GTI - Grundlagen der Technischen Informatik
3. Normalformen

Thomas Studer
Institut für Informatik
Universität Bern

UNIVERSITÄT BERN

Inhalt

Repetition:
Addition von zwei einstelligen Binärzahlen

Extensionale vs. Intensionale Darstellung von Funktionen
Bsp: natürliche Zahlen

Extensionale Darstellung als Wertetabelle

x	f(x)
0	0
1	2
2	4
3	6
4	8
5	10

Intensionale Darstellung als arithmetischer Ausdruck ist nicht eindeutig

$f(x) = x + x$

oder

$f(x) = 2x$

Extensionale vs. Intensionale Darstellung von Funktionen Bsp: Boolesche Funktionen

Extensionale Darstellung als Wertetabelle

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

Intensionale Darstellung als Boolescher Ausdruck ist nicht eindeutig

f(x,y) = (x * not(y)) + (not(x) * y)

oder

f(x,y) = (not(x)+not(y)) * (x+y)

Bestimme die Funktion

> Wie findet man die Funktion zu einer gegebenen Wahrheitstafel?

i	x1	x2	x3	f(x1,x2,x3)
0	0	0	0	0
1	0	0	1	0
2	0	1	0	0
3	0	1	1	1
4	1	0	0	0
5	1	0	1	1
6	1	1	0	0
7	1	1	1	1

Minterm

- i sei Index von $f: B^n \rightarrow B$ und $(i_1,\ldots,i_n)$ die Dualdarstellung von i
- Die Funktion $m_i(x_1,\ldots,x_n) := x_1^{i_1} \cdot x_2^{i_2} \cdot\ldots\cdot x_n^{i_n}$ heisst dann **i-ter Minterm**.
  Dabei sei $x_k^{i_k} := x_k$ falls $i_k = 1$ und $x_k^{i_k} := \neg x_k$ falls $i_k = 0$.

- Beispiele:
  - $m_3(x_1,x_2,x_3) = \neg x_1 \cdot x_2 \cdot x_3$ 011
  - $m_4(x_1,x_2,x_3) = x_1 \cdot \neg x_2 \cdot \neg x_3$ 100

- Minterm m_i nimmt genau dann den Wert 1 an, wenn das Argument $(x_1,\ldots,x_n)$ die Dualdarstellung von i liefert.

Minterm, Beispiel

$m_i\,(x_1,x_2,x_3) = 1$ g.d.w. $(x_1,x_2,x_3)_2 = (\ i\ )_{10}$

$m_3\,(x_1,x_2,x_3) = 1$ g.d.w. $(x_1,x_2,x_3)_2 = (\ 0,1,1\ )_2 = (\ 3\ )_{10}$

Einschlägiger Index

- Boolesche Funktion f: $B^3 \rightarrow B$ gegeben durch

i	**x1**	**x2**	**x3**	**f(x1,x2,x3)**
0	0	0	0	0
1	0	0	1	0
2	0	1	0	0
3	0	1	1	1
4	1	0	0	0
5	1	0	1	1
6	1	1	0	0
7	1	1	1	1

- **Minterme**, z.B. $m_3(x_1,x_2,x_3) = \neg x_1 x_2 x_3$, $m_4(x_1,x_2,x_3) = x_1 \neg x_2 \neg x$
- **i** heisst **einschlägiger Index** zu f, falls $f(i_1,\ldots,i_n)=1$
- Beispiel: einschlägige Indizes der obigen Funktion f sind 3, 5, 7

Darstellungssatz für Boolesche Funktionen

> Jede Boolesche Funktion f: $B^n \rightarrow B$ ist eindeutig darstellbar als **Summe der Minterme ihrer einschlägigen Indizes**, d.h.

ist $I \subseteq \{0, ... , 2^n-1\}$ die Menge der einschlägigen Indizes von f, so gilt

$$\sum_{i \in I} m_i$$

und keine andere Minterm-Summe stellt f dar.

> Diese Darstellung heisst auch

kanonische disjunktive Normalform (DNF).

Beispiel: DNF

i	x1	x2	x3	$m_3 = \neg x_1 x_2 x_3$	$m_5 = x_1 \neg x_2 x_3$	$m_7 = x_1 x_2 x_3$	f(x1,x2,x3)
0	0	0	0	0	0	0	0
1	0	0	1	0	0	0	0
2	0	1	0	0	0	0	0
3	0	1	1	1	0	0	1
4	1	0	0	0	0	0	0
5	1	0	1	0	1	0	1
6	1	1	0	0	0	0	0
7	1	1	1	0	0	1	1

> $f(x_1,x_2,x_3) = m_3 + m_5 + m_7 = \neg x_1 x_2 x_3 + x_1 \neg x_2 x_3 + x_1 x_2 x_3$

Darstellungssatz für Boolesche Funktionen

- Beweis Darstellungssatz
  - Zu zeigen:
    - Existenz
    - Eindeutigkeit

Folgerungen

- Jede n-stellige Boolesche Funktion ist mittels der zweistelligen Funktionen $+$ und $\cdot$ sowie der einstelligen Funktion $\neg$ darstellbar.
- Eine Menge S von Booleschen Funktionen heisst **funktional vollständig**, wenn sich jede Boolesche Funktion allein durch Komposition von Funktionen aus S darstellen lässt.

Folgerungen II

- $\{+, \cdot, \neg\}$ ist funktional vollständig
- $\{\cdot, \neg\}$ ist funktional vollständig
- $\{+, \neg\}$ ist funktional vollständig
- { NOR } ist funktional vollständig
- { NAND } ist funktional vollständig

- $x + y = \neg(\neg x \cdot \neg y)$ de Morgansche Regel

  $x \cdot y = \neg(\neg x + \neg y)$ de Morgansche Regel

  $\neg x = x \text{ NOR } x$, $x \cdot y = (x \text{ NOR } x) \text{ NOR } (y \text{ NOR } y)$

  $\neg x = x \text{ NAND } x$, $x + y = (x \text{ NAND } x) \text{ NAND } (y \text{ NAND } y)$

x + y = (x NAND x) NAND (y NAND y)

x	y	x NAND x	y NAND y	(x NAND x) NAND (y NAND y)
0	0	1	1	0
0	1	1	0	1
1	0	0	1	1
1	1	0	0	1

= x OR y

Maxterm

- i sei Index von $f: B^n \to B$ und m_i der i-te Minterm von f.
- Die Funktion $M_i(x_1,\ldots x_n) := \neg m_i(x_1,\ldots x_n)$ heisst dann **i-ter Maxterm**.

- Beispiele:
  - $M_3(x_1,x_2,x_3) = \neg\,(\neg\, x_1 \cdot x_2 \cdot x_3) \quad = x_1 + \neg x_2 + \neg x_3$
  - $M_4(x_1,x_2,x_3) = \neg\,(x_1 \cdot \neg x_2 \cdot \neg x_3) \quad = \neg\, x_1 + x_2 + x_3$

- Maxterm M_i nimmt genau dann den Wert 0 an, wenn das Argument $(x_1,\ldots x_n)$ die Dualdarstellung von i liefert.

n-stellige De Morgan Regel für M_3

$$M_3(x_1,x_2,x_3) = \neg((\underbrace{\neg x_1 x_2)}_{a} \underbrace{x_3}_{b})$$

Zur Erinnerung:
$\neg(ab) = \neg a + \neg b$
$\neg\neg a = a$

$$= \neg(\underbrace{\neg x_1}_{a} \underbrace{x_2}_{b}) + \neg x_3)$$

$$= (\neg\neg x_1 + \neg x_2) + \neg x_3$$

$$= x_1 + \neg x_2 + \neg x_3$$

Konjunktive Normalform (KNF)

> Jede Boolesche Funktion f: $B^n \rightarrow B$ ist eindeutig darstellbar als Produkt (Konjunktion) der Maxterme ihrer **nicht**-einschlägigen Indizes.

> Diese Darstellung heisst

kanonische konjunktive Normalform (KNF) von f.

Beispiel: KNF

i	x1	x2	x3	f(x1,x2,x3)
0	0	0	0	0
1	0	0	1	0
2	0	1	0	0
3	0	1	1	1
4	1	0	0	0
5	1	0	1	1
6	1	1	0	0
7	1	1	1	1

- $f(x_1,x_2,x_3) = M_0 \cdot M_1 \cdot M_2 \cdot M_4 \cdot M_6$
  $= (x_1+x_2+x_3) \cdot (x_1+x_2+\neg x_3) \cdot (x_1+\neg x_2+x_3) \cdot (\neg x_1+x_2+x_3) \cdot (\neg x_1+\neg x_2+x_3)$

Vergleich DNF und KNF

- **DNF**

  $f(x_1,x_2,x_3) = m_3 + m_5 + m_7 = \neg x_1 x_2 x_3 + x_1 \neg x_2 x_3 + x_1 x_2 x_3$

- **KNF**

  $f(x_1,x_2,x_3) = M_0 \cdot M_1 \cdot M_2 \cdot M_4 \cdot M_6$
  $= (x_1+x_2+x_3) \cdot (x_1+x_2+\neg x_3) \cdot (x_1+\neg x_2+x_3) \cdot (\neg x_1+x_2+x_3) \cdot (\neg x_1+\neg x_2+x_3)$

DNF ist zu bevorzugen, wenn die Anzahl der einschlägigen Indizes kleiner ist als die Anzahl der nicht-einschlägigen; ansonsten ist KNF besser.

Extensionale vs. Intensionale Darstellung von Booleschen Funktionen

Extensionale Darstellung als Wertetabelle

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

Intensionale Darstellung als Boolescher Ausdruck ist nicht eindeutig

DNF

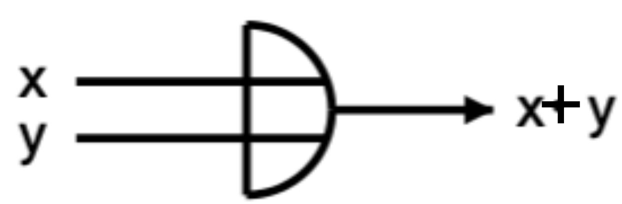
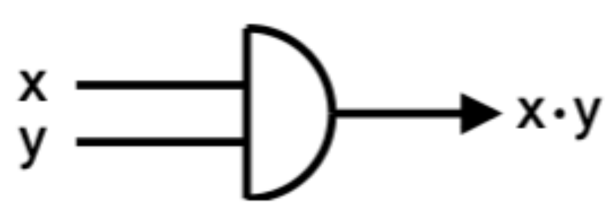
f(x,y) = (x * not(y)) + (not(x) * y)

oder

KNF

f(x,y) = (not(x)+not(y)) * (x+y)

Grundbausteine zur Realisierung von Booleschen Funktionen

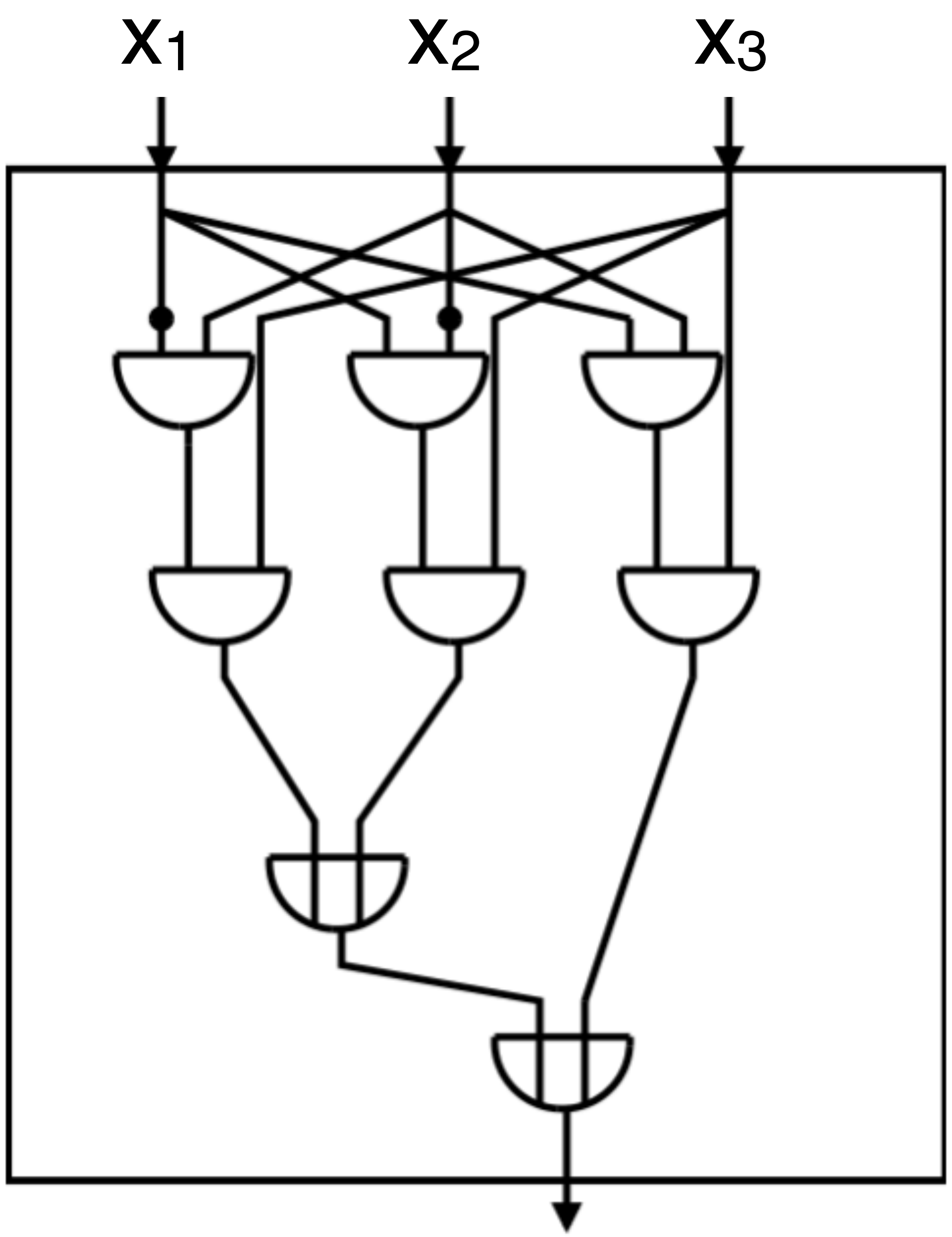
Funktion	Unser Symbol	IEEE-Symbol
Negation (Komplement-Gatter)		
Addition (Oder-Gatter)		
Multiplikation (Und-Gatter)		

Schaltung für diese Funktion?

i	**x1**	**x2**	**x3**	**f(x1,x2,x3)**
0	0	0	0	0
1	0	0	1	0
2	0	1	0	0
3	0	1	1	1
4	1	0	0	0
5	1	0	1	1
6	1	1	0	0
7	1	1	1	1

> Disjunktive Normalform:
$f(x_1,x_2,x_3) = m_3 + m_5 + m_7 = \neg x_1 x_2 x_3 + x_1 \neg x_2 x_3 + x_1 x_2 x_3$

Schaltung für Funktion in DNF



$\neg x_1 x_2 x_3 + x_1 \neg x_2 x_3 + x_1 x_2 x_3$

> Schaltzeit:
  $5 \cdot 10$ ps $= 5 \cdot 10^{-11}$s
> $f(x_1,x_2,x_3) = m_3 + m_5 + m_7 =$
  $\neg x_1 x_2 x_3 + x_1 \neg x_2 x_3 + x_1 x_2 x_3$

Beispiel (Fortsetzung)

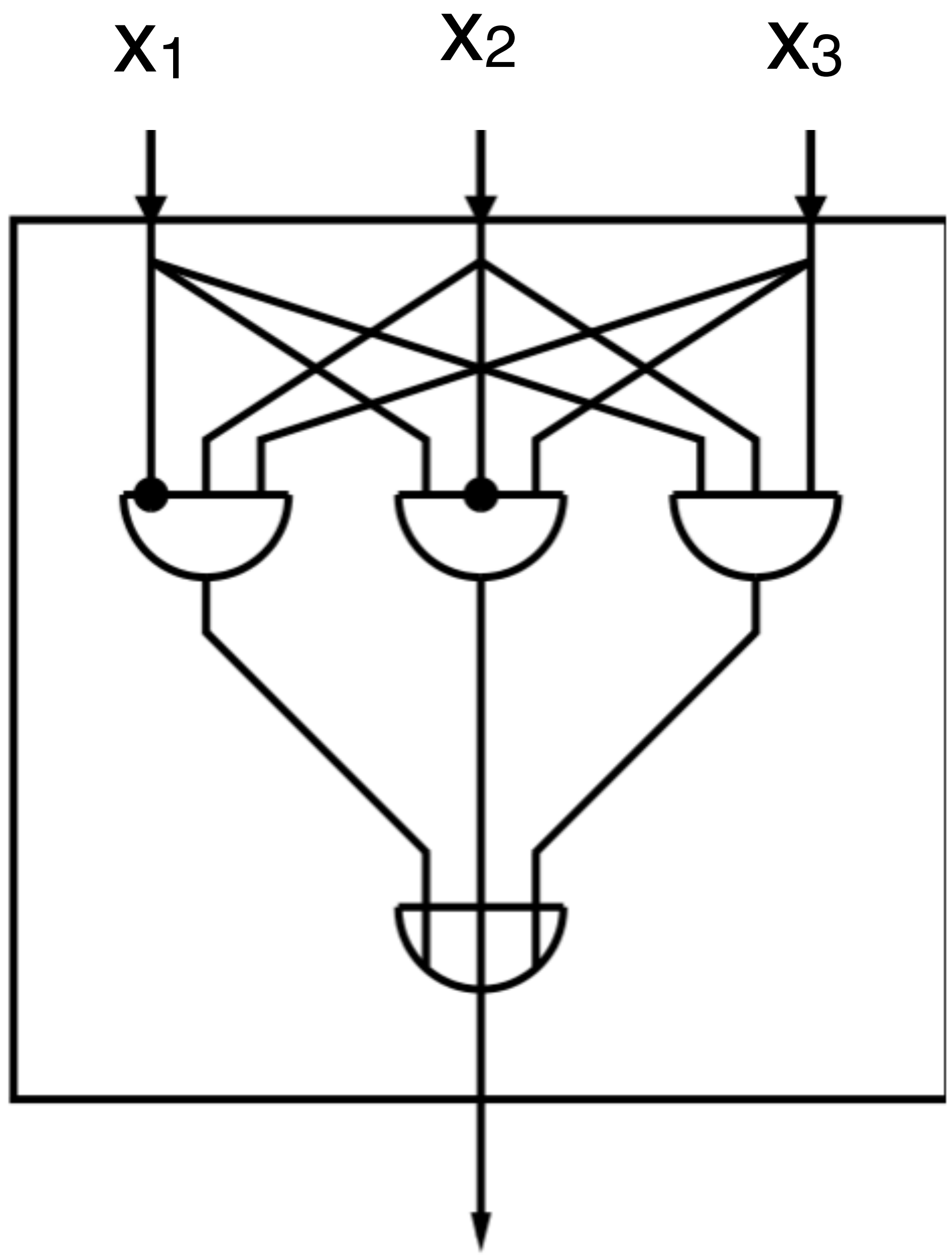
- Schaltzeit:

  $5 \cdot 10$ ps $= 5 \cdot 10^{-11}$s

- Vernachlässigt Zeit, welche Signal benötigt, um einen Leitungsweg zurückzulegen.

- In der Praxis wichtig: höchstens Lichtgeschwindigkeit $3 \cdot 10^5$ km/s $= 0.3 \cdot 10^{12}$ mm/s $= 0.3$ mm/ps

Beispiel: alternative Darstellung



$\neg x_1 x_2 x_3 + x_1 \neg x_2 x_3 + x_1 x_2 x_3$

- Negation integriert
- Assoziativgesetz angewendet

Gerichteter azyklischer Graph (DAG)

- Sei P eine endliche Menge und K Teilmenge von P x P eine Relation über P.
  Dann heisst G := (P,K) **gerichteter Graph**.

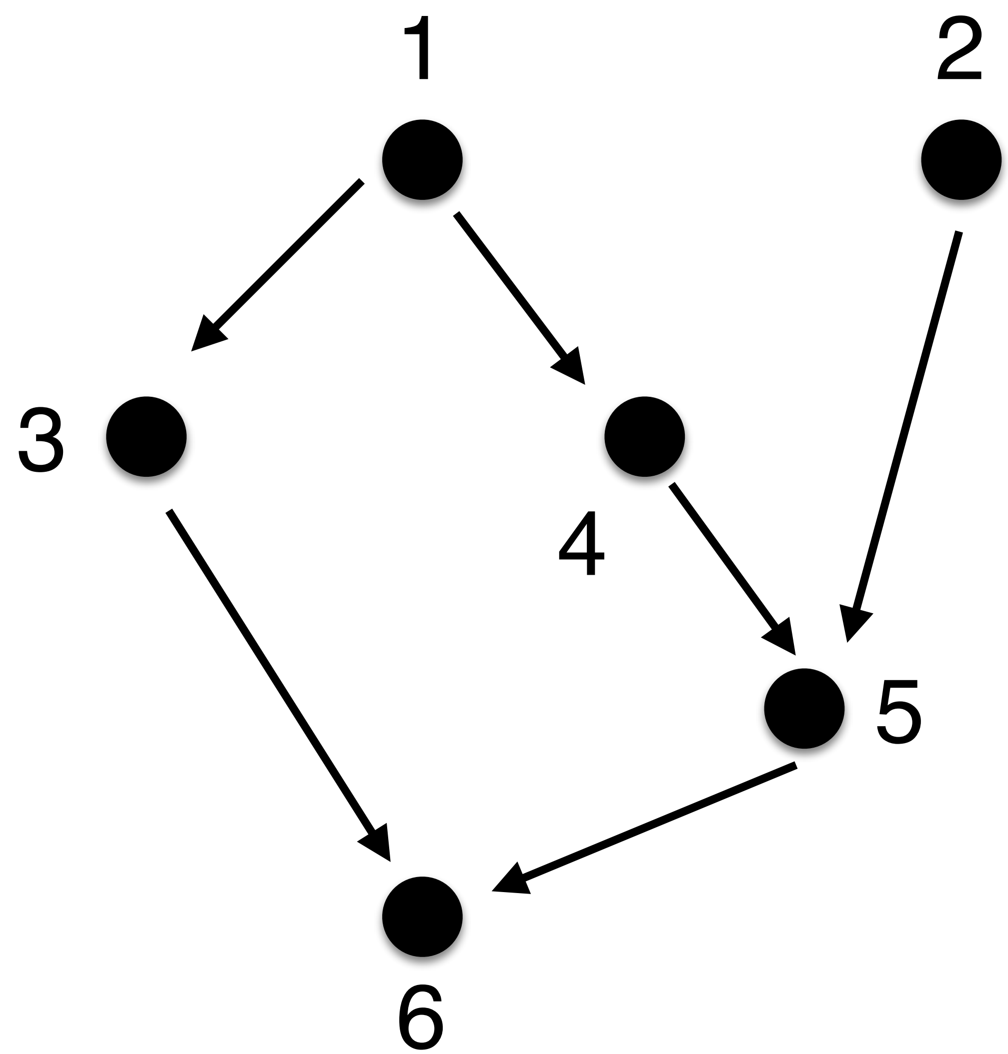
- Ein **Weg** in G ist ein Tupel $(p_1,\ldots,p_n)$ von Punkten in P so dass für alle $i=1,\ldots,n-1$ die Kante (p_i,p_{i+1}) zu K gehört.

- G heisst **azyklisch**, falls es keinen Weg in G gibt auf dem Anfangs- und Endpunkt übereinstimmen.

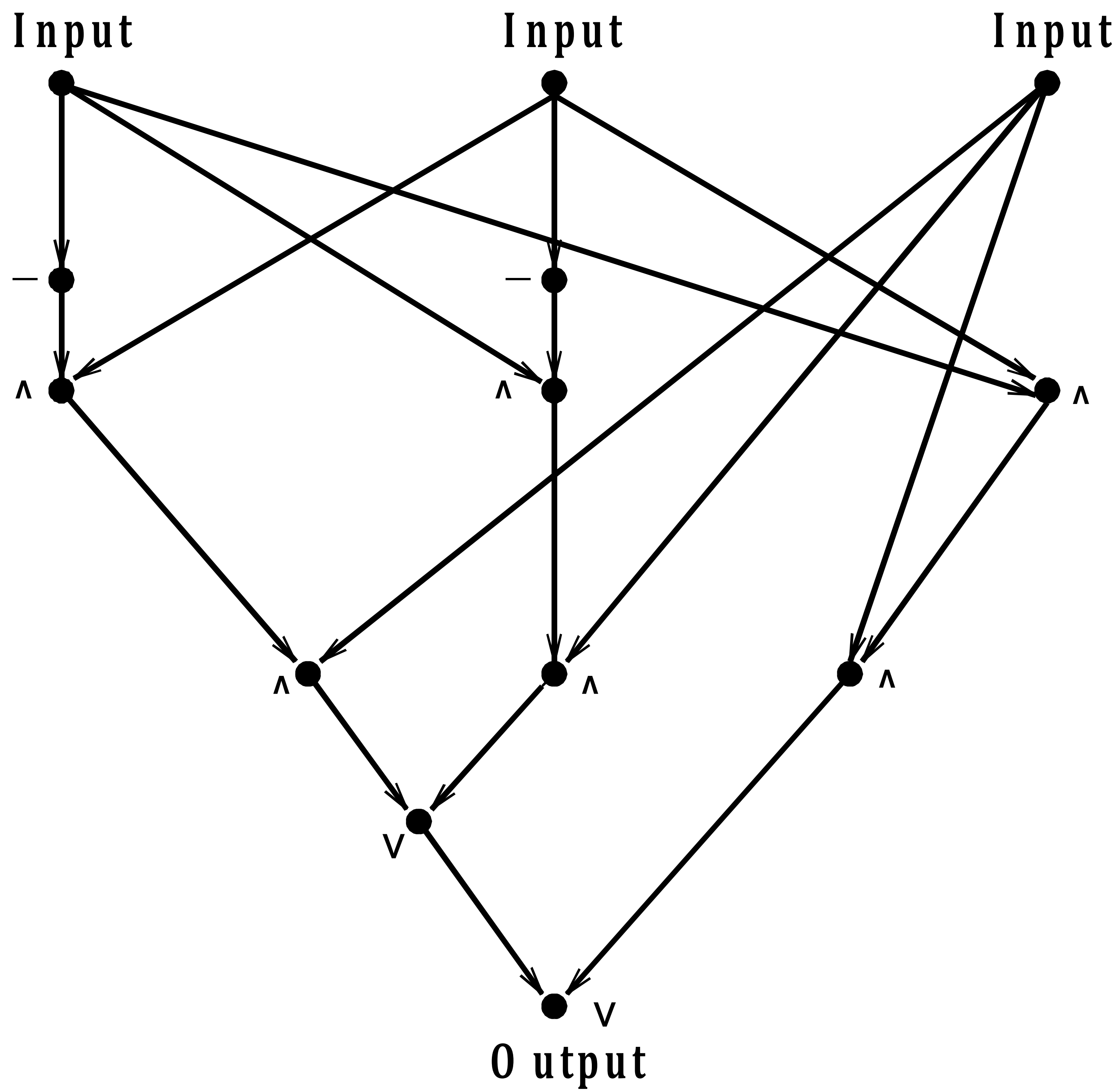
- Englisch: **directed acyclic graph (DAG)**

Beispiel DAG

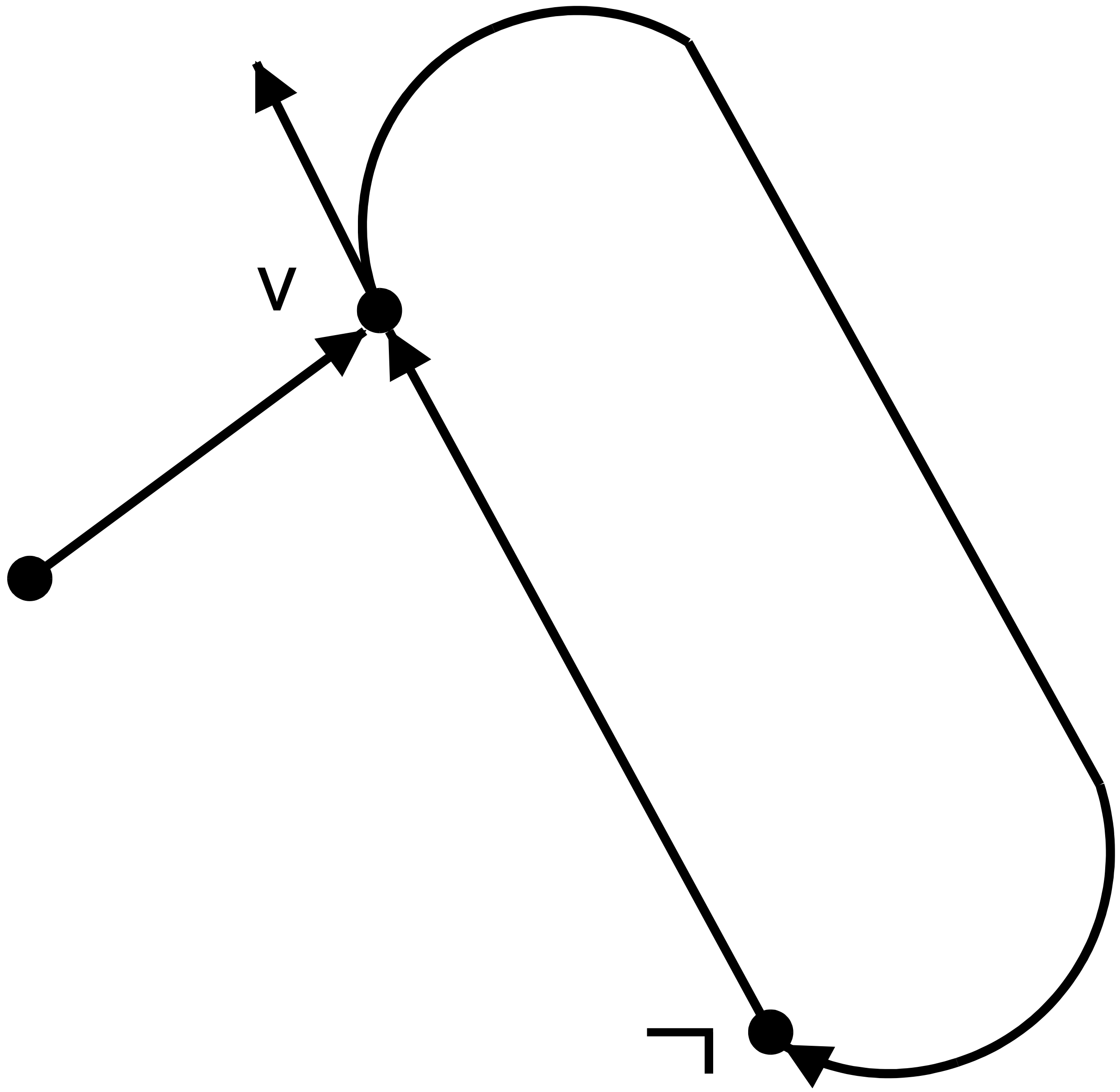
- V = { 1, 2, 3, 4, 5, 6 }
- E = { (1,3), (1,4), (4,5), (2,5), (5,6), (3,6) }

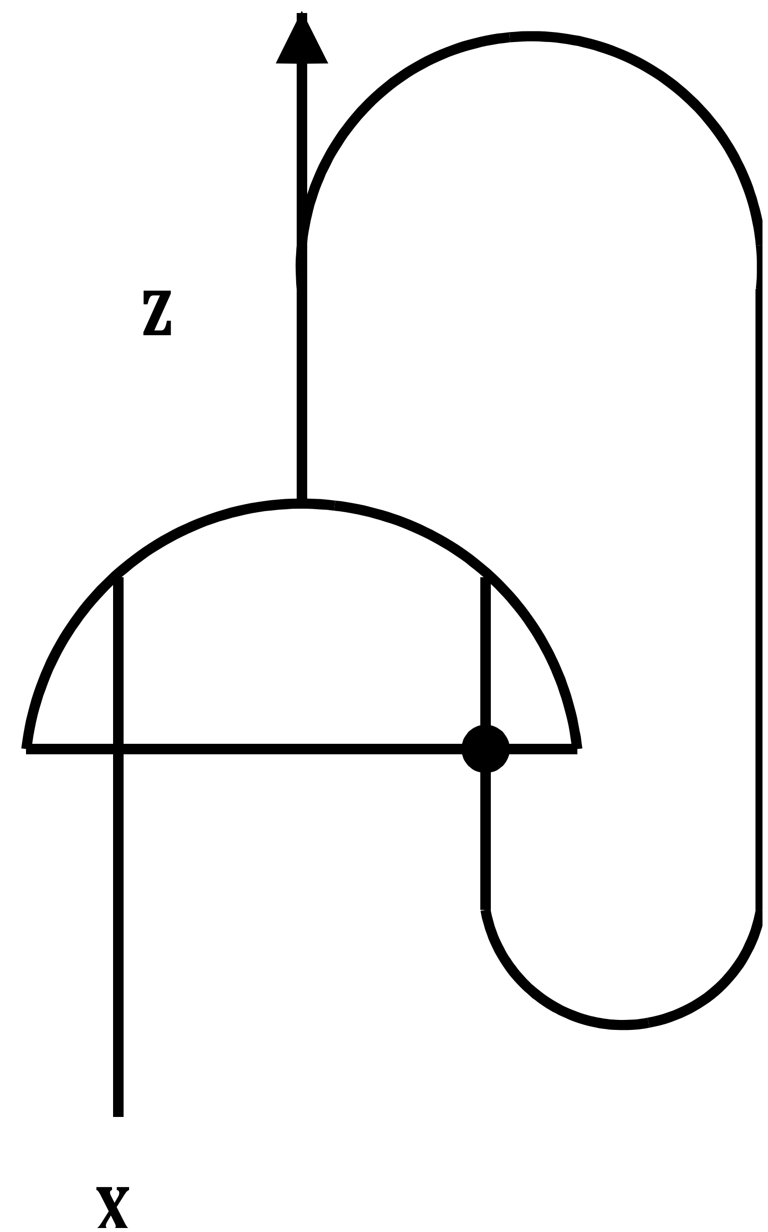


DAG-Darstellung



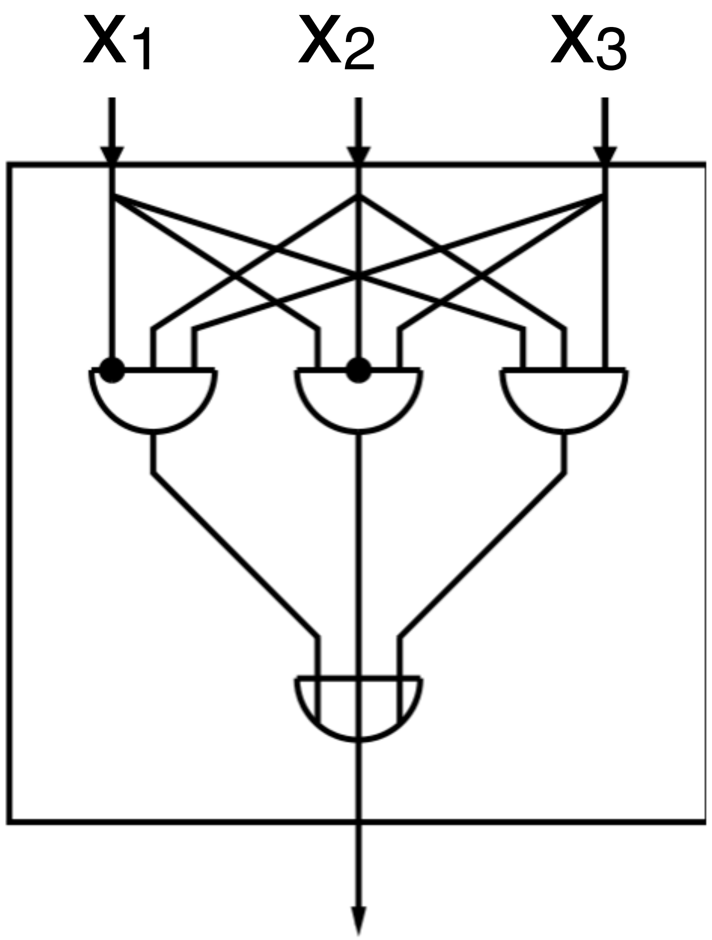
Flimmerschaltung





Fan-In, Fan-Out

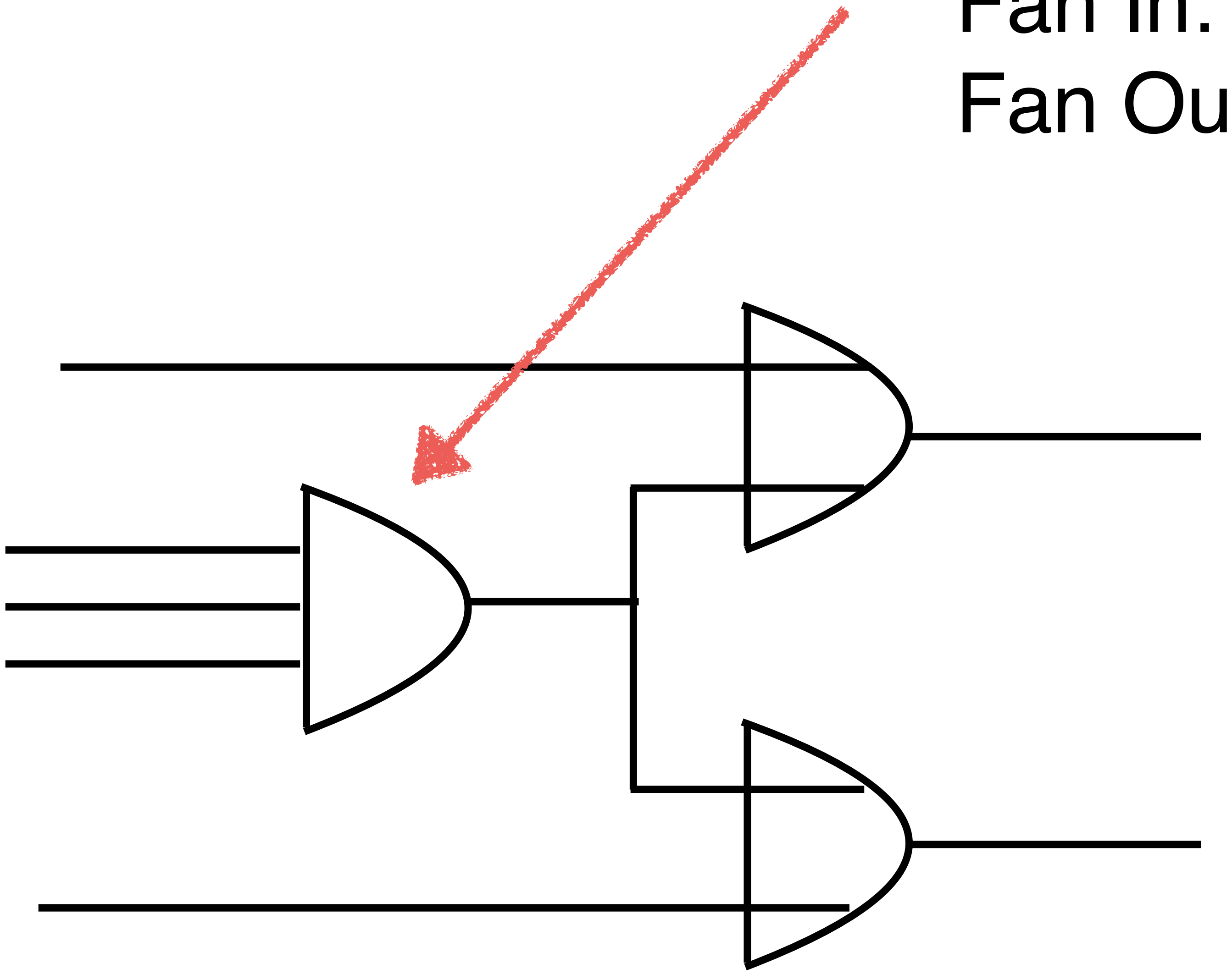
- **Fan-In**: Anzahl der Inputs eines Gatters
- **Fan-Out**: Anzahl der Inputs mit denen der Output eines Gatters verbunden ist.
- Beispiel:
  - X_i hat einen Fan-Out 3
  - Gatter haben einen Fan-Out 1 und Fan-In 3



$\neg X_1 X_2 X_3 + X_1 \neg X_2 X_3 + X_1 X_2 X_3$

Beispiel: Fan-In, Fan-Out

Fan In: 3
Fan Out: 2



Entwurfsprinzipien für Schaltungen

- **Geschwindigkeit**:
  wenige Stufen
- **Grösse**:
  kleine Schaltungen sind schneller und billiger
  grosse Schaltungen haben eine grössere Wahrscheinlichkeit für Produktionsfehler
- **Kleiner Fan-In und Fan-Out**:
  Hoher Fan-Out erfordert Treiberschaltungen zur Verstärkung
  Schaltungen mit hohem Fan-In und Fan-Out sind i.a. langsamer

Ziele

Sie kennen den Unterschied von extensionaler und intensionaler Darstellung von Funktionen

Sie können erklären, was Minterme und Maxterme sind

Sie können die DNF und KNF einer gegebenen Funktion bestimmen

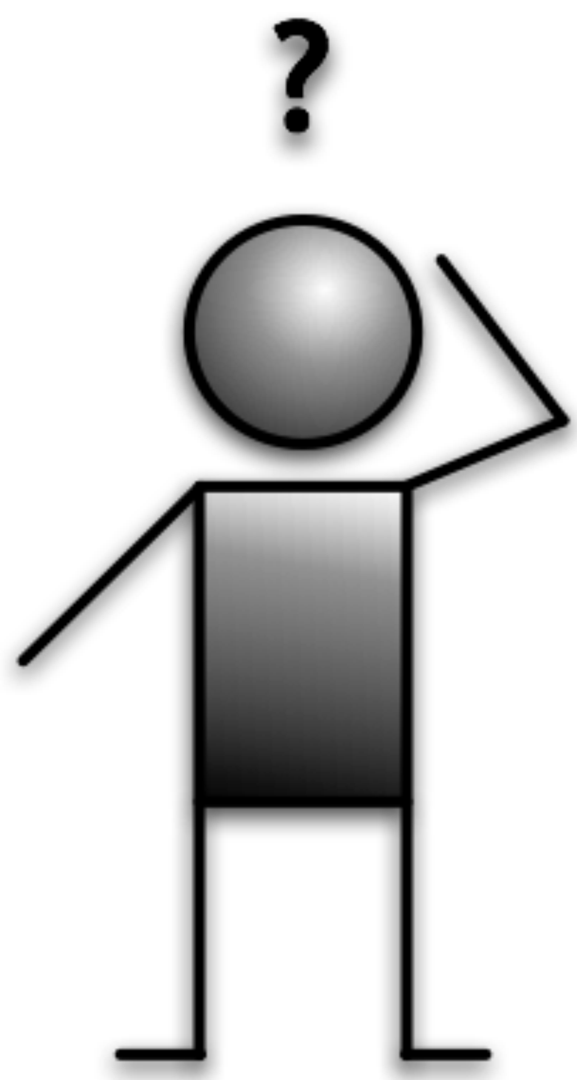
Sie können eine Menge von Funktionen auf eine andere Menge reduzieren und so funktionale Vollständigkeit beweisen

Sie kennen die Schaltsymbole für NOT, AND, NOR und können zu einer gegeben Booleschen Funktion eine Schaltung entwicklen

Sie können eine Schaltung als DAG darstellen

Sie kennen Designprinzipen für Boolesche Schaltungen

Fragen?



Zum Schluss