## Übersicht

- Bisher in Teil 1:
  - Rechner im Überblick
  - Befehlssatz, Pipelining
  - Speicherhierarchie
  - Parallelverarbeitung
- Bisher in Teil 2:
  - Darstellung von Zeichen, Zahlen
- ab jetzt:
  - „Zusammensetzung" der Zeichen und „Berechnung" von neuen Zeichen
  - Realisierung von Komponenten



# Kapitel 7

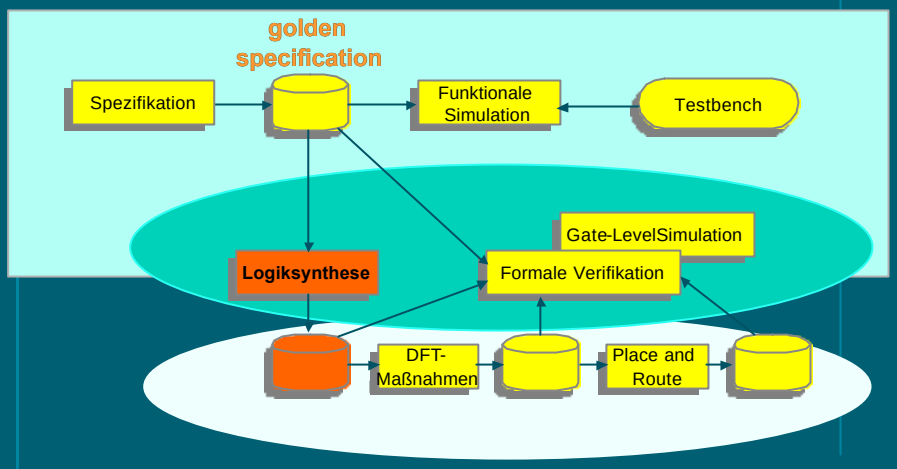
# Der Boolesche Kalkül

## 7.1 Boolesche Funktionen

## 7.2 Boolesche Algebren

## 7.3 Boolesche Ausdrücke

# Ein kurzer Blick auf den Designflow






# Logiksynthese und -minimierung

Versuch, Logik der Schaltung

- zu generieren, bzw.
- zu minimieren

Unterscheidung zwischen

- kombinatorischer Synthese
  = Realisierung einer Booleschen Funktion
- sequentieller Synthese
  = Realisierung eines endlichen Automaten

## Verfügbare Technologien

Zur Auswahl stehende 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)

Konkurrierende Optimierungsziele

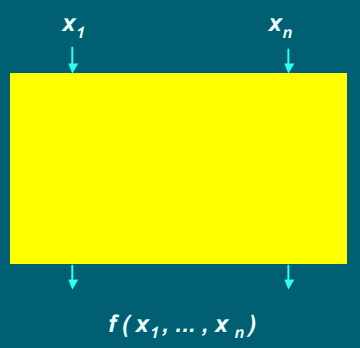
- belegte Fläche
  - Ausbeute
- benötigte Reaktionszeiten
  - Korrektheit des Gesamtsystems
- Testbarkeit
  - Korrektheit des verkauften Chips
- Leistungsverbrauch
  - neue Märkte (Notebooks,...)
- Entwicklungszeit und -kosten
  - time-to-market
  - Wirtschaftskraft der Kunden



## Ausgangspunkt der Überlegungen

$$\mathbf{B} := \{0,1\}$$

## Wir wollen






## Lampen-Beispiel:

- Lampe leuchtet/leuchtet nicht in Abhängigkeit von 4 Schaltern
- Modellierung des Problems
- Steuerung durch Schaltkreis
- Gibt es gute/schlechte Steuerungen ?

## Lampen-Beispiel genauer:

- Zustände der Lampe:
  brennt („1")
  brennt nicht („0")
- Schalter $x_1$ , $x_2$ , $x_3$ , $x_4$
  mit je zwei Stellungen „0" und „1"
- Grundzustand:
  alle Schalter in Stellung 0 ⇔ Lampe brennt nicht
- Genau ein Schalter ändert seine Stellung

  ⇒ Lampe ändert ihren Zustand



## Modellierung

| $x_1$ | $x_2$ | $x_3$ | $x_4$ | Zustand der Lampe |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 0 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 0 | 1 | 1 |
| 1 | 0 | 1 | 1 | 1 |
| 0 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 0 |

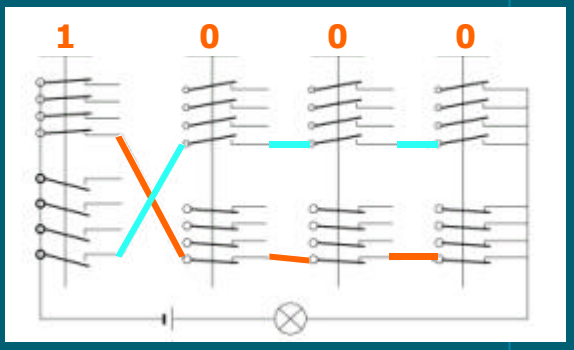
# 1. Realisierung

es gibt 8 Kombination, die zum Brennen der Lampe führen

deshalb 8 Kontakte pro Schalter,

pro Kombination, die zum Brennen der Lampe führt, Realisiere einen „leitenden Weg"

| $x_1$ | $x_2$ | $x_3$ | $x_4$ | Zustand der Lampe |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 0 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 0 | 1 | 1 |
| 1 | 0 | 1 | 1 | 1 |
| 0 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 0 |






# 1. Realisierung

- Bei k Schaltern:

  $k \cdot 2^k$ Kontakte

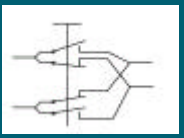
- Beobachtung:

  Lampe brennt genau dann, wenn ungerade viele Schalter auf 1

## 2. Realisierung

Neuer Baustein

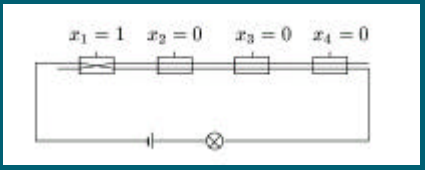
Realisierung





## 2. Realisierung

Neuer Schaltkreis




Kosten bei k Schaltern: $4 \cdot k$ Kontakte

## Fazit

Viele Fragen z.B. bzgl.

- Modellierung
- Realisierung
- Bausteintypen
- Kosten



## 7.1 Boolesche Funktionen

- Eine Abbildung $f : \mathbf{B}^n \to \mathbf{B}^m$ heißt (vollständig definierte) **Boolesche Funktion** in n Variablen.
- $\mathcal{B}_{n,m} := \{f \mid f : \mathbf{B}^n \to \mathbf{B}^m\}$
- Eine Abbildung $f : D \to \mathbf{B}^m$ mit $D \subseteq \mathbf{B}^n$ heisst (partielle) **Boolesche Funktion** in n Variablen.
- $\mathcal{B}_{n,m}(D) := \{f \mid f : D \to \mathbf{B}^m\}$ für $D \subseteq \mathbf{B}^n$

## ON(f ), OFF(f )

- Ist m =1, so bezeichnet
  - ON(f ) die **Erfüllbarkeitsmenge** von f,

    d.h. $ON(f):=\{\alpha\in\{0,1\}^n \mid f(\alpha)=1\}$
  - OFF(f ) die **Nichterfüllbarkeitsmenge** von f,

    d.h. $OFF(f):=\{\alpha\in\{0,1\}^n \mid f(\alpha)=0\}$



## def(f ), DC(f )

- Ist m =1 und f eine partielle Boolesche Funktion

  in n Variablen, so bezeichnet
  - $def(f):=\{\alpha\in\{0,1\}^n \mid f(\alpha) \text{ definiert}\}$

    den **Definitionsbereich** von f.
  - $DC(f):=\{\alpha\in\{0,1\}^n \mid f(\alpha) \text{ nicht definiert}\}$

    den **don´t care Bereich** von f.

## Beispiele

- Konjunktion AND
- Disjunktion OR
- Exclusiv Oder EXOR
- Negation

- Sind f und g zwei Boolesche Funktionen, dann heisst f **kleiner gleich** g (in Zeichen: $f \leq g$) genau dann, wenn $f(\alpha) \leq g(\alpha)\ \forall \alpha \in \{0,1\}^n$ gilt.

  Also: OR $\leq$ AND