



# Technische Informatik 1

**Prof. Dr. Rolf Drechsler**

**Christina Plump**

# Überblick

## Teil 1: Der Rechneraufbau (Kapitel 2-5)

- Rechner im Überblick
- Pipelining
- Speicher
- Parallelverarbeitung

## Teil 2: Der Funktionalitätsaufbau (Kapitel 6-12)

- Kodierung
- Grundbegriffe, Boolesche Funktionen
- Darstellungsmöglichkeiten
- **Schaltkreise, Synthese, spezielle Schaltkreise**

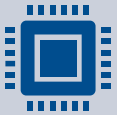


# Kapitel 10: Schaltkreise

## Lernziele

- Funktionsweise von unterschiedlichen Technologien kennenlernen
- Syntax und Semantik eines Schaltkreises kennenlernen und verstehen
- Kosten und Tiefe eines Schaltkreises kennen, verstehen und bestimmen können
- (Symbolische) Simulation von Booleschen Funktionen auf Schaltkreisen kennen, verstehen und anwenden können
- Begriffe hierarchischer Entwurf und Teilschaltkreis kennen und verstehen

## Verfügbare Technologien



### Nurlesespeicher

Read Only Memory (ROM)  
Programmable ROM (PROM)  
Erasable PROM (EPROM), ...



### Zweistufige Realisierungen

Progr. Logische Felder (PLA)



### Mehrstufige Realisierungen

Gate-Arrays- und Sea-of-Gates Entwurf  
Field Programmable Logic Arrays (FPGA),  
Application Specific Integrated Circuit (ASIC)

# Konkurrierende Optimierungsziele

## belegte Fläche

- Ausbeute

## benötigte Reaktionszeiten

- Korrektheit des Gesamtsystems

## Testbarkeit

- Korrektheit des verkauften Chips

## Leistungsverbrauch

- neue Märkte (Notebooks, Pads, Watches, IoT)

## Entwicklungszeit und -kosten

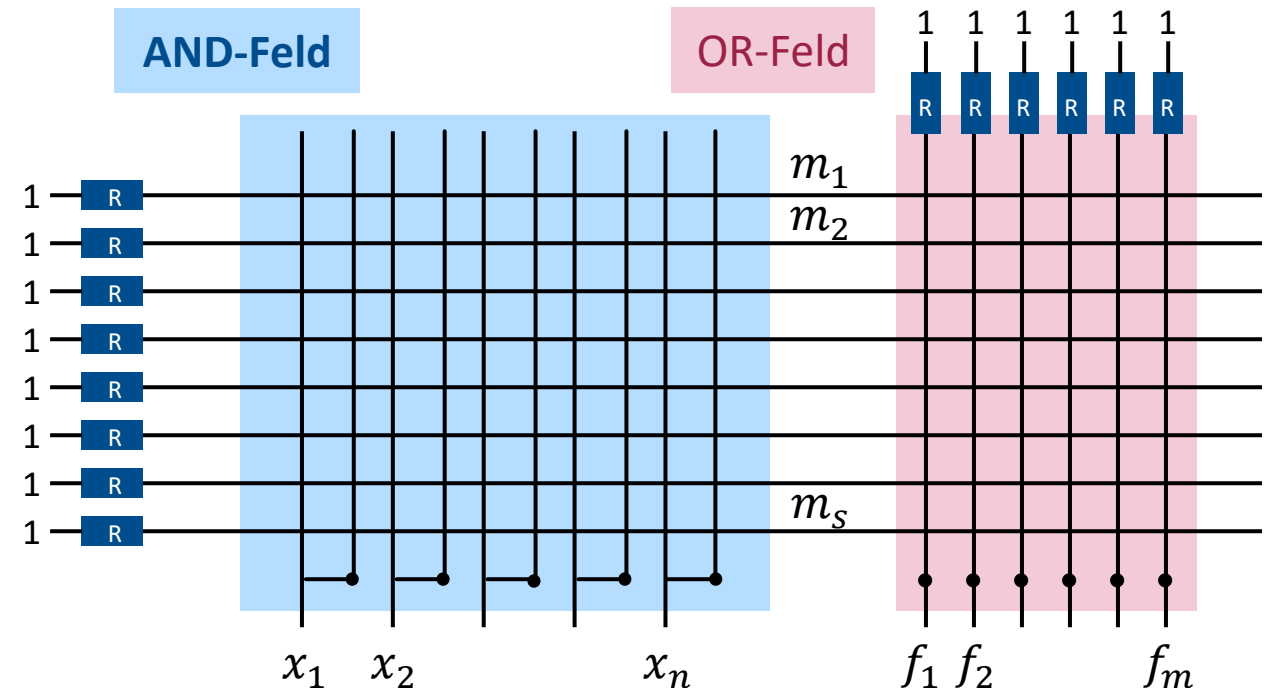
- time-to-market
- Wirtschaftskraft der Kunden

# Programmierbare logische Felder (PLA)

Zweistufige Darstellung zur Realisierung von Booleschen Polynomen der Form

$$f_i = m_{i1} + m_{i2} + \dots + m_{ik}, m_{iq} \in \{m_1, \dots, m_s\}$$

- Variablen
- Monome im AND-Feld
  - Enthält Monom  $m_j$   $k$  Literale, so werden  $k$  Transistoren in der entsprechenden Zeile des AND-Feldes benötigt
- Polynome im OR-Feld
  - Besteht die Beschreibung von Funktion  $f_t$  aus  $p$  Monomen, so benötigt man  $p$  Transistoren in der entsprechenden Spalte des OR-Feldes



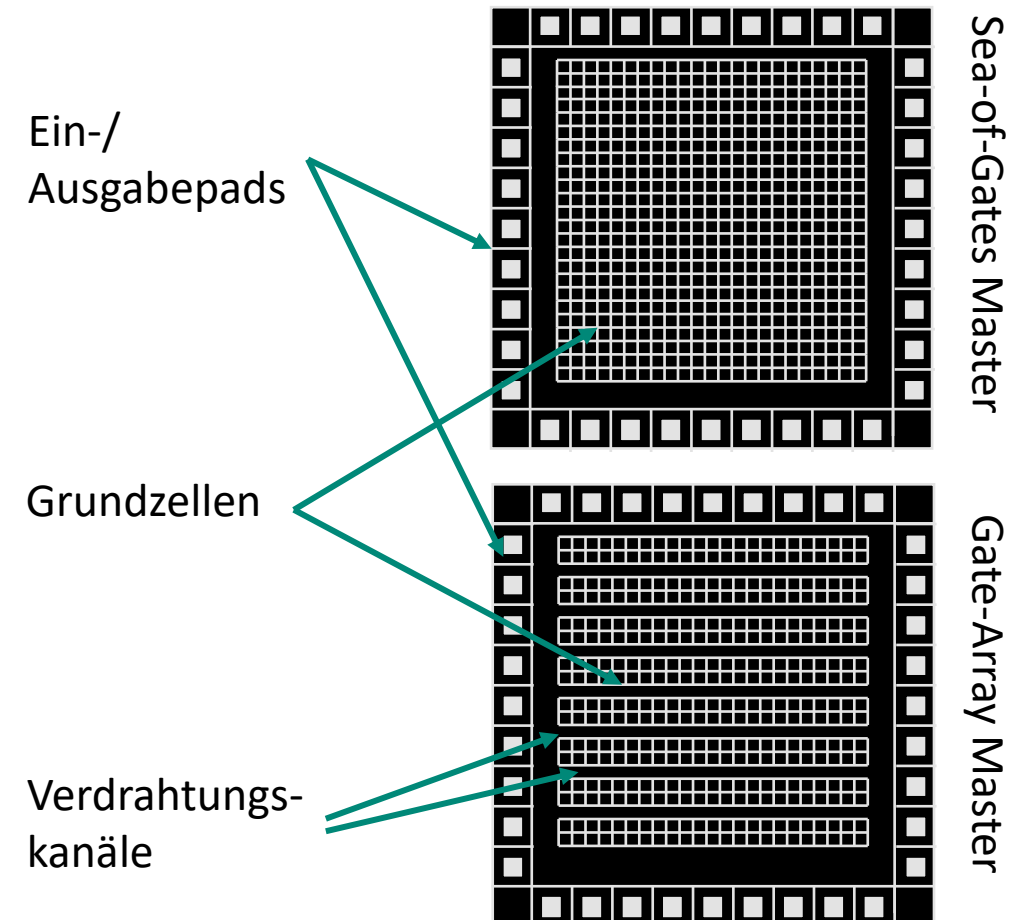
**Fläche:**  $(m + 2n) \cdot s$

**Laufzeit:** konstant (ohne Berücksichtigung der Kapazität)

# Mehrstufige Realisierungen: Gate-Array- und Sea-of-Gates-Entwürfe

## Halbkundenspezifischer Entwurf

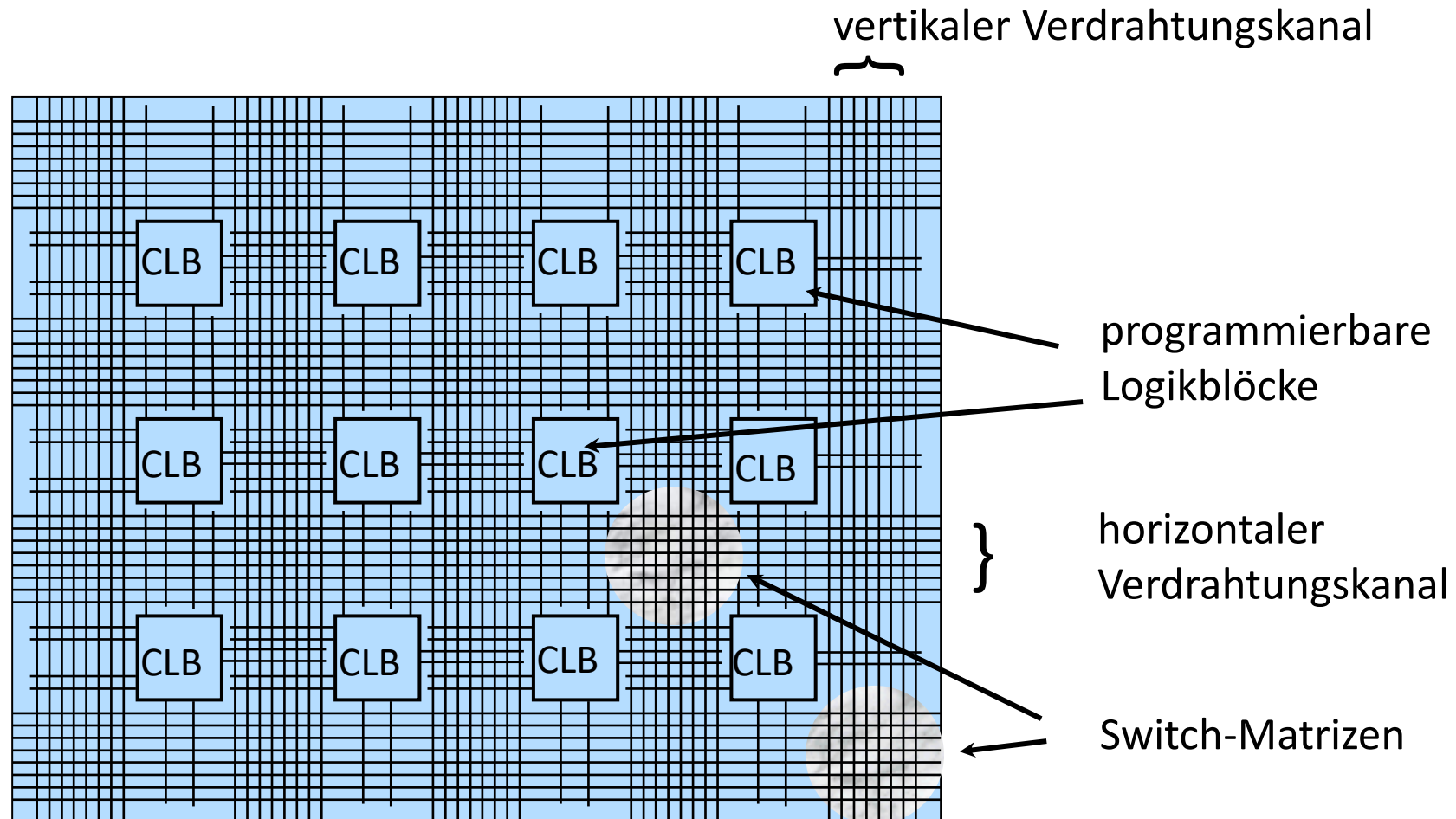
- Kundenunabhängig
  - Grundzellen sind kundenunabhängig vorgefertigt
  - Grundzellen bestehen jeweils aus 4-6 Transistoren
- Kundenabhängig
  - Intra-/Interzellenverdrahtung
  - Funktionsblöcke in Bibliotheken vorgegeben





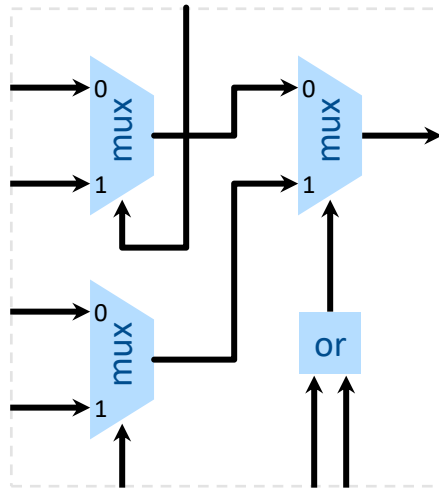
# Mehrstufige Realisierungen: Field Programmable Gate Arrays (FPGA)

FPGA von Xilinx:



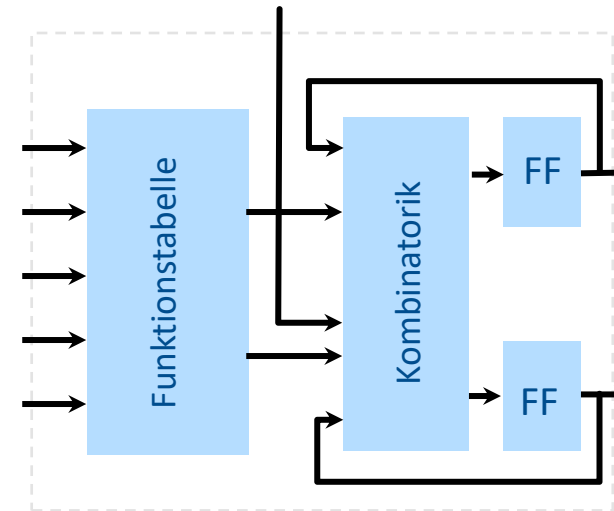
# FPGA-Logikblöcke

## Multiplexer-basierte Blöcke



(Actel Act-1 Funktionsblock)

## SRAM-basierte Blöcke

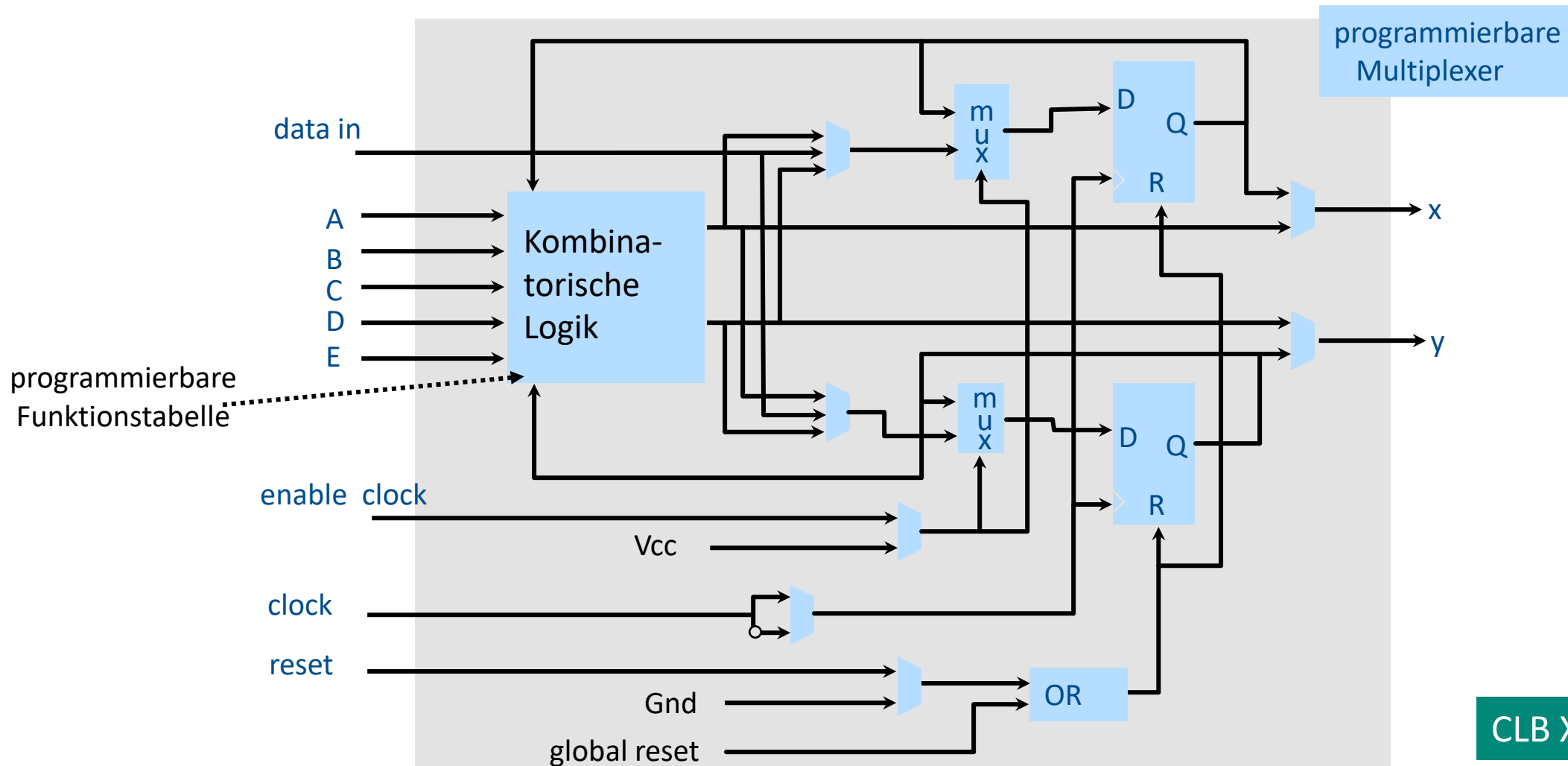


(Skizze des Xilinx 3000 Blocks)

Funktionstabelle besteht aus 32 SRAM-Speicherzellen und realisiert:

- Eine Funktion mit fünf Eingängen
- Zwei Funktionen mit vier Eingängen

## Neuere Xilinx Funktionsblöcke



CLB XC3000-Familie

# Vergleich der Realisierungsarten

	PLA/ROM	Vollkunden-spezifische ICs	Gate-array, Sea-of-Gates	FPGA
<b>Entwicklung</b>	billig	sehr teuer	mittel	mittel
<b>Herstellung</b>	mittel	teuer	mittel	sehr billig
<b>Komplexität</b>	schlecht	sehr hoch	mittel	mittel/schlecht
<b>Laufzeiten</b>	schnell	sehr schnell	mittel	mittel



veraltet

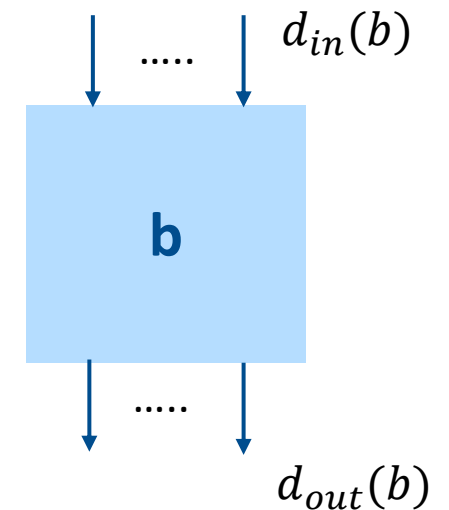
# Ein Schaltkreismodell

Idee: „gerichteter Graph mit einigen zusätzlichen Eigenschaften“

## Definition (Bibliothek):

Sei  $Bib$  eine kombinatorische Bibliothek und  $b \in Bib$ . Dann bezeichnen wir mit  $d_{in}(b)$  den Eingangsgrad von  $b$ , mit  $d_{out}(b)$  den Ausgangsgrad von  $b$ . Die Funktion  $g_b: \mathbb{B}^{d_{in}(b)} \rightarrow \mathbb{B}^{d_{out}(b)}$  beschreibt die zum Bibliothekselement  $b$  gehörende Boolesche Funktion.

Standardbibliothek:  $Bib = \{\neg, \vee, \wedge, NAND, NOR, \oplus, EQUIV\}$



# Schaltkreis über *Bib*

## Definition (Schaltkreis):

Ein Schaltkreis  $SK$  über  $Bib$  ist definiert als Sextupel  $SK_{Bib} = (X_n, Y_m, G, typ, in, out)$ , wobei

- $X_n = \{X_1, X_2, \dots, X_n\}$  Eingabesignale,  $Y_m = \{Y_1, Y_2, \dots, Y_m\}$  Ausgabesignale
- $G = (V, E)$  ein zykelfreier, gerichteter Graph
  - $V = \{0,1\} \cup X_n \cup Y_m \cup S \cup M, S \cap M = \emptyset$ 
    - $S$  Menge an Signalknoten mit einstelliger Eingabe, d.h.  $\forall s \in S: \deg_{in}(s) = 1$
    - $M$  Menge an Modulnoten
  - $E \subset V \times V$
- $typ: M \rightarrow Bib$  ; eine Zuordnung von Modulnoten auf Bibliothekselemente
- $in: V \setminus (S \cup X_n) \rightarrow S^*$  mit  $in(v) \in S^{\deg_{in}(v)}$  ; eine Zuordnung zur Angabe von eingehenden Signalen für Modulnoten und Ausgabesignale
- $out: V \setminus (S \cup Y_m) \rightarrow S^*$  mit  $out(v) \in S^{\deg_{out}(v)}$  ; eine Zuordnung zur Angabe von ausgehenden Signalen für Modulnoten und Eingabesignale

## Schaltkreis - Beispiel

$Bib = \{NOT, OR, AND, NAND, NOR, EXOR, EQUIV\}$

$$X_3 = \{x_1, x_2, x_3\}$$

$$Y_2 = \{y_1, y_2\}$$

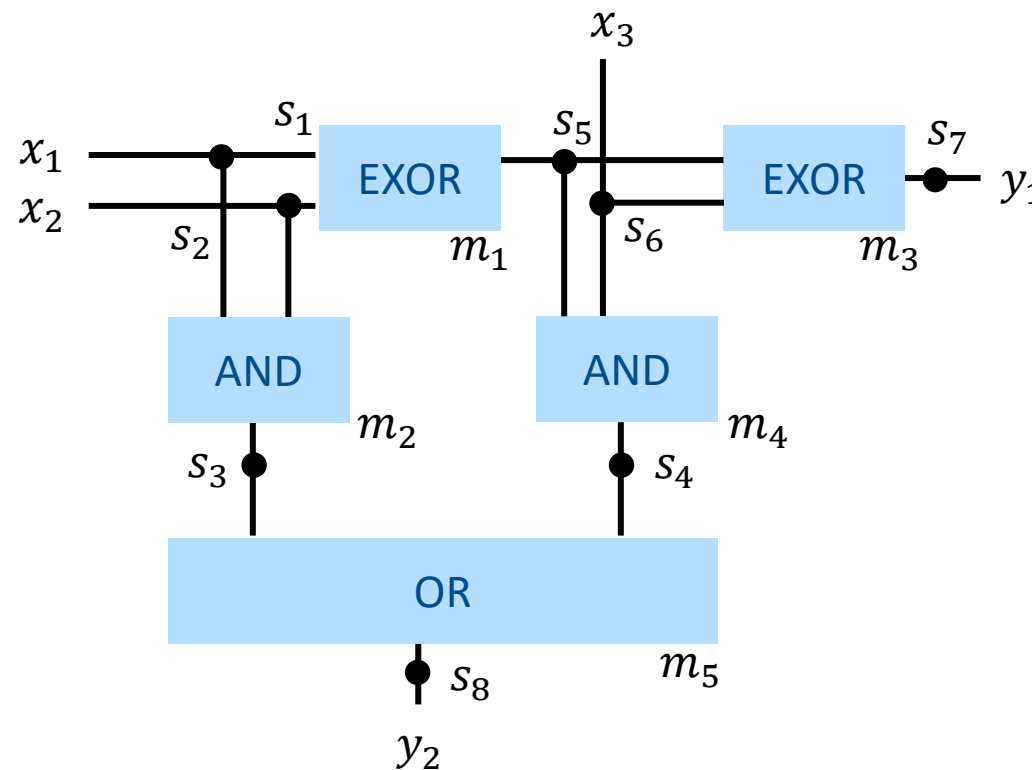
$$G = (V, E)$$

$$V = X_3 \cup Y_2 \cup M \cup S$$

$$S = \{s_1, s_2, s_3, s_4, s_5, s_6, s_7, s_8\}$$

$$M = \{m_1, m_2, m_3, m_4, m_5\}$$

$$E = \{(x_1, s_1), (x_2, s_2), (x_3, s_6), (s_1, m_1), (s_2, m_1), (m_1, s_5), (s_2, m_2), (s_1, m_2), (m_2, s_3), (s_5, m_3), (s_6, m_3), (m_3, s_7), (s_5, m_4), (s_6, m_4), (m_4, s_4), (s_3, m_5), (s_4, m_5), (m_5, s_8), (s_8, y_2), (s_7, y_1)\}$$



$$SK_{Bib} = (X_n, Y_m, G, typ, in, out)$$

$$typ(m_1) = EXOR = typ(m_3)$$

$$typ(m_2) = AND = typ(m_4)$$

$$typ(m_5) = OR$$

$$in(m_1) = in(m_2) = (s_1, s_2)$$

$$in(m_3) = in(m_4) = (s_5, s_6)$$

$$in(m_5) = (s_3, s_4)$$

$$in(y_1) = s_7$$

$$in(y_2) = s_8$$

$$out(m_1) = s_5, out(m_2) = s_3$$

$$out(m_3) = s_7, out(m_4) = s_4$$

$$out(m_5) = s_8$$

$$out(x_1) = s_1, out(x_2) = s_2$$

$$out(x_3) = s_6$$

# Kosten eines Schaltkreises über *Bib*

## Definition(Hardware-Kosten):

Die Hardware-Kosten eines Schaltkreises  $SK$  über *Bib* ist definiert als die Anzahl der Modulnoten im Schaltkreis, d.h.  
 $C(SK_{Bib}) = |M|$ .

## Definition(Schaltkreis-Tiefe):

Die Schaltkreis-Tiefe  $depth(SK_{Bib})$  eines Schaltkreises  $SK$  über *Bib* ist definiert als die Anzahl der Modulnoten auf dem längsten Pfad von einem Eingangssignal zu einem Ausgangssignal.



## Schaltkreis - Beispiel

$$Bib = \{NOT, OR, AND, NAND, NOR, \oplus, EQUIV\}$$

$$X_3 = \{x_1, x_2, x_3\}$$

$$Y_2 = \{y_1, y_2\}$$

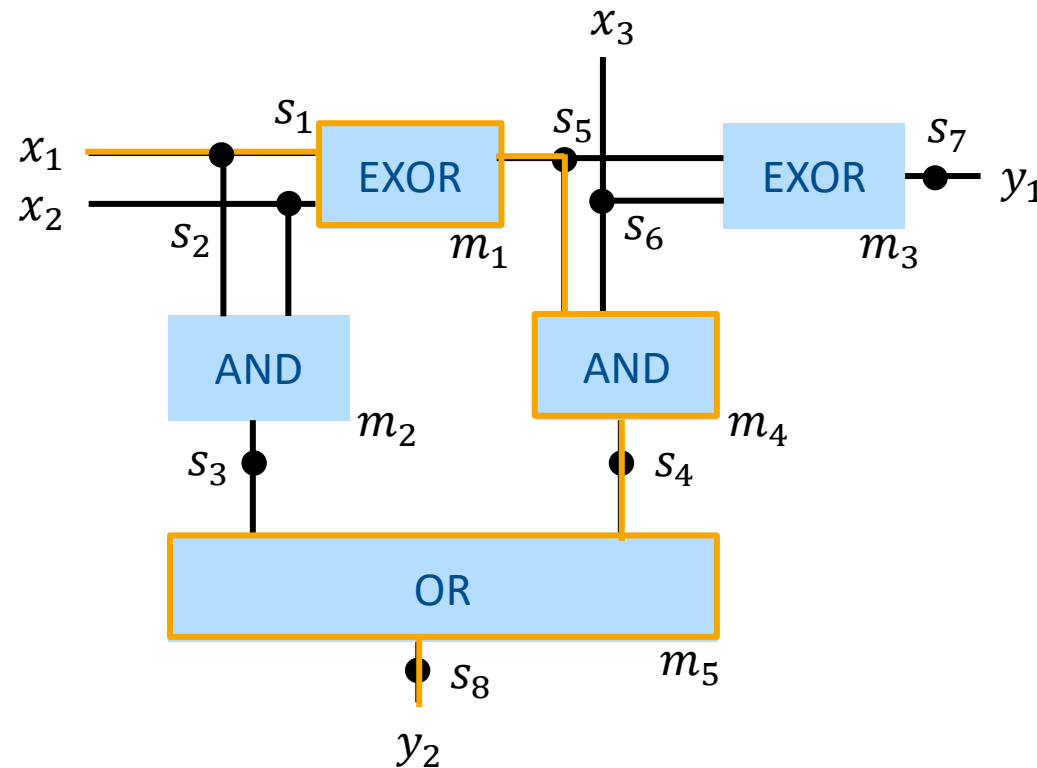
$$G = (V, E)$$

$$V = X_3 \cup Y_2 \cup M \cup S$$

$$S = \{s_1, s_2, s_3, s_4, s_5, s_6, s_7, s_8\}$$

$$M = \{m_1, m_2, m_3, m_4, m_5\}$$

$$E = \{(x_1, s_1), (x_2, s_2), (x_3, s_6), (s_1, m_1), (s_2, m_1), (m_1, s_5), (s_2, m_2), (s_1, m_2), (m_2, s_3), (s_5, m_3), (s_6, m_3), (m_3, s_7), (s_5, m_4), (s_6, m_4), (m_4, s_4), (s_3, m_5), (s_4, m_5), (m_5, s_8), (s_8, y_2), (s_7, y_1)\}$$



$$C(SK_{Bib}) = ?$$

$$C(SK_{Bib}) = |M| = 5$$

$$depth(SK_{Bib}) = ?$$

$$depth(SK_{Bib}) = |\pi(x_1, y_2)|_M = 3$$

$$SK_{Bib} = (X_n, Y_m, G, typ, in, out)$$

## Semantik von Schaltkreisen (1)

### Definition (Simulation einer Belegung):

Sei  $\alpha \in \mathbb{B}^n$  mit  $\alpha = (\alpha_1, \dots, \alpha_n)$ . Für einen Schaltkreis  $SK_{Bib}$  ist die Einsetzung  $\Phi_\alpha: X_n \cup S \cup Y_m \cup \{0,1\} = V \setminus M \rightarrow \mathbb{B}$  von  $\alpha$  wie folgt induktiv definiert:

- $\Phi(X_i) = \alpha_i$
- $\Phi(0) = 0$
- $\Phi(1) = 1$
- $\Phi_\alpha(s) = \Phi(x_i)$ , wenn  $s = out(x_i)$
- $\Phi_\alpha(y_j) = \Phi_\alpha(s)$ , wenn  $s = in(y_j)$
- $(\Phi_\alpha(s'_1), \dots, \Phi_\alpha(s'_l)) = g_{typ(m)}(\Phi_\alpha(s_1), \dots, \Phi_\alpha(s_k))$ , wenn  $out(m) = (s'_1, \dots, s'_l)$  und  $in(m) = (s_1, \dots, s_k)$

Die Berechnung von  $\Phi_\alpha(SK_{Bib})$  heißt auch Simulation von  $\alpha$  für  $SK_{Bib}$ .

## Semantik von Schaltkreisen (2)

### Definition (Simulation einer Booleschen Funktion):

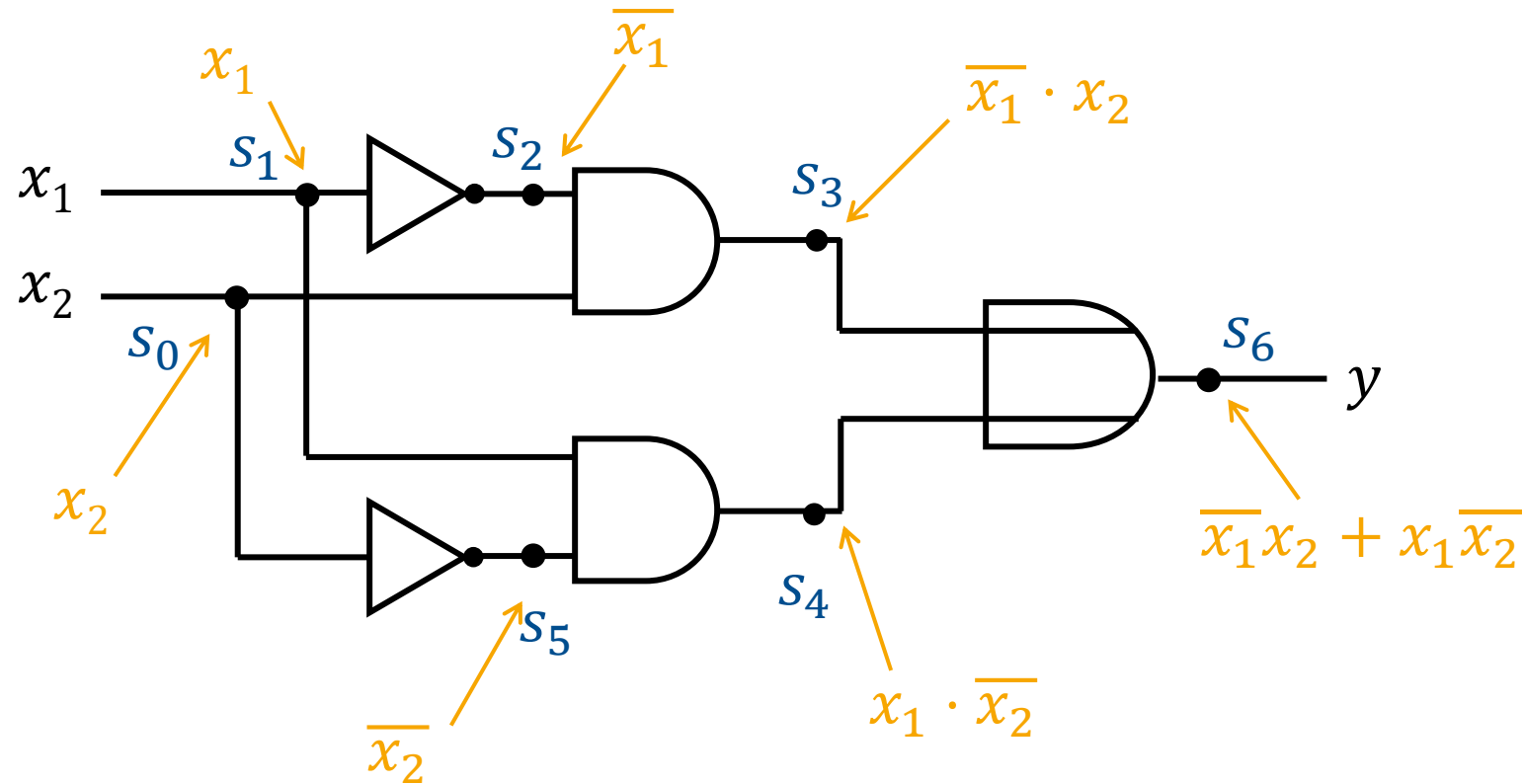
An einem Signal  $s \in S$ , einem Eingangssignal  $X_i \in X_n$  oder Ausgangssignal  $Y_j \in Y_m$  eines Schaltkreises  $SK_{Bib}$  berechnete Boolesche Funktion  $\Psi(s)$  ist definiert als  $\Psi(s)(\alpha) = \Phi_\alpha(s)$  für eine beliebige Boolesche Belegung  $\alpha = (\alpha_1, \dots, \alpha_n) \in \mathbb{B}^n$ .

Die durch den Schaltkreis berechnete Boolesche Funktion ist  $f_{SK_{Bib}} = (\Psi(y_1), \Psi(y_2), \dots, \Psi(y_m))$ .

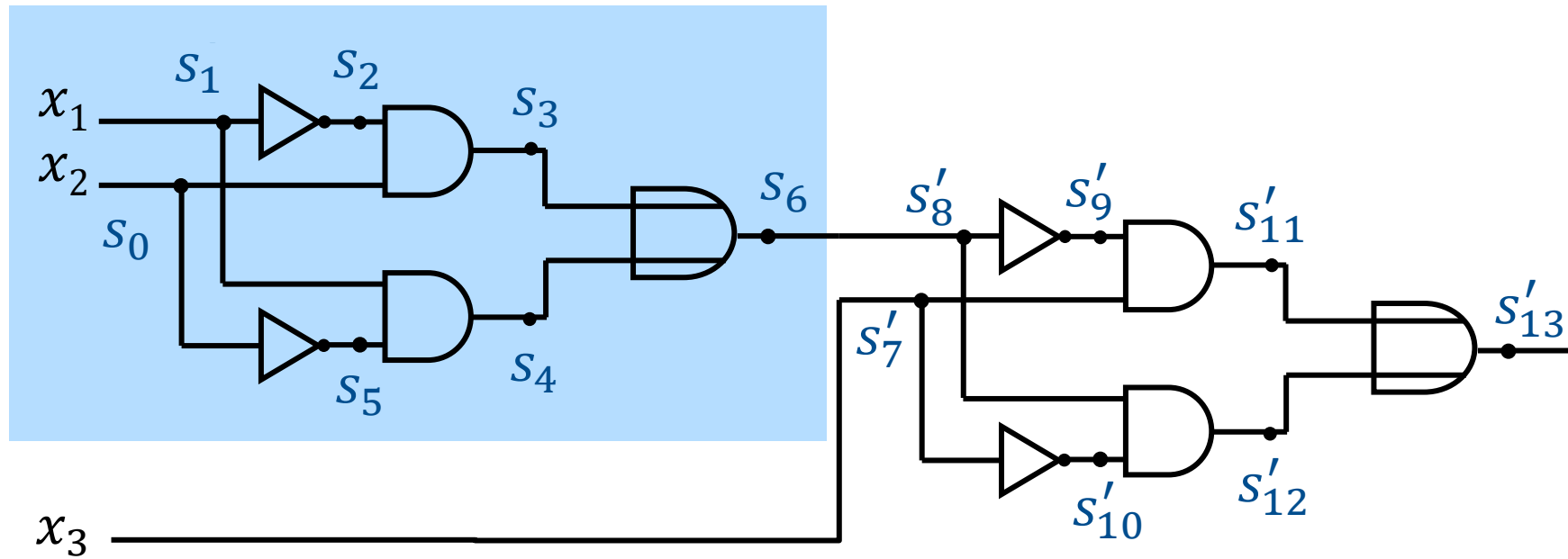
# Symbolische Simulation

- Nutzung Boolescher Variablen zur Simulation eines Schaltkreises (anstelle von festen Booleschen Werten)
- Berechnung der Booleschen Funktion zu jedem Signal in Form
  - eines Booleschen Ausdrucks
  - eines BDDs

## Beispiel



## Teilschaltkreise



## Teilschaltkreis

### Definition (Teilschaltkreis von $SK$ ):

Seien  $SK_{Bib} = (X_n, Y_m, G = (V, E), typ, in, out,)$  und  $SK'_{Bib} = (X'_n, Y'_m, G' = (V', E'), typ', in', out')$  zwei Schaltkreise.  $SK_{Bib}$  heißt Teilschaltkreis von  $SK'_{Bib}$ , falls es eine injektive Abbildung  $\theta: V \setminus (X_n \cup Y_m) \rightarrow V'$  gibt, die die folgenden Bedingungen erfüllt:

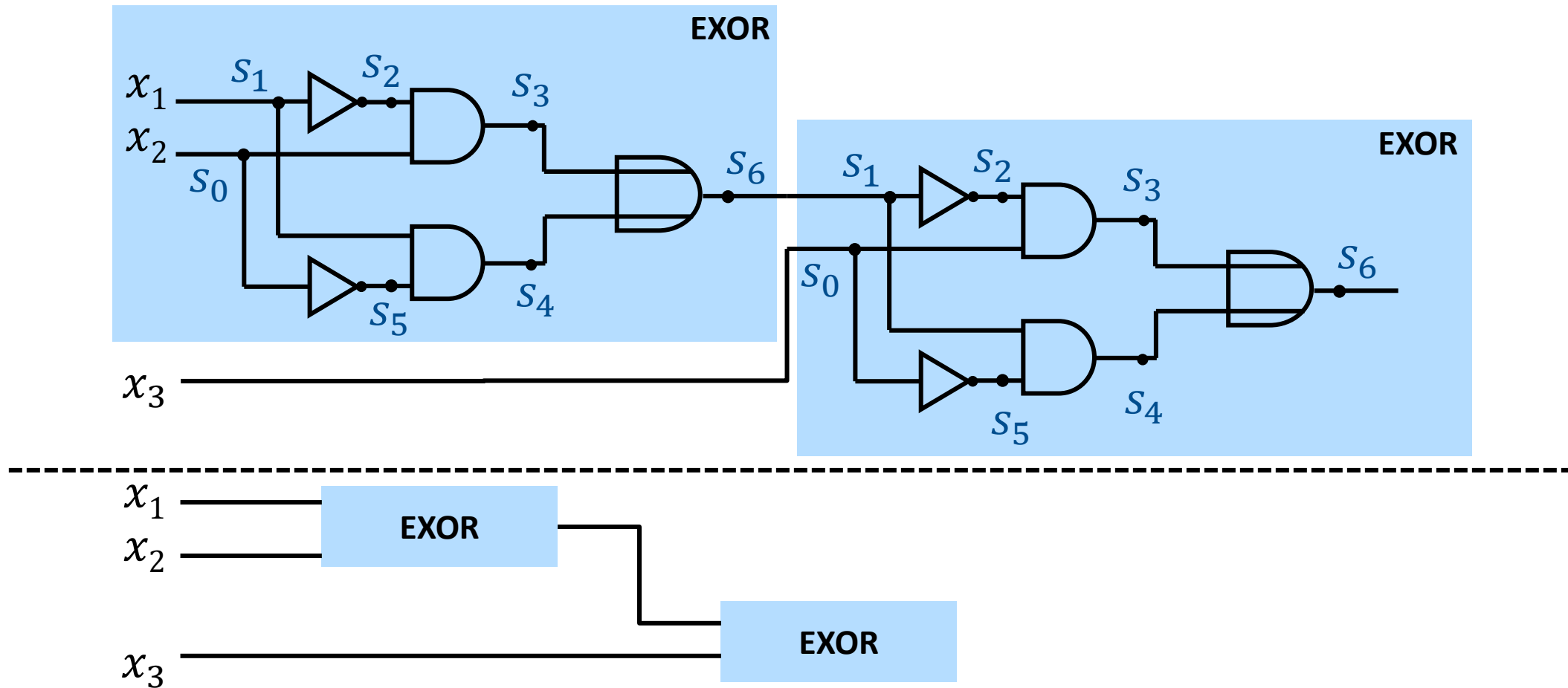
- $\theta(0) = 0', \theta(1) = 1'$
- $typ(m) = typ'(\theta(m)) \forall m \in M$
- $\theta(in(m)) = in'(\theta(m)) \forall m \in M$
- $\theta(out(m)) = out'(\theta(m)) \forall m \in M$
- $\theta(s_1, \dots, s_k) = (\theta(s_1), \dots, \theta(s_k))$  mit  $(s_1, \dots, s_k) \in S^*$

## Hierarchische Schaltkreise

- Ersetzung von Teilschaltkreisen durch Symbole
- Flacher Schaltkreis entsteht durch Ersetzen der Symbole durch Einsetzen der Teilschaltkreise



## Hierarchische Teilschaltkreise - Beispiel



# Boolesche Funktionen und Schaltkreise

**Satz:**

Sei  $f \in \mathcal{B}_{n,m}$ . Dann gibt es einen Schaltkreis, der  $f$  berechnet.

Beweis folgt.

## Rückbesinnung auf Boolesche Ausdrücke

### Lemma (Boolesche Ausdrücke):

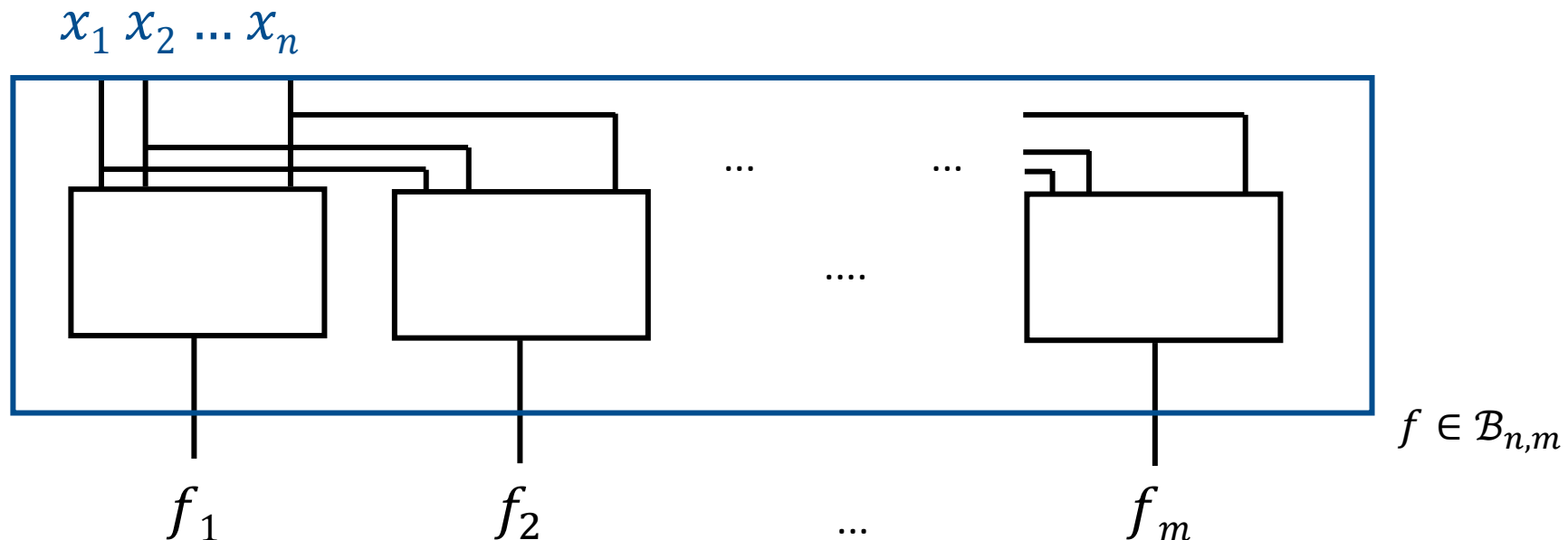
Zu jedem Booleschen Ausdruck  $e \in BE(X_n)$  gibt es einen Schaltkreis  $SK_{Bib} = (X_n, Y_m, G, typ, in, out)$ , sodass  $\Psi(e) = f_{SK_{Bib}}$ .

### Beweis:

Induktiv über die Struktur von Booleschen Ausdrücken.

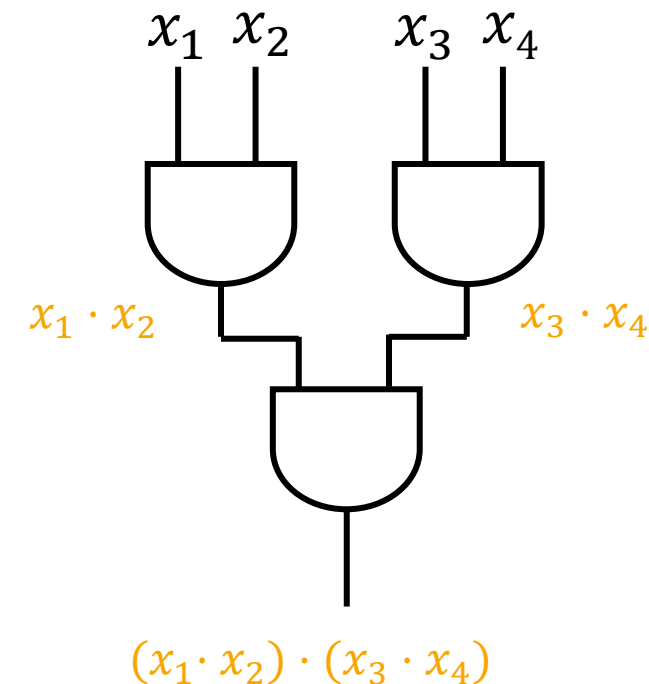
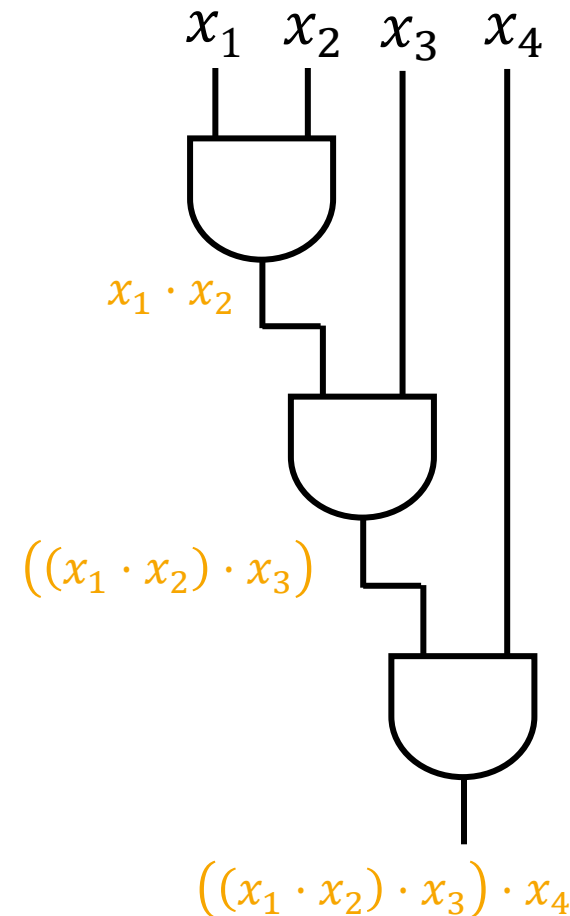
## Beweis zum Satz über Boolesche Funktionen

- **Fall 1:** Sei  $f \in \mathcal{B}_n$ . Dann gibt es einen Booleschen Ausdruck  $e \in BE(X_n)$ , der  $f$  beschreibt. Aus dem vorherigen Lemma folgt dann die Behauptung.
- **Fall 2:** Sei  $f \in \mathcal{B}_{n,m}$ ,  $m \geq 2$ . Fasse  $f: \mathbb{B}^n \rightarrow \mathbb{B}^m$  als Folge von Funktionen  $f = (f_1, \dots, f_m)$ ,  $f_i: \mathbb{B}^n \rightarrow \mathbb{B}$  auf. Konstruiere für jedes  $f_i$  einen  $SK_{Bib}$ , der  $f_i$  berechnet. Setze die Teilfunktionen zusammen.



## Ausnutzung von Gesetzmäßigkeiten (Assoziativität)

- **Annahme:** Realisiere die Funktion  $f(x_1, \dots, x_n) = x_1 x_2 \dots x_n$ 
  - unter Verwendung von *AND*-Gattern mit zwei Eingängen, also  $Bib = \{AND\}$  und  $d^{in}(AND) = 2$ ,
  - mit möglichst geringer Tiefe
- Lösung: Benutze die Assoziativität von „AND“ und nutze balancierte Bäume



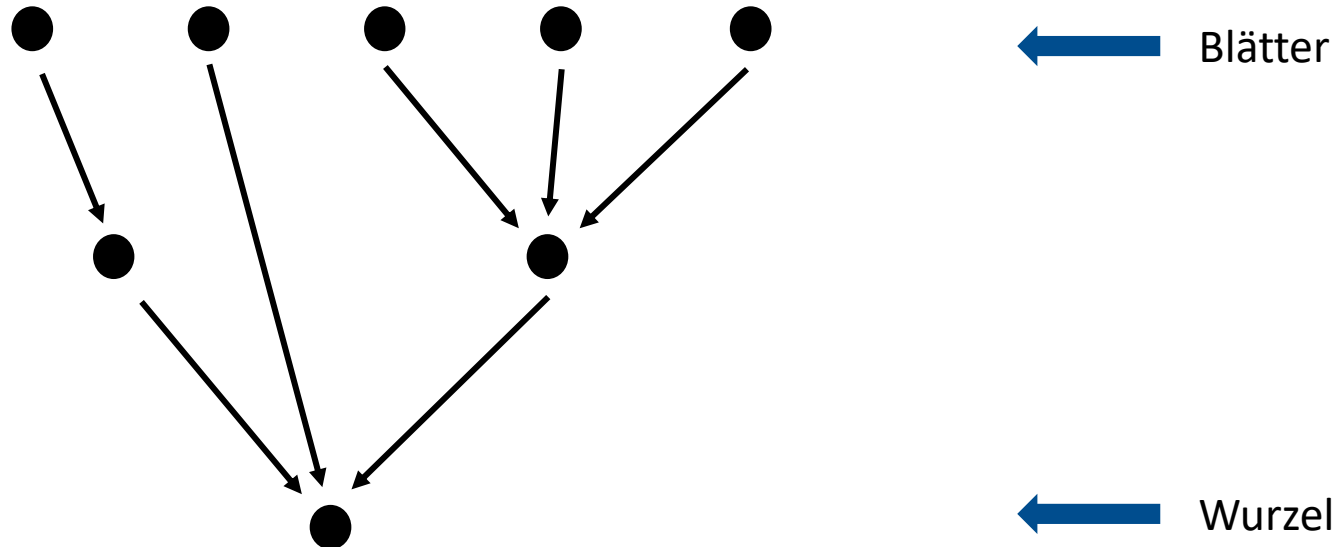
# Definition

## Definition ((Out-)Baum):

Ein (Out-)Baum ist ein gerichteter, zyklfreier, zusammenhängender Graph  $B = (V, E)$ , wobei

- $\deg_{out}(v) \leq 1 \ \forall v \in V$  und
- $\exists! v \in V: \deg_{out}(v) = 0$  - dieser Knoten heißt Wurzel

Die Quellen eines (Out-)Baumes sind die Knoten  $v$  mit  $\deg_{in}(v) = 0$ . Diese Knoten heißen auch Blätter.



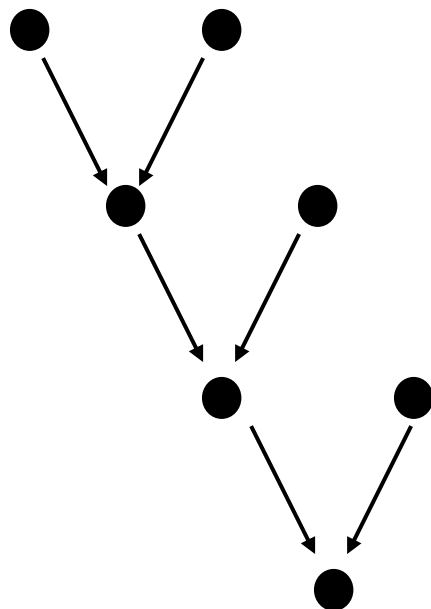
## Bemerkung:

Analog zum Out-Baum kann man In-Bäume definieren. Dreht man die Orientierung im Beispiel um, erhält man einen (In-)Baum.

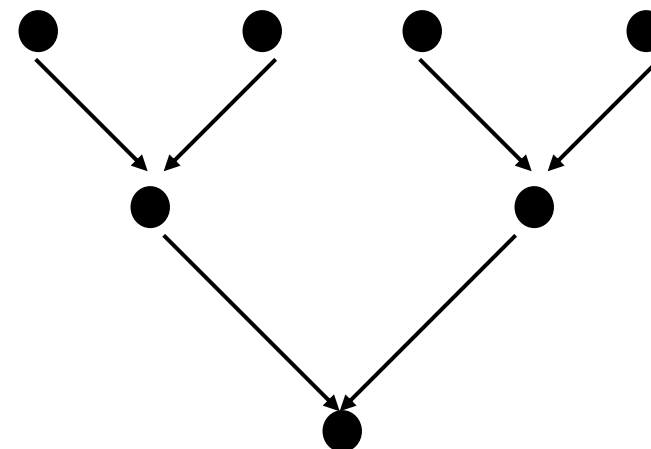
## Definition

### Definition (binärer (Out-)Baum):

Ein (Out-) Baum  $B = (V, E)$  heißt binär, falls  $\deg_{in}(v) = 2 \forall v \in V, v$  kein Blatt.



Tiefe 3



Tiefe 2

# Binäre Bäume

Lemma:

Für jedes  $n \in \mathbb{N}$  gibt es einen binären Baum  $B_n$  mit  $n$  Blättern und Tiefe  $\lceil \log_2 n \rceil$ .

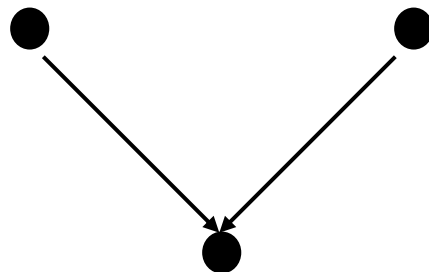
Beweis:

Induktion über  $n$ .

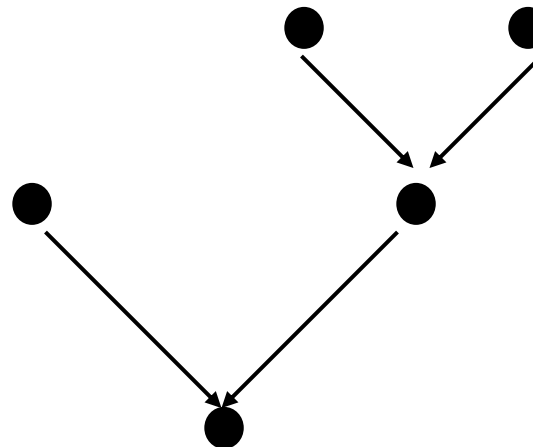
$n = 1$



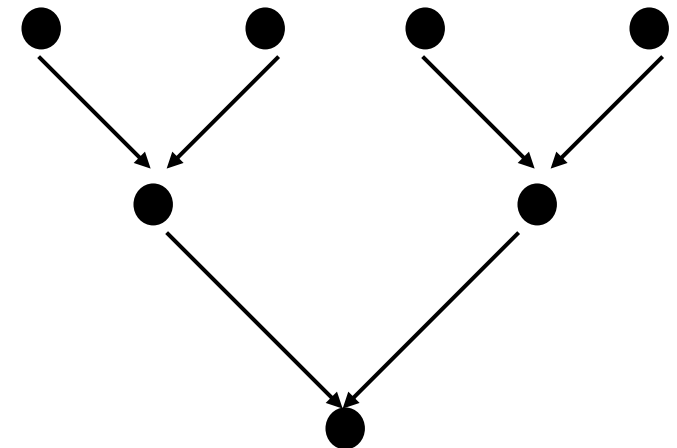
$n = 2$



$n = 3$



$n = 4$





# Realisierung der vollständigen Konjunktion

**Lemma:**

Die vollständige Konjunktion  $f = x_1 x_2 \dots x_n$  kann unter Verwendung von AND-Gattern mit zwei Eingängen durch einen Schaltkreis der Tiefe  $\lceil \log_2 n \rceil$  realisiert werden.

## Zusammenhänge zwischen BEs, BDDs und Schaltkreisen

- Definiert man die Kosten eines BEs durch die Zahl der Operationen, so folgt aus dem Beweis des Satzes, dass es zu jedem BE einen Schaltkreis gibt, dessen Kosten höchstens gleich groß sind.  
Durch Wiederverwendung von Teilschaltkreisen können die Kosten reduziert werden.
- Die Kosten einer Schaltkreisrealisierung für beliebiges  $f$  können mit dem Satz durch  $n + n2^n$ , die Tiefe durch  $n + \lceil \log_2 n \rceil + 1$  abgeschätzt werden.
- BDDs sind „spezielle Schaltkreise“, die den Vorteil der Kanonizität bieten und trotzdem noch effiziente Operationen und oftmals kleine Repräsentationen ermöglichen.