PROFESSOR ERKLÄRBÄR

Serie Informatik · Folge 4 · Advanced

INFORMATIK

LEHRERINFO ZUM VIDEO

Boolesche Algebra und Schaltungen

FACH	STUFE	DAUER	VIDEO
Informatik	Gymnasium · Sek II	14 min	youtube.com/@Erklaerbaer

01 Lernziele

Nach dem Video können die Schülerinnen und Schüler …

- die Grundgatter AND, OR, NOT, NAND, NOR, XOR und XNOR mit Tabelle, Term und Symbol angeben,
- Terme mit den Gesetzen der booleschen Algebra umformen und vereinfachen (inklusive De Morgan),
- aus einer Wahrheitstabelle die disjunktive Normalform aufstellen und vereinfachen,
- begründen, warum NAND und NOR universell sind,
- Volladdierer und 4-Bit-Addierer aufbauen und den Addierer zum Subtrahieren nutzen,
- Schaltnetz und Schaltwerk unterscheiden und das Flipflop als 1-Bit-Speicher erklären.

02 Voraussetzungen

UND, ODER, NICHT, XOR, Wahrheitstabellen und Halbaddierer (Folge 4 · Grundlagen); Zweierkomplement (Folge 2 · Advanced). Wo ALU und Register im Prozessor sitzen, zeigt die Advanced-Folge 5.

03 Ablauf des Videos

ZEIT	KAPITEL	INHALT UND BEISPIEL
0:00	Einstieg	Apollo Guidance Computer: ≈ 2800 Chips mit je zwei NOR-Gattern · Ident · Fahrplan
0:35	Was du brauchst	Grundgatter, 2^3 Zeilen, Halbaddierer, Zweierkomplement · Hinweis auf 04G
1:10	Alle Gatter	Schreibweisen A·B, A + B, $\bar{A}$ · 1 + 1 = 1 · NAND, NOR, XNOR · $A \oplus B = A\cdot\bar{B} + \bar{A}\cdot B$

2:13	Rechengesetze	Kommutativ, assoziativ, beide Distributivgesetze, A + A = A, Komplement · De Morgan mit Tabellenbeweis · A·B + A·$\bar{B}$ = A PAUSE-FRAGE (A + B)(A + $\bar{B}$) = A · Merksatz
3:58	Von der Tabelle zur Schaltung	Mehrheitsschaltung · Minterme · DNF · vereinfacht AB + AC + BC · KV-Diagramm
5:07	NAND kann alles	NICHT, UND, ODER aus NAND · universell · CMOS-NAND aus 4 Transistoren · Redstone PAUSE-FRAGE 1 und 2 NAND · Merksatz
6:23	Addieren mit Gattern	Volladdierer · Ripple-Carry · 0101 + 0110 = 1011 · Subtraktion mit XOR-Reihe: 7 − 3 = 4
7:57	Schaltwerke	SR-Latch aus zwei NOR · Entprellen · D-Flipflop · Register, Zähler, Befehlszähler · Merksatz
9:17	Projekt-Baustein	ALU der Mini-CPU: 8 Volladdierer + XOR-Reihe · Register A und B
9:40	Prüfungsfallen	fünf Fallen (siehe 04)
10:41	Zusammenfassung	Kernaussagen · zurück zum Apollo-Computer
11:17	Quiz	4 Fragen · Endkarte Folge 5 Advanced

04 Prüfungsfallen

- A + A = 2A – in der booleschen Algebra heisst «+» ODER; A + A = A.
- ¬(A ∧ B) = ¬A ∧ ¬B – De Morgan dreht den Operator um: ¬A ∨ ¬B.
- Ein Halbaddierer reicht für mehrstellige Zahlen – ab der zweiten Stelle braucht es den Übertrag-Eingang (Volladdierer).
- Ein Flipflop ist ein Schaltnetz – durch die Rückkopplung hängt der Ausgang vom Zustand ab (Schaltwerk).
- Die disjunktive Normalform ist schon die einfachste Form – sie ist korrekt, aber meist vereinfachbar.

05 Einsatz im Unterricht

Einstieg: «Wie viele verschiedene Gattertypen braucht ein Computer?» – Vermutungen sammeln, dann Einstieg schauen.

Simulator: Volladdierer und SR-Latch in Logisim Evolution oder CircuitVerse bauen und testen (kostenlos).

Beweisen: De Morgan und das zweite Distributivgesetz in Partnerarbeit per Tabelle beweisen.

Prüfungsvorbereitung: Aufgaben 2, 3 und 5 sind im Stil schriftlicher Prüfungen gestellt.

Hinweis: Die Angabe zum Apollo Guidance Computer bezieht sich auf Block II (rund 2800 integrierte Schaltungen mit je zwei NOR-Gattern mit drei Eingängen). Das SR-Latch ist ohne Takt dargestellt; der verbotene Zustand S = R = 1 wird im Video nicht behandelt (Aufgabe 6).

06 Quellen

- George Boole: „An Investigation of the Laws of Thought“ (1854) · Augustus De Morgan: „Formal Logic“ (1847)
- Claude E. Shannon: „A Symbolic Analysis of Relay and Switching Circuits“ (1938)
- Eldon C. Hall: „Journey to the Moon: The History of the Apollo Guidance Computer“, AIAA (1996)
- David Harris, Sarah Harris: „Digital Design and Computer Architecture“, 2. Aufl., Morgan Kaufmann (2012)
- IEC 60617-12: Graphische Symbole für Schaltpläne – binäre Elemente
- Gesellschaft für Informatik: Bildungsstandards Informatik für die Sekundarstufe II (2016)

NAME _____ KLASSE _____ DATUM _____

Boolesche Algebra und Schaltungen

1 Gatter und Terme ●○○

Gib für NAND, NOR und XNOR je die Wahrheitstabelle und einen Term mit UND, ODER, NICHT **an**.

2 Vereinfachen ●●○

Vereinfache schrittweise und nenne jeweils das verwendete Gesetz: a) $A{\cdot}B + A{\cdot}\bar{B} + \bar{A}{\cdot}B$ b) $\neg(\neg A \vee B)$ c) $A + \bar{A}{\cdot}B$

3 Von der Tabelle zur Schaltung ●●●

Eine Anlage soll melden, wenn ***genau einer*** der drei Sensoren A, B, C auslöst. **Stelle** die Wahrheitstabelle auf, **bestimme** die DNF und **begründe** mit dem KV-Diagramm, ob sie sich vereinfachen lässt. **Vergleiche** mit $A \oplus B \oplus C$.

4 NAND universell ●●○

Zeige mit De Morgan, dass sich ODER aus drei NAND-Gattern bauen lässt, und **skizziere** die Schaltung.

5 Addierer

a) **Berechne** mit einem 4-Bit-Ripple-Carry-Addierer 0111 + 0101 und gib alle Überträge an. b) **Berechne** 6 − 2 mit XOR-Reihe und $C_{in} = 1$ und **begründe**, warum der letzte Übertrag ignoriert wird.

6 SR-Latch

Untersuche, was beim SR-Latch aus zwei NOR passiert, wenn S = R = 1 anliegt und danach beide gleichzeitig auf 0 gehen. **Begründe**, warum man diesen Zustand vermeidet.

Lösungen

1 Gatter und Terme

NAND: 1, 1, 1, 0 = ¬(A·B) = Ā + B̄ · NOR: 1, 0, 0, 0 = ¬(A + B) = Ā·B̄ · XNOR: 1, 0, 0, 1 = A·B + Ā·B̄ (Zeilen 00, 01, 10, 11).

2 Vereinfachen

a) A·B + A·B̄ = A (Distributiv, Komplement) → A + Ā·B = A + B (zweites Distributivgesetz: (A + Ā)(A + B)). b) De Morgan: ¬¬A ∧ ¬B = A·B̄. c) A + Ā·B = (A + Ā)·(A + B) = A + B.

3 Von der Tabelle zur Schaltung

Y = 1 nur für 001, 010, 100 → DNF: Ā·B̄·C + Ā·B·C̄ + A·B̄·C̄. Im KV-Diagramm liegen die drei Einsen nicht nebeneinander, die DNF lässt sich nicht weiter zusammenfassen. A ⊕ B ⊕ C ist zusätzlich bei 111 gleich 1 – also nicht dasselbe.

4 NAND universell

A + B = ¬(¬A · ¬B) (De Morgan). ¬A = NAND(A, A), ¬B = NAND(B, B), dann NAND(¬A, ¬B) = A + B → drei NAND-Gatter.

5 Addierer

a) 0111 + 0101: Stelle 0: 1 + 1 = 0, Ü 1; Stelle 1: 1 + 0 + 1 = 0, Ü 1; Stelle 2: 1 + 1 + 1 = 1, Ü 1; Stelle 3: 0 + 0 + 1 = 1, Ü 0 → 1100 = 12. b) 6 − 2: 0110 + 1101 + 1 = 1 0100 → 0100 = 4; im festen 4-Bit-Format ist der Übertrag aus der höchsten Stelle kein Ergebnisbit (Zweierkomplement, 02A).

6 SR-Latch

Bei S = R = 1 sind beide NOR-Ausgänge 0 – Q und Q̄ sind nicht mehr komplementär. Gehen beide gleichzeitig auf 0, hängt der Endzustand vom schnelleren Gatter ab (Wettlauf) – das Verhalten ist unbestimmt. Darum gilt S = R = 1 als verbotener Zustand.

Q Quiz im Video

1 A (X̄ · Ȳ) · 2 B (NAND und NOR) · 3 B (drei Eingänge) · 4 C (Rückkopplung, speichert einen Zustand)