

Lernzettel

Technische Grundlagen der Informatik (TechGI) - Digitale Systeme

Universität: Technische Universität Berlin
Kurs/Modul: Technische Grundlagen der Informatik (TechGI) - Digitale Systeme
Erstellungsdatum: September 6, 2025



Zielorientierte Lerninhalte, kostenlos!
Entdecke zugeschnittene Materialien für deine Kurse:

<https://study.AllWeCanLearn.com>

Technische Grundlagen der Informatik (TechGI) - Digitale
Systeme

(1) Digitale Systeme

Digitale Systeme arbeiten mit diskreten Zuständen. Typische Zustände sind 0 und 1.

$$z \in \{0, 1\}$$

Wichtige Merkmale:

- Gatter und Schaltkreise setzen logische Funktionen um.
- Informationen werden mit endlicher Wortbreite dargestellt.

(2) Repräsentation und Kenngrößen

- Wortbreite w : Anzahl der Bits pro Datenwort.
- Anzahl möglicher Werte pro Wort: 2^w .
- Taktrate f_{clk} : Anzahl der Schaltungszyklen pro Sekunde.
- Dynamische Leistung:

$$P_{\text{dyn}} \approx \alpha C_{\text{load}} V^2 f_{\text{clk}}$$

(3) Boolesche Algebra und Vereinfachung

Grundoperationen:

$$x \cdot y \quad (\text{AND})$$

$$x + y \quad (\text{OR})$$

$$\neg x \quad (\text{NOT})$$

Identitäten (Beispiele):

$$x \cdot x = x$$

$$x + x = x$$

$$x \cdot (y + z) = x \cdot y + x \cdot z$$

De Morgan-Sätze:

$$\neg(x \cdot y) = \neg x + \neg y$$

$$\neg(x + y) = \neg x \cdot \neg y$$

(4) Schaltnetze und Schaltungsprinzipien

Gatter:

$$z = x \cdot y \quad (\text{AND})$$

$$z = x + y \quad (\text{OR})$$

$$z = \neg x \quad (\text{NOT})$$

$$z = \overline{x \cdot y} \quad (\text{NAND})$$

$$z = x \oplus y \quad (\text{XOR})$$

(5) Flip-Flops und Schaltwerke

D-Flipflop: $Q^+ = D$

$$Q^+ = D$$

SR-Flipflop: $Q^+ = S \vee (\overline{R} \cdot Q)$

$$Q^+ = S \vee (\overline{R} \cdot Q)$$

JK-Flipflop: $Q^+ = J\overline{Q} + \overline{K}Q$

$$Q^+ = J\overline{Q} + \overline{K}Q$$

T-Flipflop: $Q^+ = Q \oplus T$

$$Q^+ = Q \oplus T$$

(6) Arithmetische Schaltungen

Ripple-Carry-Adder: Addition zweier n -Bit-Wörter.

$$S_i = A_i \oplus B_i \oplus C_i$$

$$C_{i+1} = (A_i \wedge B_i) \vee (A_i \wedge C_i) \vee (B_i \wedge C_i)$$

mit $C_0 = 0$.

(7) Programmierbare Logik

LUT-basierte Logik (Look-Up-Table): Eine LUT k -Input hat

$$f : \{0, 1\}^k \rightarrow \{0, 1\}$$

bzw. eine Bitfolge der Länge 2^k als Speicherinhalt.

(8) Entwurfsmethoden und Typische Schaltungen

Minimierung und Vereinfachung: Ziel ist es, eine logische Funktion mit möglichst wenigen Gattern umzusetzen.

$$f(a, b) = ab + a\bar{b} + \bar{a}b$$

$$\Rightarrow f(a, b) = a + b$$

Hinweis zu Designmethoden:

- Schaltnetze entwerfen, danach Schaltwerke bestimmen.
- Kleingruppendarstellungen (Karnaugh-Karten) zur Minimierung verwenden.