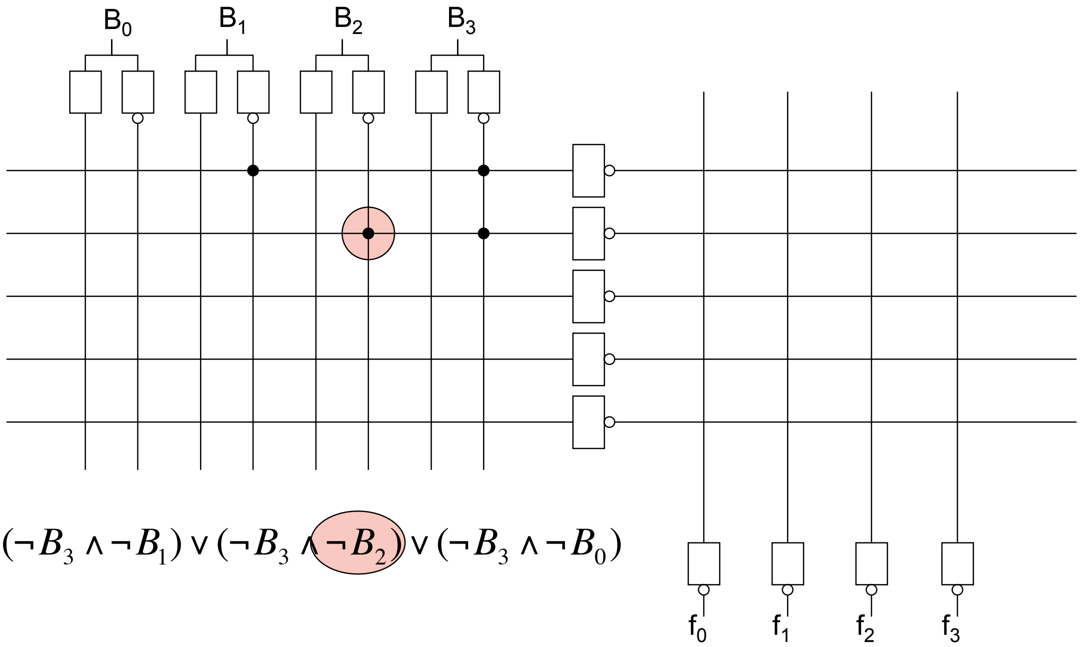
$$(\neg B_3 \wedge \neg B_1) \vee (\neg B_3 \wedge \neg B_2) \vee (\neg B_3 \wedge \neg B_0)$$

Beispiel - Schaltung



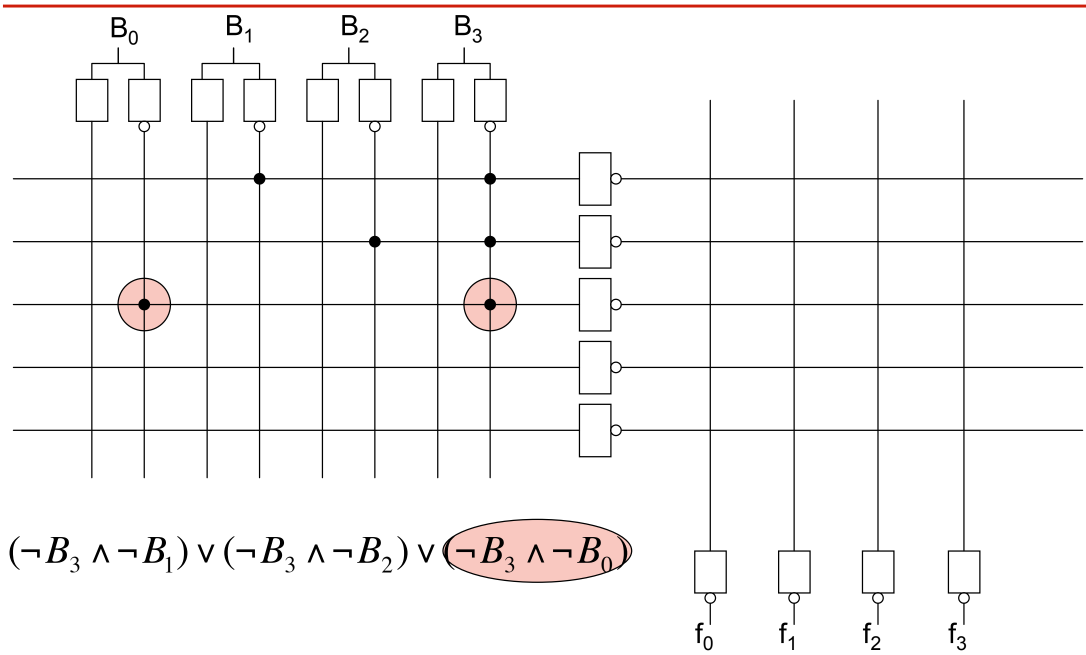
$$(\neg B_3 \wedge \neg B_1) \vee (\neg B_3 \wedge \neg B_2) \vee (\neg B_3 \wedge \neg B_0)$$

Beispiel - Schaltung



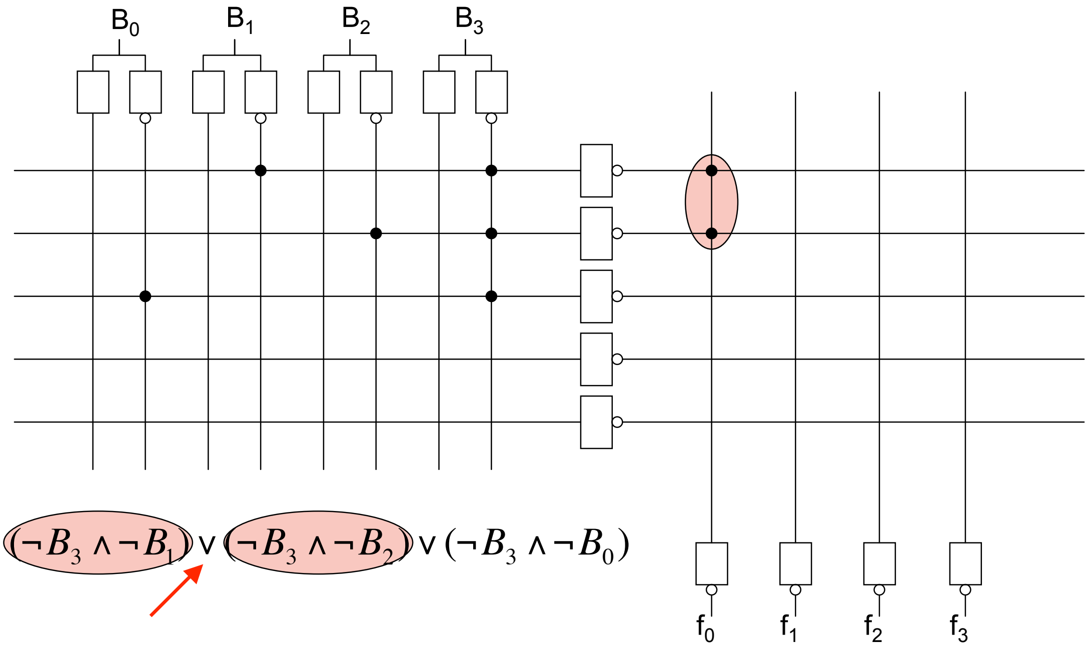
$$(\neg B_3 \wedge \neg B_1) \vee (\neg B_3 \wedge \neg B_2) \vee (\neg B_3 \wedge \neg B_0)$$

Beispiel - Schaltung



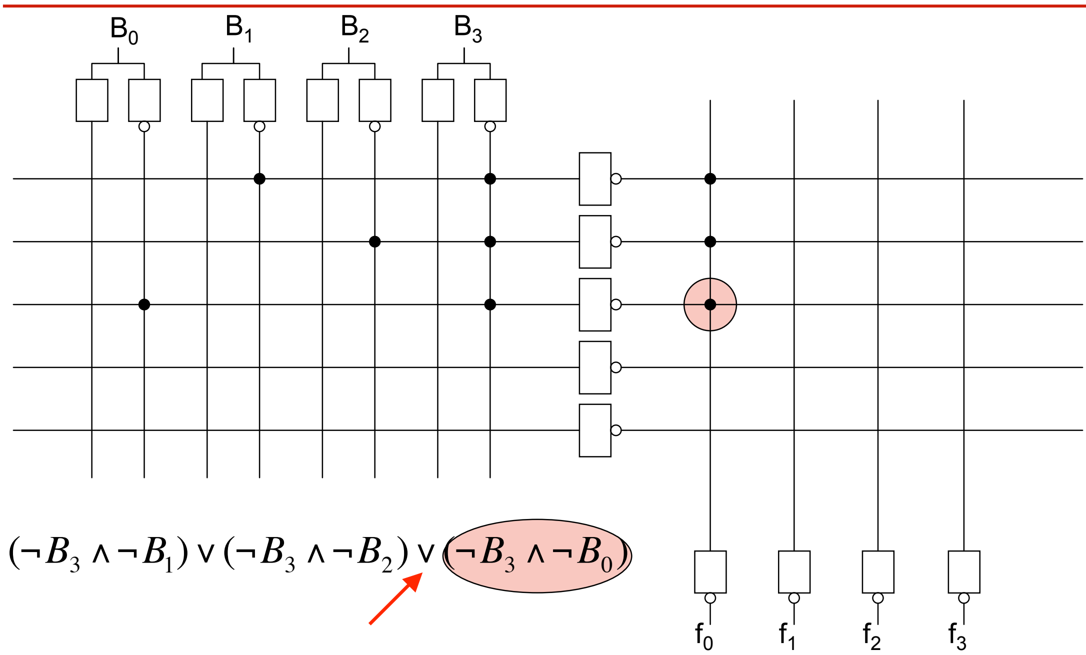
$$(\neg B_3 \wedge \neg B_1) \vee (\neg B_3 \wedge \neg B_2) \vee (\neg B_3 \wedge \neg B_0)$$

Beispiel - Schaltung



$$(\neg B_3 \wedge \neg B_1) \vee (\neg B_3 \wedge \neg B_2) \vee (\neg B_3 \wedge \neg B_0)$$

Beispiel - Schaltung



$$(\neg B_3 \wedge \neg B_1) \vee (\neg B_3 \wedge \neg B_2) \vee (\neg B_3 \wedge \neg B_0)$$

Komplexe Schaltungen

- Für manche Anwendungen sind mehrere Ausgänge erforderlich
  - diese Ausgänge können unterschiedliche Werte liefern
  - sie realisieren unterschiedliche logische Funktionen

Beispiel

Angabe

Eine Schaltung soll als PLA realisiert werden. Diese Schaltung soll zwei 2-bit Zahlen (X und Y) vergleichen und am Ausgang A_1 genau dann den Wert 1 liefern, wenn X kleiner als Y ist. Zusätzlich soll am Ausgang A_2 genau dann den Wert 1 ausgegeben werden, wenn X gleich 0 und Y ungleich 0 ist.

Beispiel - Schaltung

- 2 x 2-bit Zahlen
  - 4 Eingänge
  - Tabelle mit 2^4 Zeilen
- 2 Funktionen
  - A_1 : X < Y
  - A_2 : X == 0 && Y != 0

X_1	X_0	Y_1	Y_0	A_1	A_2
0	0	0	0		
0	0	0	1		
0	0	1	0		
0	0	1	1		
0	1	0	0		
0	1	0	1		
0	1	1	0		
0	1	1	1		
1	0	0	0		
1	0	0	1		
1	0	1	0		
1	0	1	1		
1	1	0	0		
1	1	0	1		
1	1	1	0		
1	1	1	1		

Beispiel - Schaltung

Automation Systems Group

- A_1 : X < Y

X = 0

X_1	X_0	Y_1	Y_0	A_1	A_2
0	0	0	0	**0**	
0	0	0	1	**1**	
0	0	1	0	**1**	
0	0	1	1	**1**	
0	1	0	0		
0	1	0	1		
0	1	1	0		
0	1	1	1		
1	0	0	0		
1	0	0	1		
1	0	1	0		
1	0	1	1		
1	1	0	0		
1	1	0	1		
1	1	1	0		
1	1	1	1		

Beispiel - Schaltung

- A_1 : X < Y

X = 1 (rows 5–8)

X_1	X_0	Y_1	Y_0	A_1	A_2
0	0	0	0	0	
0	0	0	1	1	
0	0	1	0	1	
0	0	1	1	1	
0	1	0	0	0	
0	1	0	1	0	
0	1	1	0	1	
0	1	1	1	1	
1	0	0	0		
1	0	0	1		
1	0	1	0		
1	0	1	1		
1	1	0	0		
1	1	0	1		
1	1	1	0		
1	1	1	1		

Beispiel - Schaltung

Automation Systems Group

- A_1 : X < Y

X = 2 (rows 9–12)

X_1	X_0	Y_1	Y_0	A_1	A_2
0	0	0	0	0	
0	0	0	1	1	
0	0	1	0	1	
0	0	1	1	1	
0	1	0	0	0	
0	1	0	1	0	
0	1	1	0	1	
0	1	1	1	1	
1	0	0	0	0	
1	0	0	1	0	
1	0	1	0	0	
1	0	1	1	1	
1	1	0	0		
1	1	0	1		
1	1	1	0		
1	1	1	1		

Beispiel - Schaltung

- A_1 : $X < Y$

X_1	X_0	Y_1	Y_0	A_1	A_2
0	0	0	0	0	
0	0	0	1	1	
0	0	1	0	1	
0	0	1	1	1	
0	1	0	0	0	
0	1	0	1	0	
0	1	1	0	1	
0	1	1	1	1	
1	0	0	0	0	
1	0	0	1	0	
1	0	1	0	0	
1	0	1	1	1	
1	1	0	0	**0**	
1	1	0	1	**0**	
1	1	1	0	**0**	
1	1	1	1	**0**	

X = 3

Beispiel - Schaltung

- A_2 : X == 0 && Y != 0

X_1	X_0	Y_1	Y_0	A_1	A_2
0	0	0	0	0	
0	0	0	1	1	
0	0	1	0	1	
0	0	1	1	1	
0	1	0	0	0	
0	1	0	1	0	
0	1	1	0	1	
0	1	1	1	1	
1	0	0	0	0	
1	0	0	1	0	
1	0	1	0	0	
1	0	1	1	1	
1	1	0	0	0	
1	1	0	1	0	
1	1	1	0	0	
1	1	1	1	0	

Beispiel - Schaltung

- A_2 : X == 0 && Y != 0

X = 0

X_1	X_0	Y_1	Y_0	A_1	A_2
0	0	0	0	0	
0	0	0	1	1	
0	0	1	0	1	
0	0	1	1	1	
0	1	0	0	0	0
0	1	0	1	0	0
0	1	1	0	1	0
0	1	1	1	1	0
1	0	0	0	0	0
1	0	0	1	0	0
1	0	1	0	0	0
1	0	1	1	1	0
1	1	0	0	0	0
1	1	0	1	0	0
1	1	1	0	0	0
1	1	1	1	0	0

Beispiel - Schaltung

Automation Systems Group

- A_2 : X == 0 && Y != 0

X = 0

X_1	X_0	Y_1	Y_0	A_1	A_2
0	0	0	0	0	0
0	0	0	1	1	
0	0	1	0	1	
0	0	1	1	1	
0	1	0	0	0	0
0	1	0	1	0	0
0	1	1	0	1	0
0	1	1	1	1	0
1	0	0	0	0	0
1	0	0	1	0	0
1	0	1	0	0	0
1	0	1	1	1	0
1	1	0	0	0	0
1	1	0	1	0	0
1	1	1	0	0	0
1	1	1	1	0	0

Beispiel - Schaltung

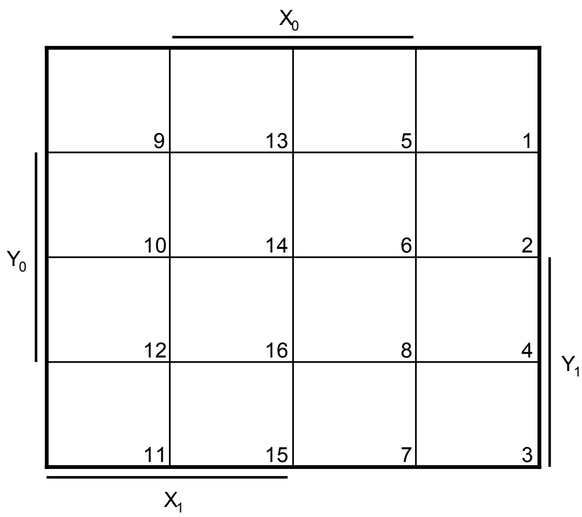
- A_2 : X == 0 && Y != 0

X = 0

X_1	X_0	Y_1	Y_0	A_1	A_2
0	0	0	0	0	0
0	0	0	1	1	1
0	0	1	0	1	1
0	0	1	1	1	1
0	1	0	0	0	0
0	1	0	1	0	0
0	1	1	0	1	0
0	1	1	1	1	0
1	0	0	0	0	0
1	0	0	1	0	0
1	0	1	0	0	0
1	0	1	1	1	0
1	1	0	0	0	0
1	1	0	1	0	0
1	1	1	0	0	0
1	1	1	1	0	0

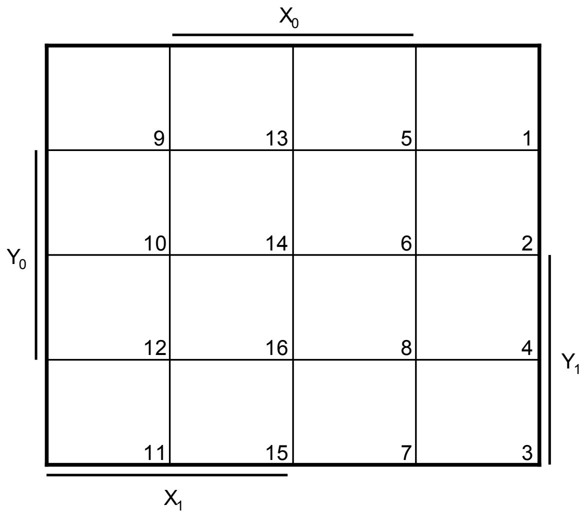
Beispiel - Schaltung

X_1	X_0	Y_1	Y_0	A_1	A_2
0	0	0	0	0	0
0	0	0	1	1	1
0	0	1	0	1	1
0	0	1	1	1	1
0	1	0	0	0	0
0	1	0	1	0	0
0	1	1	0	1	0
0	1	1	1	1	0
1	0	0	0	0	0
1	0	0	1	0	0
1	0	1	0	0	0
1	0	1	1	1	0
1	1	0	0	0	0
1	1	0	1	0	0
1	1	1	0	0	0
1	1	1	1	0	0



Beispiel - Schaltung

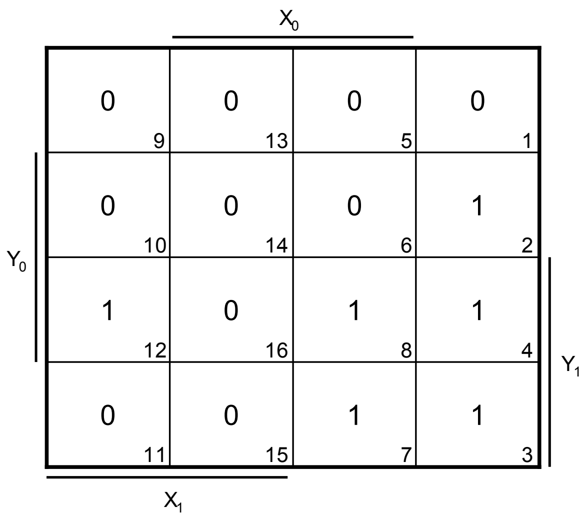
X_1	X_0	Y_1	Y_0	A_1	A_2
0	0	0	0	0	0
0	0	0	1	1	1
0	0	1	0	1	1
0	0	1	1	1	1
0	1	0	0	0	0
0	1	0	1	0	0
0	1	1	0	1	0
0	1	1	1	1	0
1	0	0	0	0	0
1	0	0	1	0	0
1	0	1	0	0	0
1	0	1	1	1	0
1	1	0	0	0	0
1	1	0	1	0	0
1	1	1	0	0	0
1	1	1	1	0	0



Beispiel - Schaltung

Automation Systems Group

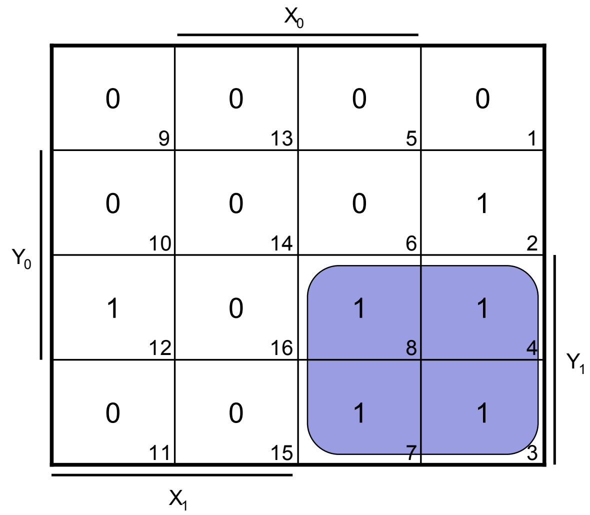
X_1	X_0	Y_1	Y_0	A_1	A_2
0	0	0	0	0	0
0	0	0	1	1	1
0	0	1	0	1	1
0	0	1	1	1	1
0	1	0	0	0	0
0	1	0	1	0	0
0	1	1	0	1	0
0	1	1	1	1	0
1	0	0	0	0	0
1	0	0	1	0	0
1	0	1	0	0	0
1	0	1	1	1	0
1	1	0	0	0	0
1	1	0	1	0	0
1	1	1	0	0	0
1	1	1	1	0	0



Beispiel - Schaltung

Ausgang A_1

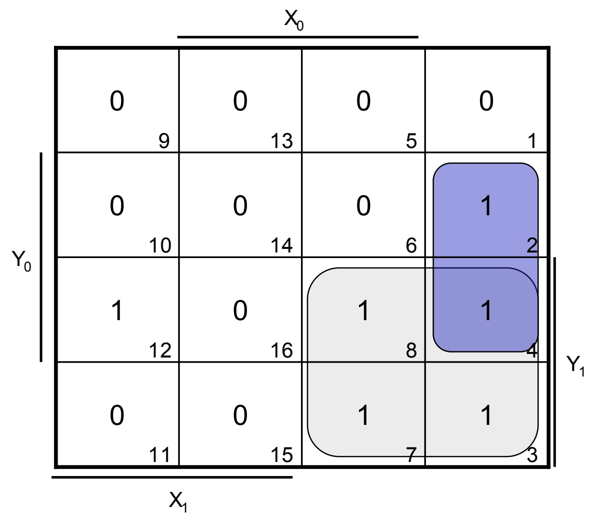
$$(\neg X_1 \wedge Y_1)$$



Beispiel - Schaltung

Ausgang A_1

$$(\neg X_1 \wedge Y_1) \vee$$
$$(\neg X_1 \wedge \neg X_0 \wedge Y_0)$$

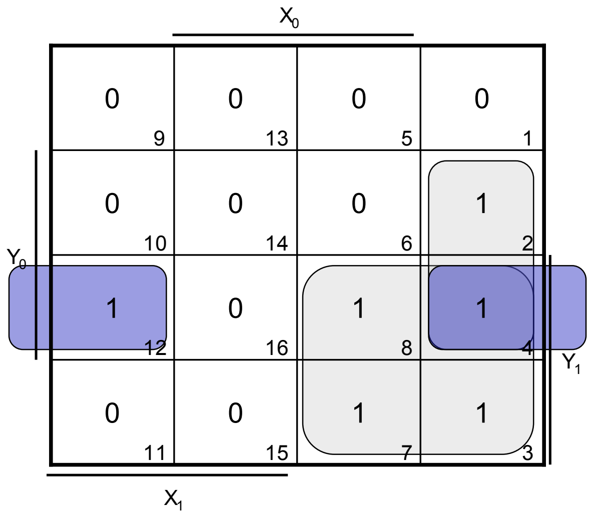


Beispiel - Schaltung

Automation Systems Group

Ausgang A_1

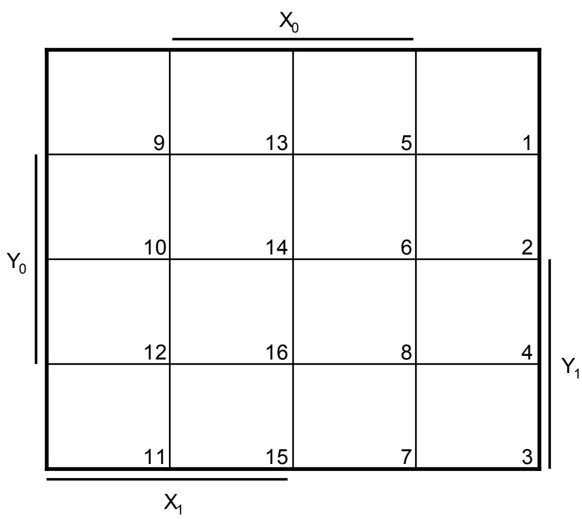
$$(\neg X_1 \wedge Y_1) \vee$$
$$(\neg X_1 \wedge \neg X_0 \wedge Y_0) \vee$$
$$(\neg X_0 \wedge Y_1 \wedge Y_0)$$



Beispiel - Schaltung

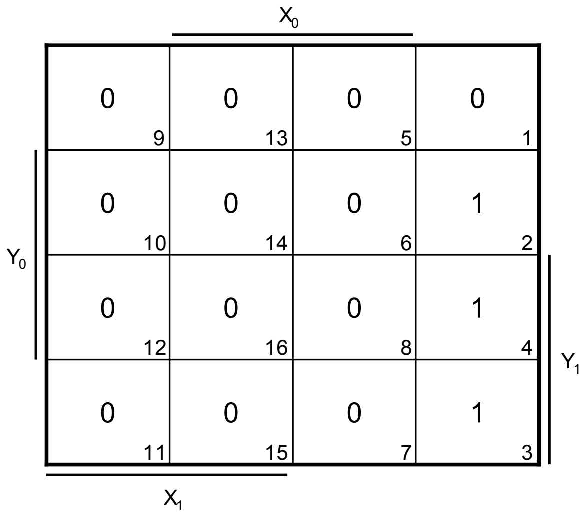
Automation Systems Group

X_1	X_0	Y_1	Y_0	A_1	A_2
0	0	0	0	0	0
0	0	0	1	1	1
0	0	1	0	1	1
0	0	1	1	1	1
0	1	0	0	0	0
0	1	0	1	0	0
0	1	1	0	1	0
0	1	1	1	1	0
1	0	0	0	0	0
1	0	0	1	0	0
1	0	1	0	0	0
1	0	1	1	1	0
1	1	0	0	0	0
1	1	0	1	0	0
1	1	1	0	0	0
1	1	1	1	0	0



Beispiel - Schaltung

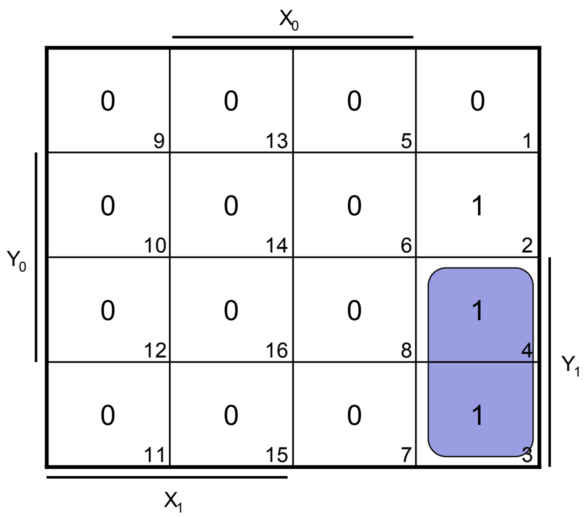
X_1	X_0	Y_1	Y_0	A_1	A_2
0	0	0	0	0	0
0	0	0	1	1	1
0	0	1	0	1	1
0	0	1	1	1	1
0	1	0	0	0	0
0	1	0	1	0	0
0	1	1	0	1	0
0	1	1	1	1	0
1	0	0	0	0	0
1	0	0	1	0	0
1	0	1	0	0	0
1	0	1	1	1	0
1	1	0	0	0	0
1	1	0	1	0	0
1	1	1	0	0	0
1	1	1	1	0	0



Beispiel - Schaltung

Ausgang A_2

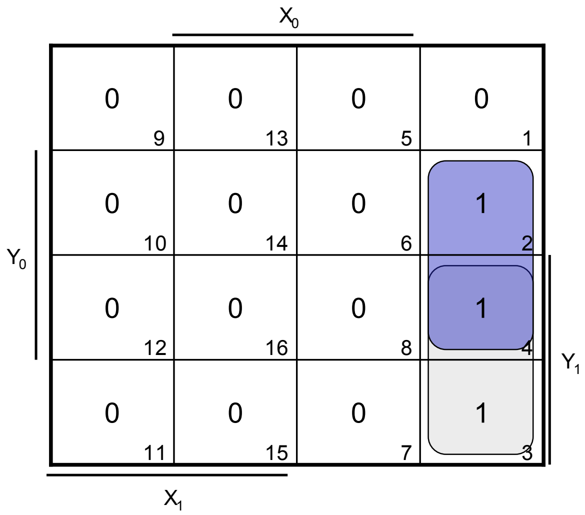
$$(\neg X_1 \wedge \neg X_0 \wedge Y_1)$$



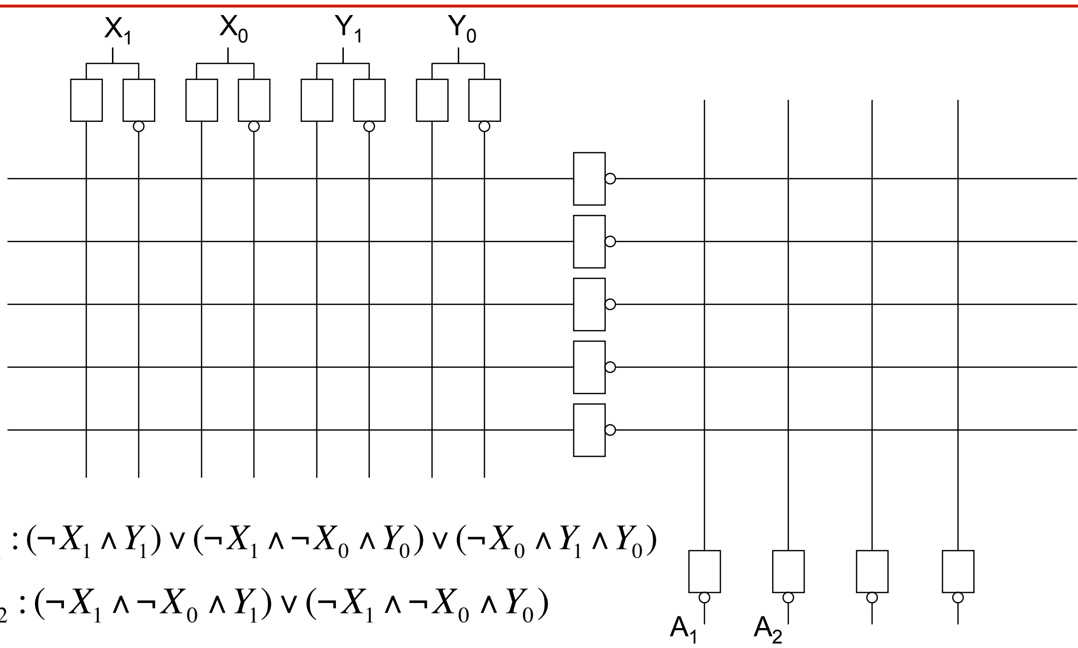
Beispiel - Schaltung

Ausgang A_2

$$(\neg X_1 \wedge \neg X_0 \wedge Y_1) \vee$$
$$(\neg X_1 \wedge \neg X_0 \wedge Y_0)$$

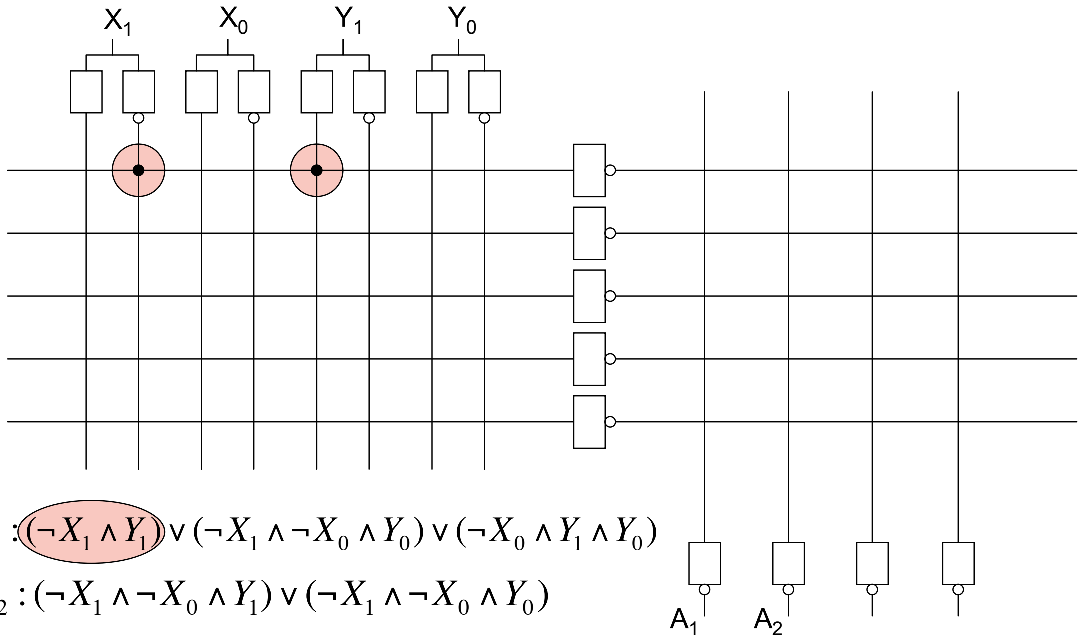


Beispiel - Schaltung



$$A_1 : (\neg X_1 \wedge Y_1) \vee (\neg X_1 \wedge \neg X_0 \wedge Y_0) \vee (\neg X_0 \wedge Y_1 \wedge Y_0)$$
$$A_2 : (\neg X_1 \wedge \neg X_0 \wedge Y_1) \vee (\neg X_1 \wedge \neg X_0 \wedge Y_0)$$

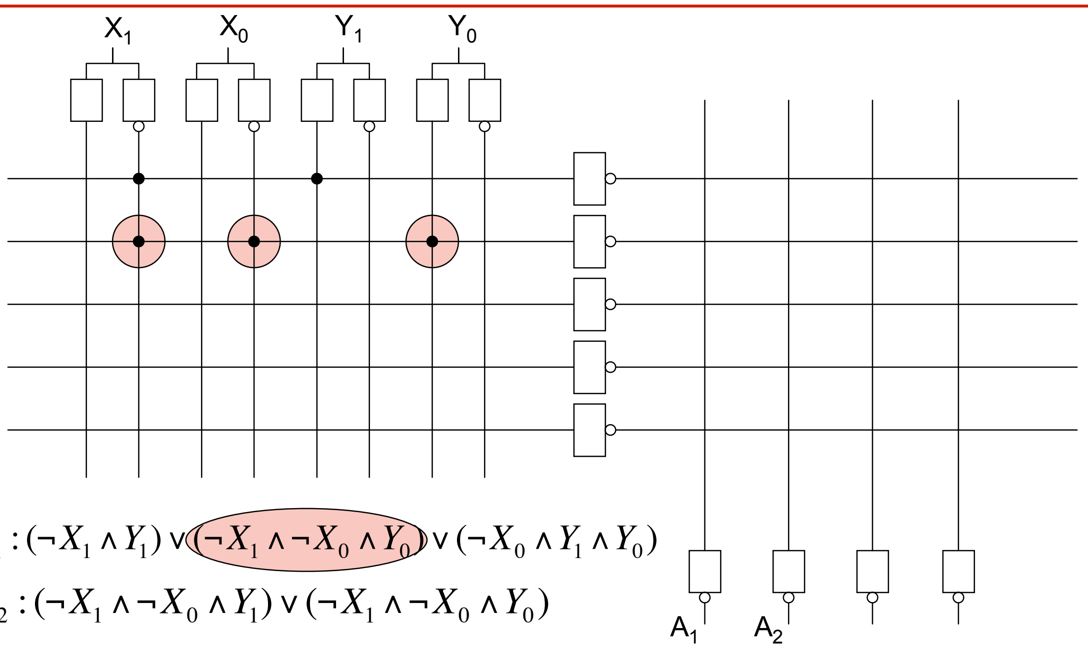
Beispiel - Schaltung



$A_1 : (\neg X_1 \wedge Y_1) \vee (\neg X_1 \wedge \neg X_0 \wedge Y_0) \vee (\neg X_0 \wedge Y_1 \wedge Y_0)$

$A_2 : (\neg X_1 \wedge \neg X_0 \wedge Y_1) \vee (\neg X_1 \wedge \neg X_0 \wedge Y_0)$

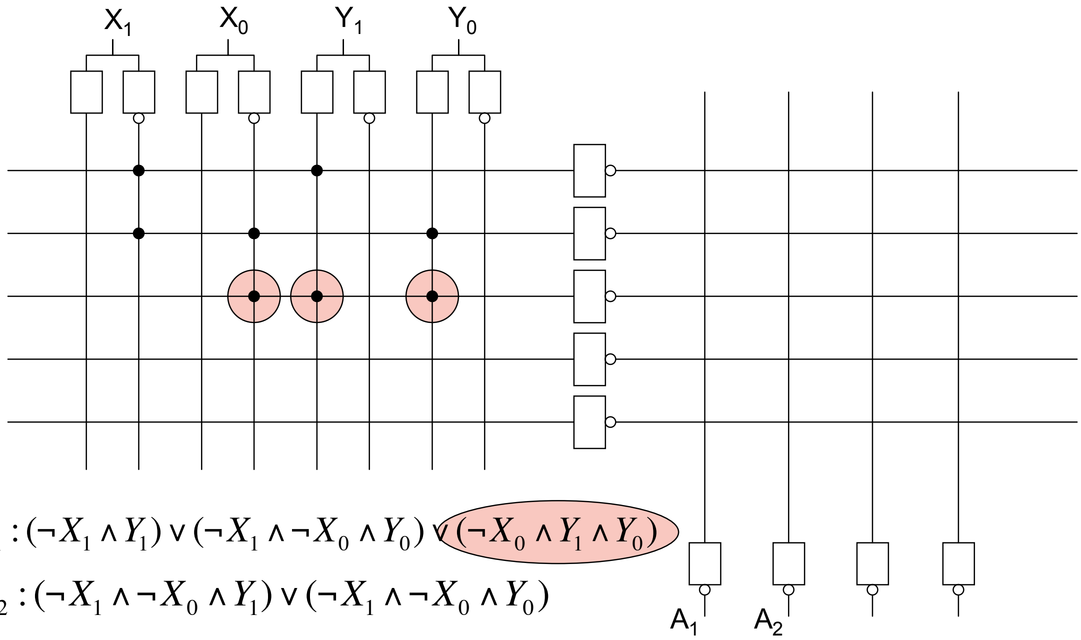
Beispiel - Schaltung



$A_1 : (\neg X_1 \wedge Y_1) \vee (\neg X_1 \wedge \neg X_0 \wedge Y_0) \vee (\neg X_0 \wedge Y_1 \wedge Y_0)$

$A_2 : (\neg X_1 \wedge \neg X_0 \wedge Y_1) \vee (\neg X_1 \wedge \neg X_0 \wedge Y_0)$

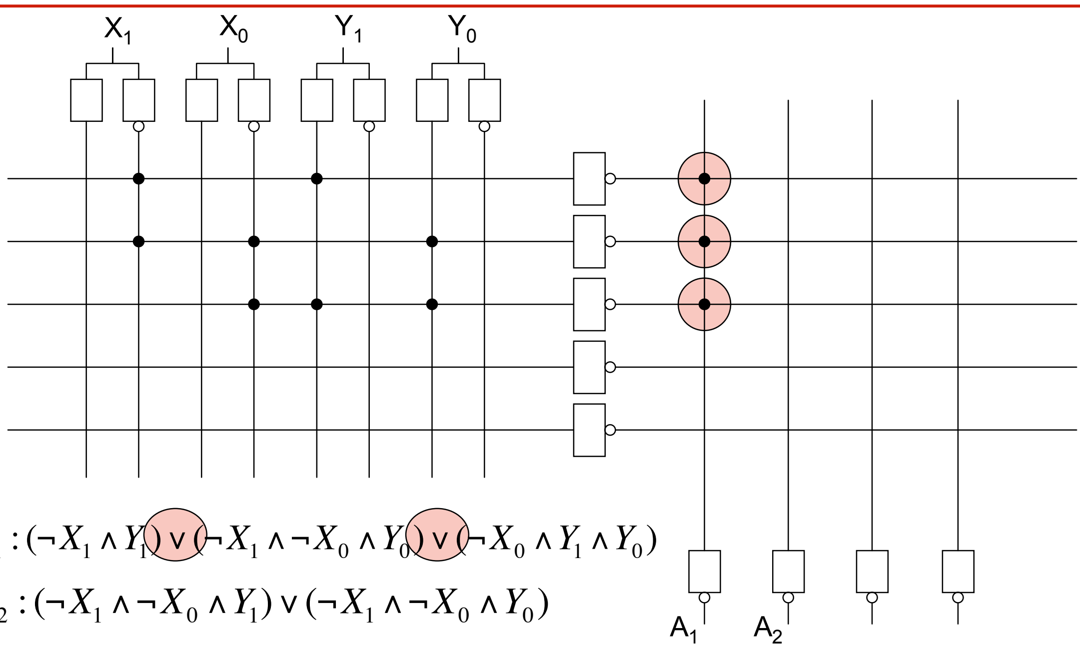
Beispiel - Schaltung



$A_1 : (\neg X_1 \wedge Y_1) \vee (\neg X_1 \wedge \neg X_0 \wedge Y_0) \vee (\neg X_0 \wedge Y_1 \wedge Y_0)$

$A_2 : (\neg X_1 \wedge \neg X_0 \wedge Y_1) \vee (\neg X_1 \wedge \neg X_0 \wedge Y_0)$

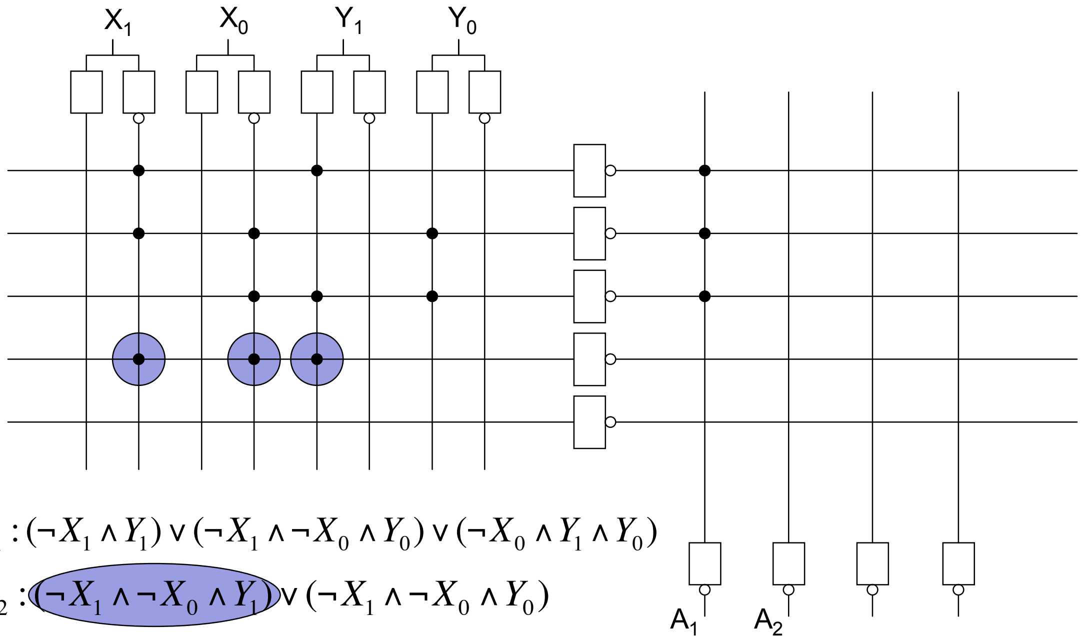
Beispiel - Schaltung



$A_1 : (\neg X_1 \wedge Y_1) \vee (\neg X_1 \wedge \neg X_0 \wedge Y_0) \vee (\neg X_0 \wedge Y_1 \wedge Y_0)$

$A_2 : (\neg X_1 \wedge \neg X_0 \wedge Y_1) \vee (\neg X_1 \wedge \neg X_0 \wedge Y_0)$

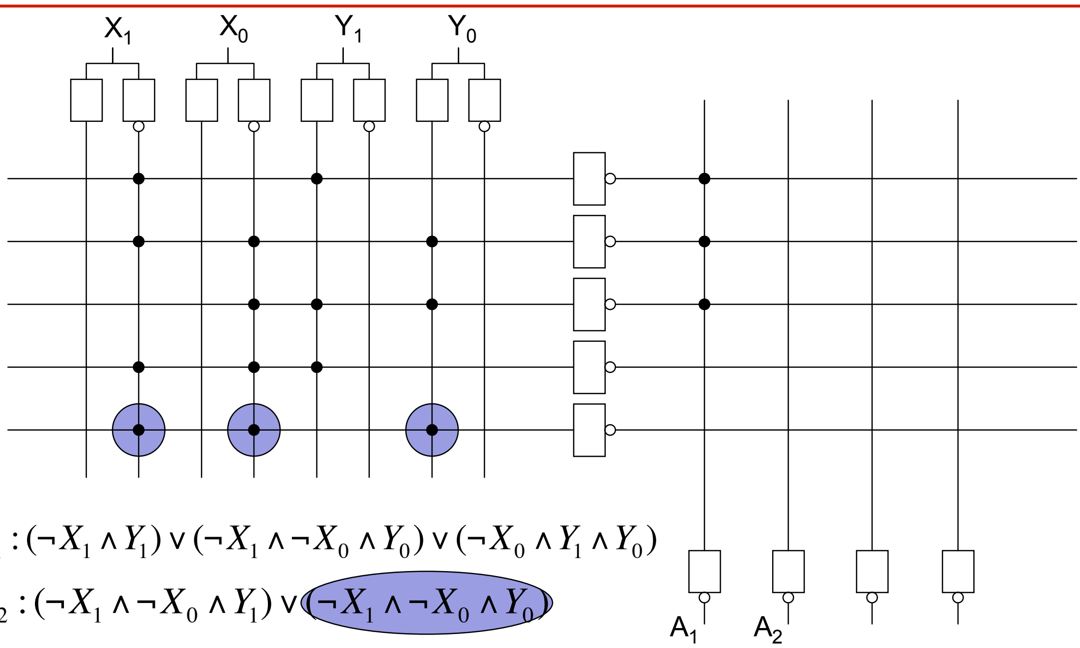
Beispiel - Schaltung



$$A_1 : (\neg X_1 \wedge Y_1) \vee (\neg X_1 \wedge \neg X_0 \wedge Y_0) \vee (\neg X_0 \wedge Y_1 \wedge Y_0)$$

$$A_2 : (\neg X_1 \wedge \neg X_0 \wedge Y_1) \vee (\neg X_1 \wedge \neg X_0 \wedge Y_0)$$

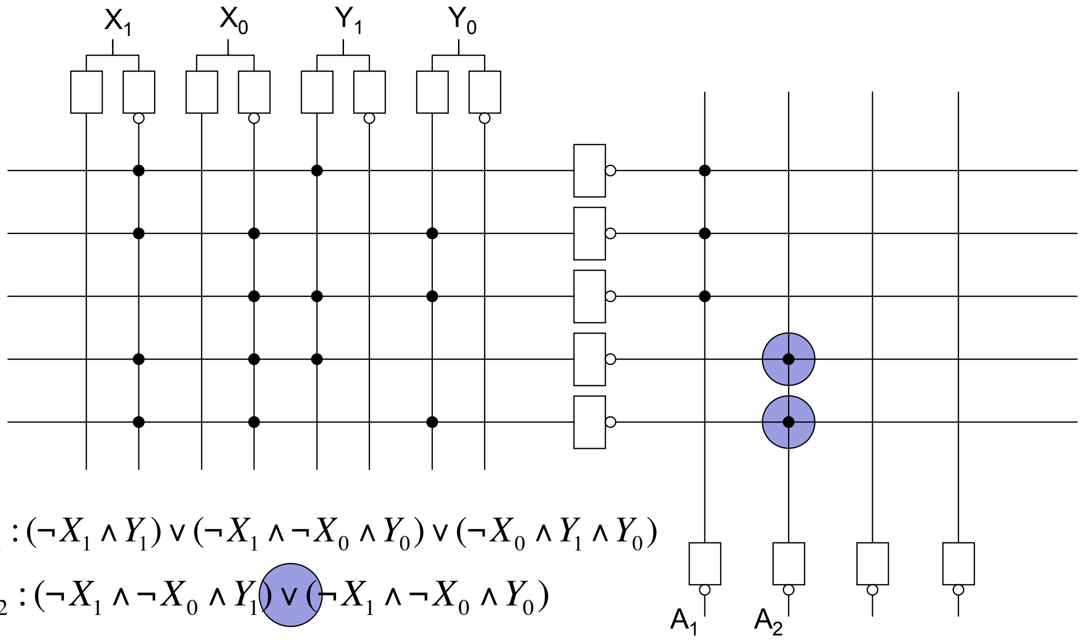
Beispiel - Schaltung



$$A_1 : (\neg X_1 \wedge Y_1) \vee (\neg X_1 \wedge \neg X_0 \wedge Y_0) \vee (\neg X_0 \wedge Y_1 \wedge Y_0)$$

$$A_2 : (\neg X_1 \wedge \neg X_0 \wedge Y_1) \vee (\neg X_1 \wedge \neg X_0 \wedge Y_0)$$

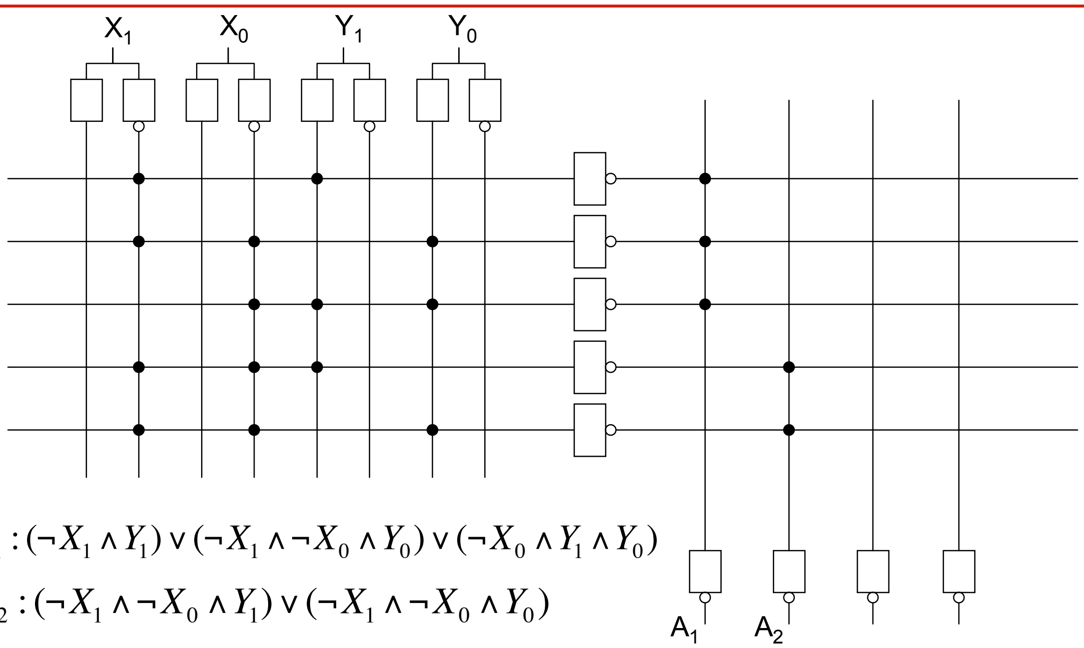
Beispiel - Schaltung



$$A_1 : (\neg X_1 \wedge Y_1) \vee (\neg X_1 \wedge \neg X_0 \wedge Y_0) \vee (\neg X_0 \wedge Y_1 \wedge Y_0)$$

$$A_2 : (\neg X_1 \wedge \neg X_0 \wedge Y_1) \vee (\neg X_1 \wedge \neg X_0 \wedge Y_0)$$

Beispiel - Schaltung



$$A_1 : (\neg X_1 \wedge Y_1) \vee (\neg X_1 \wedge \neg X_0 \wedge Y_0) \vee (\neg X_0 \wedge Y_1 \wedge Y_0)$$

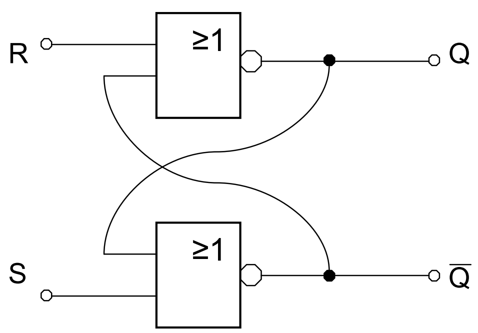
$$A_2 : (\neg X_1 \wedge \neg X_0 \wedge Y_1) \vee (\neg X_1 \wedge \neg X_0 \wedge Y_0)$$

Sequentielle Schaltungen

- Bisher einfache Schaltungen
  - Ausgänge hängen immer direkt von Eingängen ab
  - keine Speicherung von Information
  - kein interner Zustand
- Sequentielle (komplexe) Schaltungen
  - Ausgänge hängen von Eingängen *und* aktuellem Zustand ab
  - Zustand wird in Speicherelementen gehalten
  - Beispiel
    - Zähler
    - Speicherregister

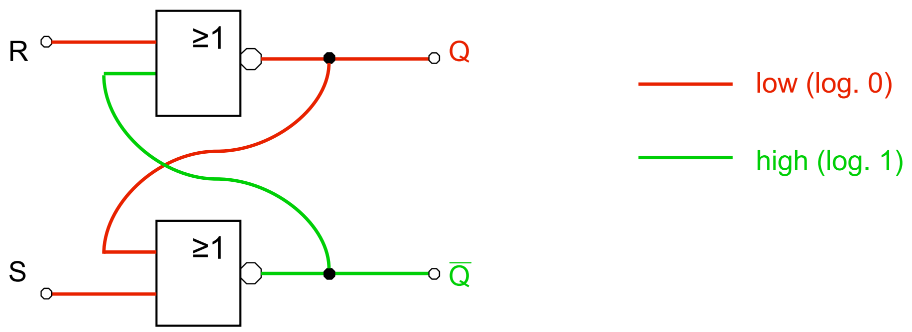
Sequentielle Schaltungen

- Speicherelemente
  - Flip-Flops / Latches
  - arbeiten mit Rückkoppelung



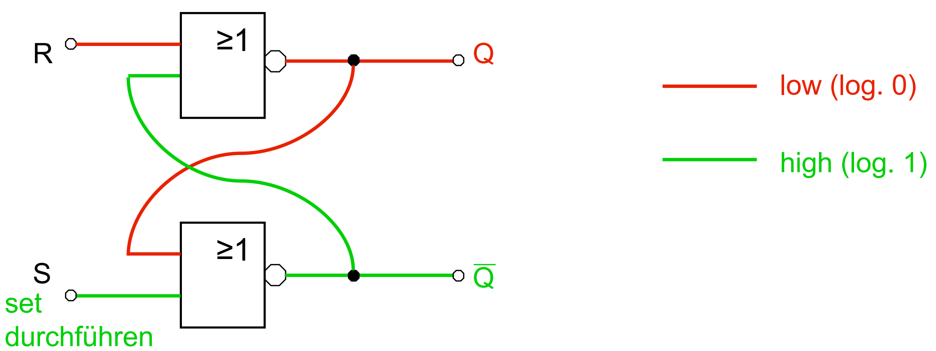
Sequentielle Schaltungen

- Speicherelemente
  - Flip-Flops / Latches
  - arbeiten mit Rückkoppelung



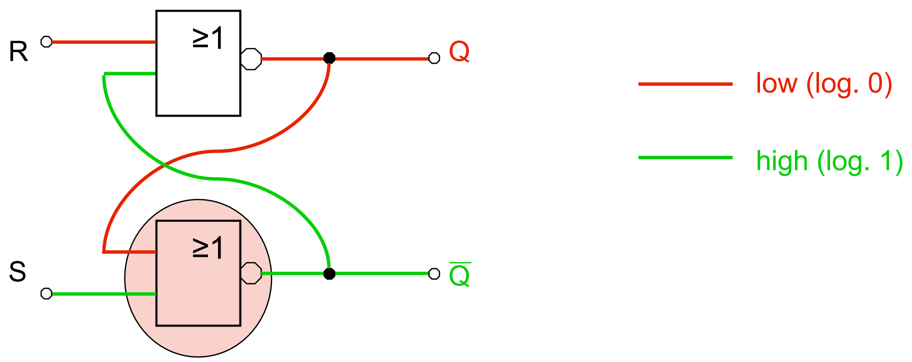
Sequentielle Schaltungen

- Speicherelemente
  - Flip-Flops / Latches
  - arbeiten mit Rückkoppelung



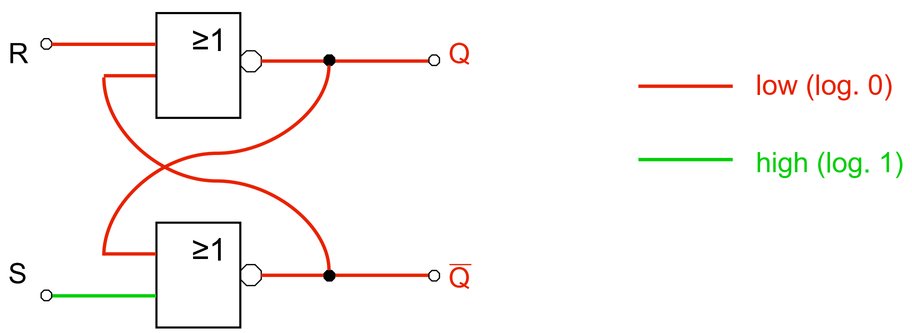
Sequentielle Schaltungen

- Speicherelemente
  - Flip-Flops / Latches
  - arbeiten mit Rückkoppelung



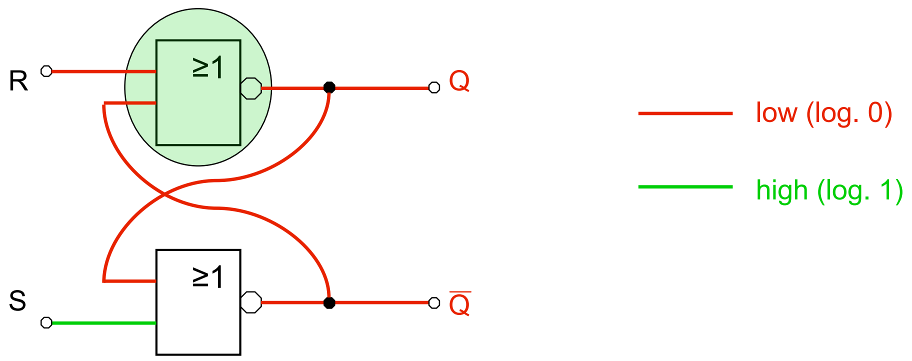
Sequentielle Schaltungen

- Speicherelemente
  - Flip-Flops / Latches
  - arbeiten mit Rückkoppelung



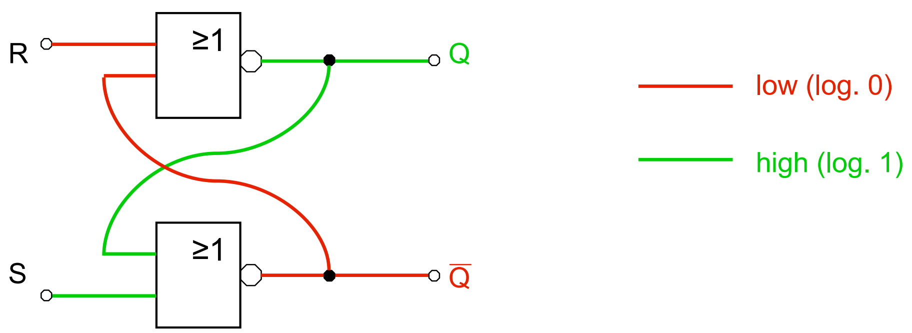
Sequentielle Schaltungen

- Speicherelemente
  - Flip-Flops / Latches
  - arbeiten mit Rückkoppelung



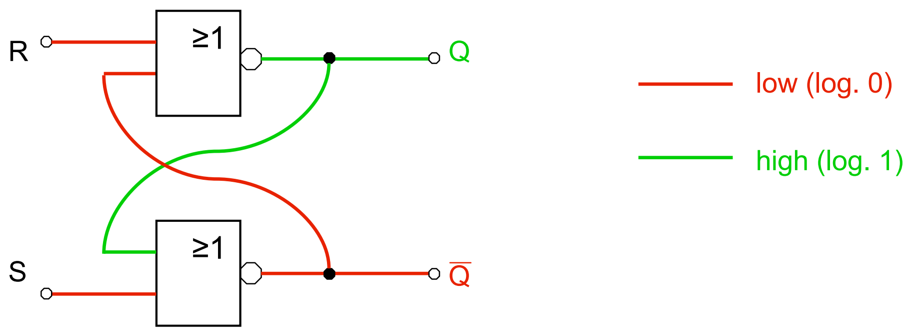
Sequentielle Schaltungen

- Speicherelemente
  - Flip-Flops / Latches
  - arbeiten mit Rückkoppelung



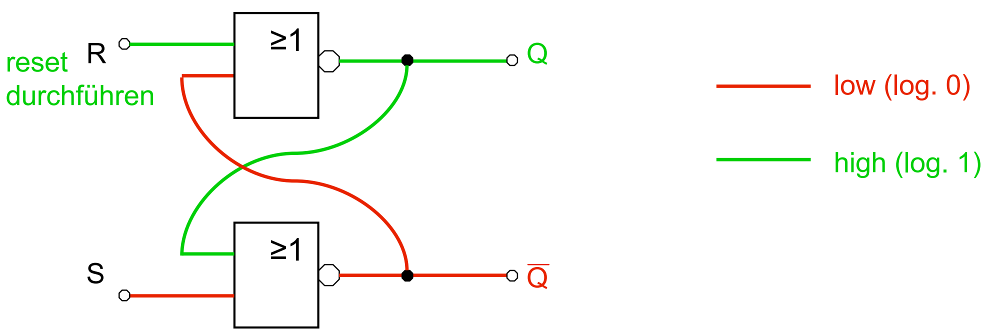
Sequentielle Schaltungen

- Speicherelemente
  - Flip-Flops / Latches
  - arbeiten mit Rückkoppelung



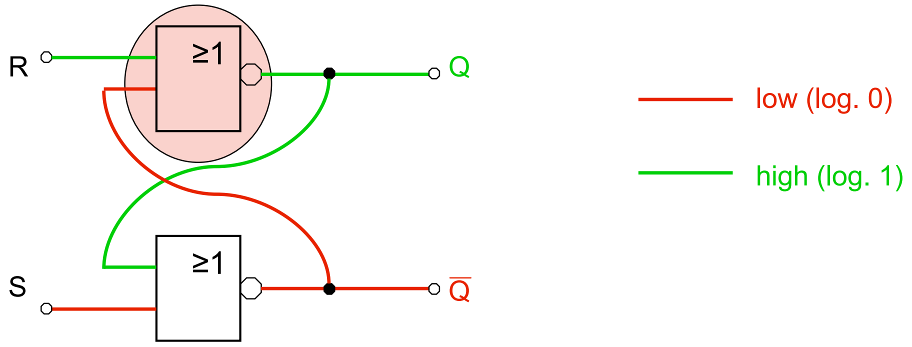
Sequentielle Schaltungen

- Speicherelemente
  - Flip-Flops / Latches
  - arbeiten mit Rückkoppelung



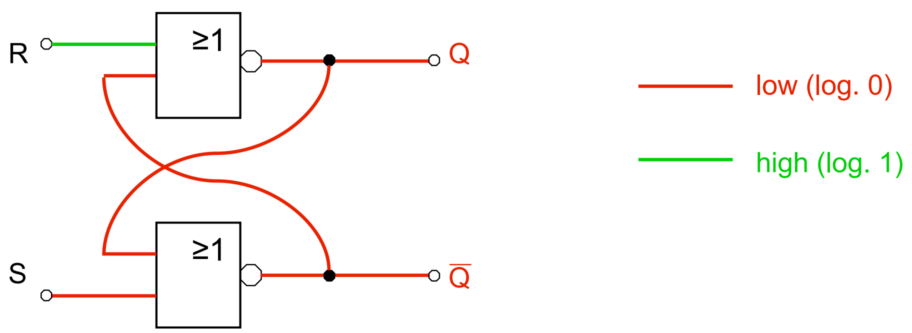
Sequentielle Schaltungen

- Speicherelemente
  - Flip-Flops / Latches
  - arbeiten mit Rückkoppelung



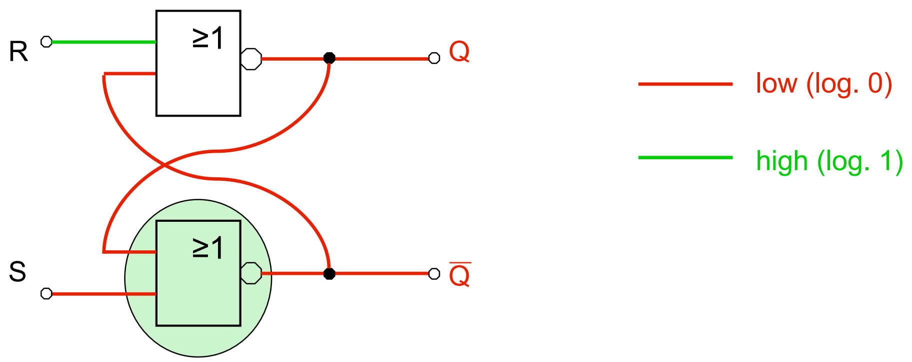
Sequentielle Schaltungen

- Speicherelemente
  - Flip-Flops / Latches
  - arbeiten mit Rückkoppelung



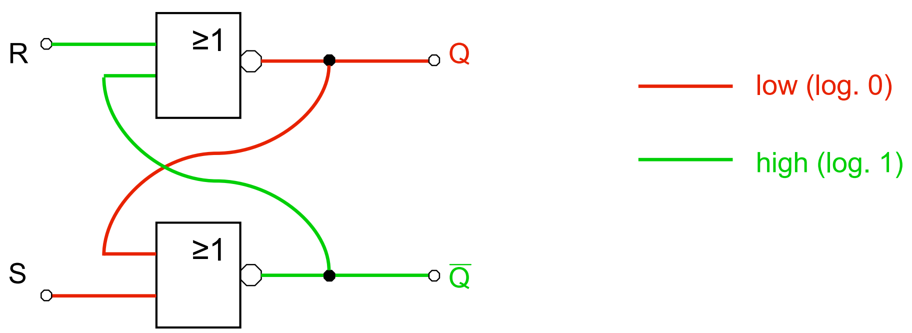
Sequentielle Schaltungen

- Speicherelemente
  - Flip-Flops / Latches
  - arbeiten mit Rückkoppelung



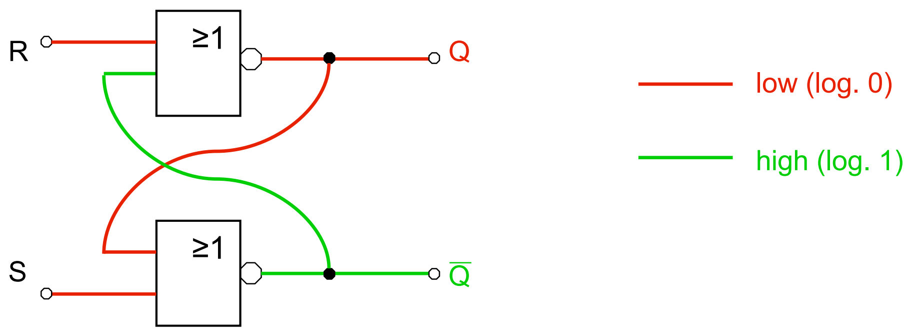
Sequentielle Schaltungen

- Speicherelemente
  - Flip-Flops / Latches
  - arbeiten mit Rückkoppelung



Sequentielle Schaltungen

- Speicherelemente
  - Flip-Flops / Latches
  - arbeiten mit Rückkoppelung



Sequentielle Schaltungen

- Typen von Speicherelementen
  - RS-Flip-Flop
    - auf vorigen Folien beschrieben
    - undefinierter Zustand wenn R = S = 1
  - D-Latch
    - nur ein Eingang
    - verhindert undefinierten Zustand des RS-Flip-Flops
  - JK-Flip-Flop
    - ähnlich wie RS-Flip-Flop
    - wenn J = K = 1, dann schaltet der Ausgang des Flip-Flop in den anderen logischen Zustand (aus low wird high und umgekehrt)

Zusammenfassung

- Logische Schaltungen
  - Boolsche Funktionen
  - Wahrheitstabellen
- Minimierung
  - minimale disjunktive Normalform
  - KV Diagramme
- Implementierung
  - Gatter
  - PLA
- Sequentielle Schaltungen