

Lernzettel

Kombinatorische Schaltungen: Entwurfsmethoden, Realisierung und Optimierung

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

Lernzettel: Kombinatorische Schaltungen: Entwurfsmethoden, Realisierung und Optimierung

(1) Begriffsklärung. Eine kombinatorische Schaltung ist ein Schaltnetz, dessen Ausgangswerte ausschließlich von den aktuellen Eingaben abhängen, ohne Speicher (kein Flipflop). Typische Bausteine: Gatter wie AND, OR, NOT, NAND, NOR, XOR, XNOR. Boolesche Funktionen werden meist in Form von

$$F(A, B, \dots) = A \cdot B + \overline{C}$$

mit den Operatoren $\cdot$ (AND), $+$ (OR) und $'$ (NOT) dargestellt. Wichtige Konzepte:

- Wahrheitstabelle, Minterms ($m(i)$) und Maxterms ($M(i)$).
- Normalformen: Summe der Produkte (SOP), Produkt der Summen (POS).
- Vereinfachung durch Boolesche Algebra und Karnaugh-Maps.

(2) Entwurfsmethoden. - Top-Down-Ansatz: Funktionsumfang wird schrittweise in Teilfunktionen zerlegt; Teillösungen werden zusammengeführt. - Shannon-Entwicklung:

$$F = xF_x + \overline{x}F_{\overline{x}},$$

wobei F_x die Teilfunktion ist, wenn $x = 1$. - Boolesche Algebra und Minimierung: Umformungen, Identitäten, De Morgan. - Karnaugh-Karten (2-, 3-, 4-Variablen) zur Minimierung der logischen Ausdrücke; Ziel: Minimalform mit möglichst wenigen Gattern. - Quine–McCluskey-Verfahren (algorithmische Minimierung) für größere Funktionen. - Entwurf auf Gatterebene vs. strukturierter Entwurf (Modularisierung, Wiederverwendung von Teillösungen).

(3) Realisierung. - Gatterebene: Umsetzung der Booleschen Funktionen durch Gatter wie AND, OR, NOT; Minimierung reduziert die Gate-Anzahl und Tiefe. - Universelle Gatter: NAND- oder NOR-Gatter ermöglichen komplette Realisierung einzelner Funktionen. - Implementierung mit Schaltnetzen: Bei kombinatorischen Schaltungen keine Speicherelemente (Flipflops). - Zeitverhalten: Tiefe des Schaltplans bestimmt den kritischen Pfad und die Verzögerung; tiefe Netzwerke sind langsamer, aber oft kleiner. - Praktische Aspekte: Layout, Fan-out, Überschneidungen, Kopplung und Grenzparameter.

(4) Optimierung. - Zielgrößen: Minimale Gatteranzahl, geringe Tiefe (Verzögerung), niedriger Stromverbrauch. - Methoden: Weiterführung der Karnaugh-Minimierung, Nutzung von geteilten Unterausdrücken, NAND/NOR-Implementierung, Negationsverlagerung (De Morgan). - Typische Strategien: - Eliminieren redundanter NOT-Gatter durch direkte Negation der Ausdrücke. - Strukturieren des Netzes, um Mehrfachverwendung von Unterausdrücken zu ermöglichen. - Balance der Tiefe durch parallele Pfade, wenn möglich.

(5) Beispielaufgabe. Gegeben sei die Funktion $F(A, B, C) = AB + C$. Entwerfe eine minimale Realisierung. Vorschlag 1 (Gatterebene): - AND-Gatter für AB , OR-Gatter mit AB und C ergibt F . - Gatteranzahl: 2 (1 AND, 1 OR). Vorschlag 2 (NAND-NAND-Implementierung): $\overline{F} = \overline{AB + C} = (\overline{A} + \overline{B}) \cdot \overline{C}$; realisiert mit NANDs entsprechend. Hinweis: Je nach verfügbaren Gattern kann eine NAND-NAND-Variante bevorzugt werden.

(6) Übungsfragen. - Warum ist die Minimierung der Gate-Anzahl in der Praxis oft wichtiger als die theoretische Minimalform? - Zeige mit einer Karnaugh-Karte, wie man $F = A\bar{B} + \bar{A}C$ minimiert. - Welche Auswirkungen hat die Gate-Tiefe auf das Timing eines digitalen Systems?

(7) Glossar und Formeln.

- Boolesche Operatoren: AND ($\cdot$ oder direkte Nebenstelle), OR ($+$), NOT ($'$) .
- Min-/Maxterms: Minterm $m(i)$ entspricht einem Eintrag der Wahrheitstabelle, Maxterm $M(i)$ der negierte Eintrag.
- De Morgan: $\overline{A \cdot B} = \bar{A} + \bar{B}$, $\overline{A + B} = \bar{A} \cdot \bar{B}$.
- Karnaugh-Map: grafische Minimierung durch Gruppierung von Einsen in benachbartem Bezugsraum.