Kapitel 3 – Kombinatorische Logik

1. Kombinatorische Schaltkreise
2. Boolesche Algebren
3. **Boolesche Ausdrücke, Normalformen, zweistufige Synthese**
   3.1 Boolesche Ausdrücke, Disjunktive Normalform
   3.2 zweistufige Logikminimierung
4. Berechnung eines Minimalpolynoms

Albert-Ludwigs-Universität Freiburg

Prof. Dr. Armin Biere

Institut für Informatik

Sommersemester 2024

Kombinatorische Schaltkreise – zweistufig

- **Ziel:**
  Wir werden zeigen, dass sich jede Boolesche Funktion als ein Polynom, d.h. als eine Disjunktion (ODER-Verknüpfung) von Monomen, die ihrerseits Konjunktionen (UND-Verknüpfungen) von Eingangsvariablen und negierten Eingangsvariablen sind, darstellen lässt.
- Wir werden für solche Darstellungen Kostenkriterien aufstellen und diese optimieren.
- Monome und Polynome sind spezielle Boolesche Ausdrücke
- Beginne daher mit einer exakten Definition, was wir unter einem Booleschen Ausdruck verstehen.

Boolesche Ausdrücke - allgemein

- Formal vollständige Definition Boolescher Ausdrücke
    - Syntax (korrekte Schreibweise) $\rightarrow$ Def. Boolescher Ausdrücke $BE(X_n)$
    - Semantik (Bedeutung) $\rightarrow$ Interpretationsfunktion Ψ von $BE(X_n)$

Syntax Boolescher Ausdrücke

- Sei $X_n = \{x_1, \dots, x_n\}$ eine endliche Menge von Variablen.
- Sei $A = X_n \cup \{0, 1, +, \cdot, \sim, (,)\}$ ein Alphabet.

Definition

Die Menge $BE(X_n)$ der vollständig geklammerten Booleschen Ausdrücke über X_n ist die kleinste Teilmenge von A^*, die folgendermaßen induktiv definiert ist:

- $0, 1$ und $x_i \in X_n$ $i = 1, \dots, n$ sind Boolesche Ausdrücke
- Sind g und h Boolesche Ausdrücke, so auch
  - die Disjunktion $(g + h)$,
  - die Konjunktion $(g \cdot h)$,
  - die Negation $(\sim g)$.

Bsp.: $((x_1 \cdot x_2) + (\sim x_3)) \in BE(X_3)$.

Schreibweise von $BE(X_n)$

- **Konvention**: Negation $\sim$ bindet stärker als Konjunktion $\cdot$, Konjunktion $\cdot$ bindet stärker als Disjunktion $+$.
  - → Klammern können weggelassen werden, ohne dass Mehrdeutigkeiten entstehen.
- Man schreibt auch
  - statt $\cdot$: $\wedge$,
  - statt $+$: $\vee$,
  - statt $\sim x$: $\neg x, x', \overline{x}$.

→ So „vereinfachte" Ausdrücke entsprechen zwar nicht genau der obigen Definition, es gibt aber für jeden solchen Ausdruck einen äquivalenten vollständig geklammerten Ausdruck im Sinne der Definition.

- **Beispiel**: Der äquivalente vollständige geklammerte Ausdruck für „$x_1 \wedge x_2 + \neg x_3$" wäre „$((x_1 \cdot x_2) + (\sim x_3))$".

Semantik Boolescher Ausdrücke

Definition

Jedem Booleschen Ausdruck $BE(X_n)$ kann durch eine Interpretationsfunktion $\Psi : BE(X_n) \to \mathbb{B}_n$ eine Boolesche Funktion zugeordnet werden.

Ψ wird folgendermaßen induktiv definiert:

- $\Psi(0) = \mathbf{0}$; $\Psi(1) = \mathbf{1}$;
- $\Psi(x_i)(\alpha_1, \ldots, \alpha_n) = \alpha_i$ für alle $\alpha \in \mathbb{B}^n$ (Projektion)
- $\Psi((g + h)) = \Psi(g) + \Psi(h)$ (Disjunktion)
- $\Psi((g \cdot h)) = \Psi(g) \cdot \Psi(h)$ (Konjunktion)
- $\Psi((\sim g)) = \sim (\Psi(g))$ (Negation)

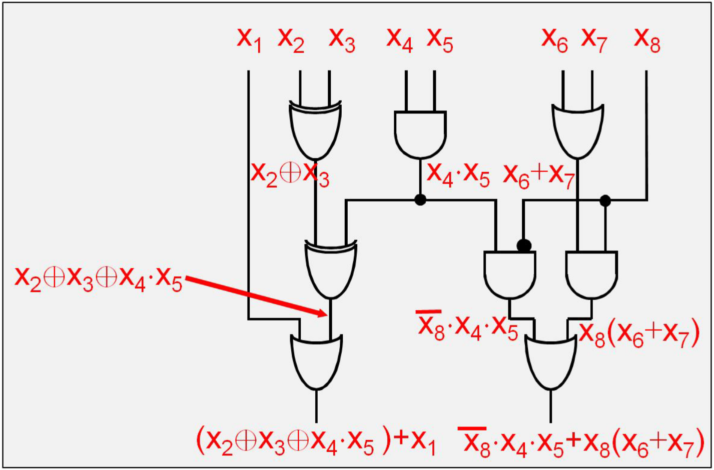
Alternative Betrachtung der Semantik Boolescher Ausdrücke

- Sei e ein Boolescher Ausdruck.
  - $\Psi(e)(\alpha)$ für ein $\alpha \in \mathbb{B}^n$ ergibt sich durch Ersetzen von x_i durch α_i in e, für alle i und Rechnen in der Booleschen Algebra $\mathbb{B}$.
  - **Bsp.:**
    $\Psi(((x_1 \cdot x_2) + (\sim x_3)))(0, 0, 1) = ((0 \cdot 0) + (\sim 1)) = (0 + 0) = 0$
  - Gilt $\Psi(e) = f$ für eine Boolesche Funktion $f \in \mathbb{B}_n$, so sagen wir, dass e ein Boolescher Ausdruck für f ist, bzw. dass e die Boolesche Funktion f beschreibt.
    In Zukunft schreiben wir auch $f = e$ statt $\Psi(e) = f$.
  - Zwei Boolesche Ausdrücke e_1 und e_2 heißen äquivalent ($e_1 \equiv e_2$) genau dann, wenn $\Psi(e_1) = \Psi(e_2)$.
    Sie sind gleich, wenn $e_1 = e_2$.

Beziehung zwischen Schaltkreisen und Booleschen Ausdrücken (1/2)

- Zu jedem Ausgang eines Schaltkreises lässt sich durch “symbolische Simulation” ein Boolescher Ausdruck berechnen, der die entsprechende Boolesche Funktion darstellt.
- Symbolische Simulation benutzt zur Simulation eines Schaltkreises keine festen Booleschen Werte an den Inputs, sondern Boolesche Variablen.
- Es wird dann für jeden Knoten ein Boolescher Ausdruck zu der Funktion bestimmt, die der Knoten berechnet.

Beziehung zwischen Schaltkreisen und Booleschen Ausdrücken (2/2)



($x_1 \oplus x_2$ ist eine abkürzende Schreibweise für $x_1\overline{x_2} + \overline{x_1}x_2$.)

Spezielle Boolesche Ausdrücke: Literale

Definition

Als Literal einer Booleschen Funktion $f \in \mathbb{B}_n$ wird der Ausdruck x_i oder x_i' bezeichnet, wobei $x_i, i \in 1, \ldots, n$, eine Eingangsvariable von f.

- x_i (auch x_i^1 geschrieben) wird positives Literal,
  x_i' (auch x_i^0 geschrieben) wird negatives Literal genannt.

- Anmerkung:
  Eine andere Notation für ein negatives Literal x' ist $\neg x$ oder $\overline{x}$.
- Wie schon erwähnt bezeichnet das Literal x_i die Boolesche Funktion $g \in \mathbb{B}_n$ mit $g(\alpha_1, \ldots, \alpha_n) = 1$ genau dann, wenn $\alpha_i = 1$ und ...
- ... x_i' bezeichnet die Boolesche Funktion $h \in \mathbb{B}_n$ mit $h(\alpha_1, \ldots, \alpha_n) = 1$ genau dann, wenn $\alpha_i = 0$.

UNI FREIBURG

Spezielle Boolesche Ausdrücke: Monome

Definition

- Ein Monom ist eine Konjunktion von Literalen, in der kein Literal mehr als einmal vorkommt und zu keiner Variable sowohl das positive als auch das negative Literal vorkommt. Außerdem ist „1” ein Monom.
  - $x_1 x_2 x_3'$ und $x_1' x_3$ sind Monome, $x_2 x_3 x_3'$ ist kein Monom, $x_2 x_3 x_3$ ist kein Monom.
- Ein Monom heißt vollständig oder Minterm, wenn jede Variable entweder als positives oder als negatives Literal vorkommt.
  - Wenn drei Variablen x_1, x_2, x_3 betrachtet werden, ist $x_1 x_2 x_3'$ ein Minterm, $x_1' x_3$ ist kein Minterm.
- Für eine Eingabebelegung $\alpha \in \mathbb{B}^n$ heißt

$$m(\alpha) = \bigwedge_{i=1}^{n} x_i^{\alpha_i} \qquad \text{(Notation: } x_i^1 := x_i,\ x_i^0 := x_i'\text{)}$$

der zu α gehörende Minterm.

Monome als Beschreibung Boolescher Funktionen

- Beispiel: Das Monom $m = x_i x_j'$ bezeichnet die Boolesche Funktion $f \in \mathbb{B}_n$ mit $f(\alpha_1, \ldots, \alpha_n) = 1$ genau dann, wenn $\alpha_i = 1$ und $\alpha_j = 0$. Wir schreiben vereinfachend auch $f = x_i x_j'$.
- Beispiel: Der Minterm $m(0,1,0,1) = \overline{x_1} x_2 \overline{x_3} x_4$ bezeichnet die Boolesche Funktion $f \in \mathbb{B}_4$ mit $f(\alpha) = 1$ genau dann, wenn $\alpha = (0,1,0,1)$.

Spezielle Boolesche Ausdrücke: Polynome

- Eine Disjunktion von paarweise verschiedenen Monomen heißt Polynom. Außerdem ist „0” ein Polynom. Sind alle Monome des Polynoms vollständig, so heißt das Polynom vollständig.

  Beispiel: Bei einer Booleschen Funktion mit drei Variablen x_1, x_2, x_3 ist
  - $x_1' x_2 + x_2' x_3$ ein Polynom,
  - $x_1' x_2' x_3 + x_1 x_2' x_3$ ein vollständiges Polynom.
  - Das Polynom $x_1' x_2 + x_2' x_3$ beschreibt die Boolesche Funktion $f \in \mathbb{B}_3$ mit $f(\alpha_1, \alpha_2, \alpha_3) = 1$ genau dann, wenn $\alpha_1 = 0, \alpha_2 = 1$ oder $\alpha_2 = 0, \alpha_3 = 1$. Wir schreiben vereinfachend auch $f = x_1' x_2 + x_2' x_3$.

Disjunktive Normalform

- Ein Polynom für f heißt auch disjunktive Normalform (DNF) von f. Ein vollständiges Polynom für f heißt auch kanonische disjunktive Normalform (KDNF) von f.
  - $f_1 = x_1'x_2' + x_2'x_3 + x_1x_2$ ist in DNF, aber nicht in KDNF.
- Manchmal auch als Summe von Produkten bezeichnet

  engl. "sum of products (SOP)"
- nicht zu verwechseln mit Konjunktiver Normalform (KNF) bei SAT und in der Logik

  engl. "conjunctive normal form (CNF)" bzw. "product of sums (POS)"

Bestimmung der kanonischen disjunktiven Normalform

- Für $f : \mathbb{B}^n \to \mathbb{B}$ heißt $ON(f) := \{\alpha \in \mathbb{B}^n \mid f(\alpha) = 1\}$ die Erfüllbarkeitsmenge (ON-Menge) von f.
- Die KDNF ist gegeben durch $f = \sum_{\alpha \in ON(f)} m(\alpha)$
- Die KDNF ist (bis auf Anordnung von Literalen in Mintermen und von Monomen im Polynom) eindeutig.
- Beispiel: KDNF für $f_1 = x_1' x_2' + x_2' x_3 + x_1 x_2$

$$
\begin{aligned}
f_1 &= m(000) + m(001) \\
&\quad + m(101) + m(110) + m(111) \\
&= x_1' x_2' x_3' + x_1' x_2' x_3 \\
&\quad + x_1 x_2' x_3 + x_1 x_2 x_3' + x_1 x_2 x_3
\end{aligned}
$$

x_1	x_2	x_3	f_1
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

- Anmerkung: Analog zur Erfüllbarkeitsmenge ist $OFF(f) := \{\alpha \in \mathbb{B}^n \mid f(\alpha) = 0\}$ als die Unerfüllbarkeitsmenge (OFF-Menge) definiert.

Realisierungen von DNF

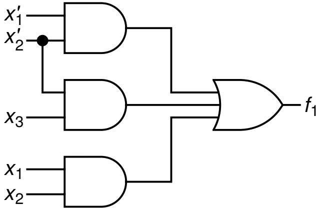
- Erste Möglichkeit:

  Benutze „gewöhnliche” UND- und ODER-Gatter.

- Zweite Möglichkeit: PLAs (Programmable Logic Arrays)
  - Programmierbare logische Felder können nur Funktionen in DNF implementieren.
  - Sie benötigen dafür weniger Transistoren als eine Realisierung mit UND- und ODER-Gattern.

Realisierung durch Logikgatter

- Bilde erst alle Monome durch UND-Gatter.
- Verbinde dann alle Monome mit ODER-Gattern.
  - Notation: Man verzichtet in der Regel auf die Abbildung von Invertern.

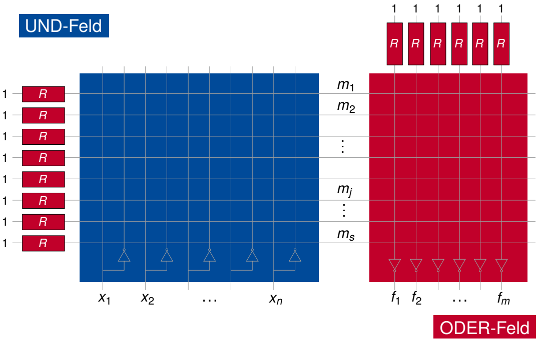


- Die Kosten ergeben sich dann aus allen benötigten UND- und ODER-Gattern.

Programmierbare logische Felder (PLA)

Zweistufige Darstellung zur Realisierung von Booleschen Polynomen.

$f_i = m_{i1} + m_{i2} + \cdots + m_{ik}$ mit m_{iq} aus $\{m_1, \ldots, m_s\}$



Enthält Monom m_j k Literale, so werden k Transistoren in der entsprechenden Zeile des **UND-Feldes** benötigt.

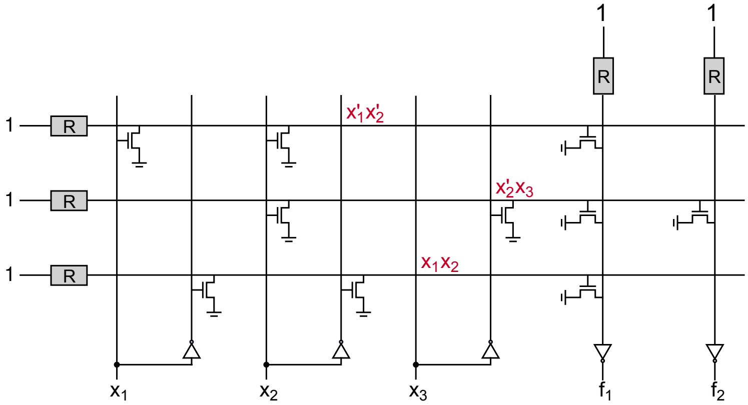
Besteht die Beschreibung von Funktion f_t aus p Monomen, so benötigt man p Transistoren in der entsprechenden Spalte des **ODER-Feldes**.

Fläche: $\sim (m + 2n) \times$ (Anzahl der benötigten Monome)

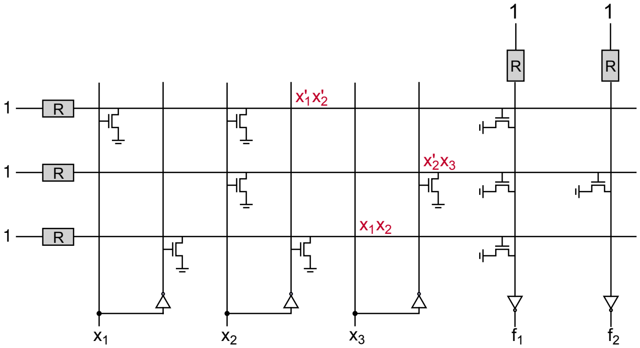
PLAs: Realisierungsdetails

Beispiel

$f_1 = x_1' x_2' + x_2' x_3 + x_1 x_2$
$f_2 = x_2' x_3$

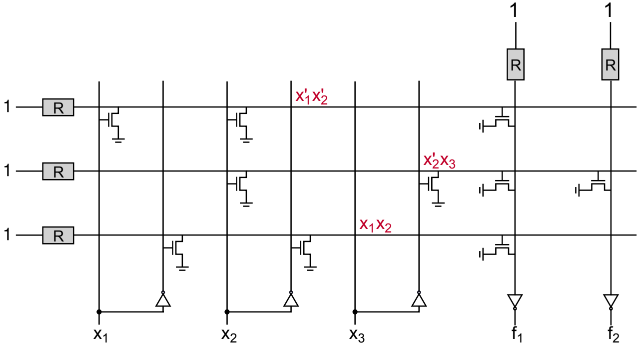


Realisierungsdetails, waagerechte Monomleitungen



- Ein n-Kanal-Transitor leitet, wenn sein Gate mit 1 belegt ist.
- Wenn ein n-Kanal-Transistor leitet, dann zieht er die entsprechende Monomleitung auf 0.
- Bsp.: Die erste Monomleitung ist genau dann 1, wenn *beide* Transistoren der Reihe sperren, d.h. wenn $x_1 = 0$ und $x_2 = 0$.
  $\Rightarrow$ Die Leitung realisiert die Funktion $x_1' x_2'$.

Realisierungsdetails, senkrechte Funktionsleitungen



- Ein n-Kanaltransitor leitet, wenn sein Gate mit 1 belegt ist.
- Wenn n-Kanal-Transistor leitet, dann zieht er die entsprechende Leitung auf 0.
- Bsp.: Der f_1-Ausgang ist genau dann 1, wenn der Eingang des zugehörigen Inverters 0 ist.
  Der Inverter-Eingang ist genau dann 0, wenn mindestens einer der drei Transistoren der Spalte leitet, d.h. wenn $x_1'x_2' = 1$ oder $x_2'x_3 = 1$ oder $x_1x_2 = 1$.
  $\Rightarrow$ Der f_1-Ausgang realisiert die Funktion $x_1'x_2' + x_2'x_3 + x_1x_2$.

- Sei $q = q_1 \cdot \dots \cdot q_r$ ein Monom, dann sind die Kosten $|q|$ von q gleich der Anzahl der zur Realisierung von q benötigten Transistoren im PLA, also $|q| := r$.
- Seien $p_1, \dots, p_m$ Polynome, dann bezeichne $M(p_1, \dots, p_m)$ die Menge der in diesen Polynomen verwendeten Monome.
  - Die primären Kosten $cost_1(p_1, \dots, p_m)$ einer Menge $p_1, \dots, p_m$ von Polynomen sind gleich der Anzahl der benötigten Zeilen im PLA, um $p_1, \dots, p_m$ zu realisieren, also $cost_1(p_1, \dots, p_m) = |M(p_1, \dots, p_m)|$.
  - Die sekundären Kosten $cost_2(p_1, \dots, p_m)$ einer Menge $\{p_1, \dots, p_m\}$ von Polynomen sind gleich der Anzahl der benötigten Transistoren im PLA, also $cost_2(p_1, \dots, p_m) = \sum_{q \in M(p_1, \dots, p_m)} |q| + \sum_{i=1,\dots,m} |M(p_i)|$.

Kombiniertes Kostenmaß

- Sei im Folgenden $cost = (cost_1, cost_2)$ die Kostenfunktion mit der Eigenschaft, dass für zwei Polynommengen $\{p_1, \ldots, p_m\}$ und $\{p'_1, \ldots, p'_m\}$ die Ungleichung

$$cost(p_1, \ldots, p_m) \leq cost(p'_1, \ldots, p'_m)$$

  genau dann gilt, wenn
  - entweder $cost_1(p_1, \ldots, p_m) < cost_1(p'_1, \ldots, p'_m)$
  - oder $cost_1(p_1, \ldots, p_m) = cost_1(p'_1, \ldots, p'_m)$ und $cost_2(p_1, \ldots, p_m) \leq cost_2(p'_1, \ldots, p'_m)$

Kosten der Realisierung

Welche der folgenden PLAs ist die kostengünstigste Lösung?

a. PLA 1 mit 10 Zeilen und 19 Transistoren.
b. PLA 2 mit 15 Zeilen und 30 Transistoren.
c. PLA 3 mit 12 Zeilen und 15 Transistoren.
d. PLA 4 mit 12 Zeilen und 19 Transistoren.