DS 15 – Grundgatter AND, OR, XOR, NOT und Belegungstabellen

Informatik 10 – MINT-Zweig | Informatiksysteme

Lernziel

Ich kann die logischen Grundgatter benennen und skizzieren und die zugehörigen Belegungstabellen angeben.

🕘 Stundenverlaufsplan (Doppelstunde, 90 Minuten)

Orientierungsrahmen für 11 SuS: Die ersten 80 Minuten bilden den Kern. Die letzten 10 Minuten sind bewusst flexibel und werden nicht mit neuem Pflichtstoff belegt.

Phase	Zeit	Inhalt/Aktion	Sozialform
Einstieg und Aktivierung	10'	Vermutungen sammeln, wie ein Computer mit den Zuständen 0 und 1 Entscheidungen realisiert.	Plenum; kurze Einzeldenkzeit
Fragend-entwickelnde Erarbeitung	25'	AND, OR, XOR und NOT aus Alltagssituationen und Schüleraussagen entwickeln.	Unterrichtsgespräch; Beiträge der SuS werden sichtbar festgehalten
Gemeinsame Umsetzung und Sicherung	20'	Belegungstabellen gemeinsam mit Handzeichen oder Signalkarten aufbauen.	Plenum; Live-Coding, Tafelbild oder gemeinsame Simulation
Kurzer Transfer	15'	Grundgatter in CircuitVerse bauen und systematisch testen.	Einzelarbeit mit anschließendem Tandemabgleich (4 Tandems + 1 Trio)
Auswertung	10'	IEC- und ANSI-Symbole zuordnen und Tabellen mit Fachsprache erläutern.	Plenum; Lösung erklären und Fachsprache sichern
Reserve / Vertiefung	10'	Majority-Schaltung oder Python-Kontrolle als Vertiefung.	Flexibel; bei Bedarf für Fragen, Fehlersuche oder Abschluss nutzen

🗣️ Sprachhilfe

- „Für die Eingangswerte … liefert die Schaltung den Ausgangswert …“
- „Der Term/die Tabelle repräsentiert die Schaltung, weil …“
- „Die Rückkopplung bzw. Gruppierung bewirkt …“

⚠️ Wichtiger Hinweis zur Gatter-Symbolik in CircuitVerse

In deutschen Lehrbüchern werden Logikgatter meist nach **IEC 60617** dargestellt: rechteckige Symbole mit Kennbuchstaben, z. B. `&` für AND, `≥1` für OR, `=1` für XOR und ein kleiner Kreis am Ausgang für die Negation.

CircuitVerse verwendet stattdessen die in den USA gebräuchliche Symbolik (ANSI/IEEE Std 91) – also die klassischen, unterschiedlich geformten Gattersymbole (AND = "D-Form", OR = spitz zulaufende Kurvenform, XOR = OR-Form mit zusätzlicher Linie, NOT = Dreieck mit kleinem Kreis am Ausgang).

Funktional sind beide Notationen identisch – nur die Zeichnung unterscheidet sich. Achtet trotzdem darauf, in Klassenarbeiten/Heftaufschrieben weiterhin die IEC-Symbole zu verwenden, wie sie im Schulbuch stehen. In CircuitVerse geht es uns nur um das **Verhalten** der Schaltungen, nicht um die Prüfungsschreibweise.

📌 Merksätze

Gatter	IEC-Symbol	Verhalten
AND (Konjunktion)	`&`	Ausgang ist `1`, wenn **alle** Eingänge `1` sind
OR (Disjunktion)	`≥1`	Ausgang ist `1`, wenn **mindestens ein** Eingang `1` ist
XOR (Exklusiv-Oder)	`=1`	Ausgang ist `1`, wenn die Eingänge **unterschiedlich** sind
NOT (Negation)	kleiner Kreis	Ausgang ist das **Gegenteil** des Eingangs

- Eine **Belegungstabelle** (Wahrheitstabelle) listet für jede mögliche Kombination von Eingangswerten den zugehörigen Ausgangswert auf.
- Bei n Eingängen gibt es 2^n mögliche Belegungen.

✏️ Zusatzübung – Grundgatter zusätzlich von Hand skizzieren

*Hinweis: „Skizzieren" bedeutet laut Lehrplan-Operatorendefinition, einen Sachverhalt aufs Wesentliche reduziert **grafisch oder sprachlich** darzustellen – das Bauen der Gatter in CircuitVerse gleich im Praxisteil unten erfüllt diese Kompetenz bereits. Die folgende Papierübung ist daher freiwillig, aber eine gute zusätzliche Übung, z. B. als Vorbereitung auf eine Klassenarbeit ohne Rechner.*

Zeichnet **freiwillig** (auf Papier oder in dieser Zelle als Beschreibung) die vier Grundgatter AND, OR, XOR und NOT **mit ihren IEC-Symbolen** (siehe Tabelle oben) und beschriftet jeweils Ein- und Ausgänge mit `A`, `B` bzw. `Q`.

🖥️ Praxisteil: CircuitVerse

1. Öffnet circuitverse.org/simulator (Login optional, aber empfohlen zum Speichern).
2. Im linken Bereich unter **Input** findet ihr **Input**-Bausteine (Schalter) – zieht zwei davon ins Arbeitsfeld.
3. Unter **Output** findet ihr die **Output**-LED – zieht eine davon ins Arbeitsfeld.
4. Unter **Gates** findet ihr AND, OR, XOR, NOT – zieht ein Gatter eurer Wahl dazwischen.
5. Verbindet die Bausteine per Drag&Drop von Anschluss zu Anschluss mit Leitungen.
6. Klickt auf die Input-Schalter, um zwischen `0` und `1` umzuschalten, und beobachtet die Output-LED (leuchtet = `1`).

📝 Übung 1 – Alle vier Grundgatter testen

Baut in CircuitVerse **nacheinander** je eine Schaltung mit AND, OR, XOR und NOT (bei NOT reicht ein Eingang). Testet **alle** Eingangskombinationen und übertragt eure Beobachtungen in die Tabellen unten (Doppelklick zum Bearbeiten dieser Zelle).

AND

A	B	Ausgang
0	0	
0	1	
1	0	
1	1	

OR

A	B	Ausgang
0	0	
0	1	
1	0	
1	1	

XOR

A	B	Ausgang
0	0	
0	1	
1	0	
1	1	

🐍 Kontrolle mit Python

Nutzt die Zelle unten, um eure handschriftlich ermittelten Belegungstabellen zu überprüfen. Python bringt die logischen Operatoren `and`, `or` bereits mit; XOR entspricht `!=` (ungleich) bei Wahrheitswerten.

```python
from itertools import product

def zeige_belegungstabelle(gattername, funktion, anzahl_eingaenge=2):
    print(f"--- {gattername} ---")
    for eingaben in product([0, 1], repeat=anzahl_eingaenge):
        ausgang = funktion(*eingaben)
        print(eingaben, "->", int(ausgang))

zeige_belegungstabelle("AND", lambda a, b: a and b)
zeige_belegungstabelle("OR", lambda a, b: a or b)
zeige_belegungstabelle("XOR", lambda a, b: a != b)
zeige_belegungstabelle("NOT", lambda a: not a, anzahl_eingaenge=1)
```

```
--- AND ---
(0, 0) -> 0
(0, 1) -> 0
(1, 0) -> 0
(1, 1) -> 1
--- OR ---
(0, 0) -> 0
(0, 1) -> 1
(1, 0) -> 1
(1, 1) -> 1
--- XOR ---
(0, 0) -> 0
(0, 1) -> 1
(1, 0) -> 1
(1, 1) -> 0
--- NOT ---
(0,) -> 1
(1,) -> 0
```

📝 Übung 2 – Kombinierte Schaltung

Baut in CircuitVerse eine Schaltung mit **drei Eingängen** A, B, C, die genau dann `1` ausgibt, wenn **mindestens zwei** der drei Eingänge `1` sind (Mehrheitsentscheidung/"Majority-Gatter"). Nutzt dafür mehrere AND- und OR-Gatter.

Tipp: Die Schaltung lässt sich als `(A AND B) OR (B AND C) OR (A AND C)` beschreiben.

✅ Kontrolle Übung 2 mit Python

```python
def majority(a, b, c):
    return (a and b) or (b and c) or (a and c)

zeige_belegungstabelle("Majority", majority, anzahl_eingaenge=3)
```

```
--- Majority ---
(0, 0, 0) -> 0
(0, 0, 1) -> 0
(0, 1, 0) -> 0
(0, 1, 1) -> 1
(1, 0, 0) -> 0
(1, 0, 1) -> 1
(1, 1, 0) -> 1
(1, 1, 1) -> 1
```