

## Schaltnetze

### Didaktische Hinweise

#### Didaktische Einordnung

Von Schüler\*innen hört man zwar oft, dass der Rechner mit Nullen und Einsen arbeitet. Der Zusammenhang zwischen einem Programm, das sie in Scratch (s. [5]) oder für den Calliope mini (s. [1]) erstellen, und der Verarbeitung auf der Ebene von Nullen und Einsen, ist aber in der Regel unklar. Gerade einfache Probleme, die ohne Variablen unter Verwendung der Sensoren und Aktoren des Calliope mini gelöst werden können, lassen sich jedoch häufig auch als Schaltnetz umsetzen, so dass hier eine direkte Anknüpfung möglich ist.

Da das Themengebiet der technischen Informatik im niedersächsischen Kerncurriculum für die Oberstufe (s. [7]) keinen Platz gefunden hat, bietet es sich an, entsprechende Vorstellungen zu den Vorgängen im Rechner in der Sekundarstufe I zu entwickeln, wenn hier ausreichend Zeit und ein entsprechendes Interesse der Schüler\*innen vorhanden sind.

Weiterhin bietet sich eine Verknüpfung mit dem Modul „Aufbau von Computersystemen“ aus dem Lernfeld *Computerkompetenz* an (s. [6]).

#### Zielgruppe und Voraussetzungen

Die vorliegenden Materialien richten sich vor allem an Schüler\*innen der Sekundarstufe I in den Jahrgängen 9 und 10. Im Kerncurriculum für die Sekundarstufe I (s. [6]) ist die Konstruktion von Schaltnetzen als Ergänzung zur Implementierung von Algorithmen für Sensor-Aktor Systeme, wie dem Calliope mini (s. [1]) oder Robotern, vorgesehen.

Sinnvoll sind außerdem Vorkenntnisse im Bereich Codierung, um einen Zusammenhang zwischen der Verarbeitung auf der Ebene der Nullen und Einsen und der Verarbeitung komplexerer Daten herstellen zu können. Ggf. können entsprechende Aspekte der Codierung auch erst in diesem Kontext thematisiert werden.

#### Werkzeuge

##### Simulationssoftware

Um die konstruierten Schaltungen testen zu können, kommt in den Aufgaben eine Simulationssoftware zum Einsatz. Die Aufgabenstellungen sind unabhängig von einem konkreten Werkzeug formuliert. Die Musterlösungen in den Lehrermaterialien werden als Schaltungen für den Digitalsimulator angeboten (s. [2]). Der Einsatz dieses Simulationsprogramms hat sich im Unterricht bewährt, da es sehr intuitiv zu bedienen und relativ übersichtlich aufgebaut ist. Die Software wird jedoch leider nicht mehr weiterentwickelt und steht nicht für alle Betriebssysteme zur Verfügung. Ein vergleichbares Werkzeug wäre z. B. LogicSim. Die Verdrahtung ist hier in der ursprünglichen Version (s. [8]) nicht so komfortabel, wurde aber in der Nachfolgeversion (s. [3]) verbessert. Vorteil ist, dass sich eigene Schaltungen in Bausteinen (Modulen) zusammenfassen lassen.

##### Konstruktion realer Schaltungen

Darüber hinaus ist die Realisierung der Schaltungen auf Steckbrettern mit echten Bauteilen möglich. Hier muss zusätzlich zu den logischen Verknüpfungen die Stromversorgung der Bauteile sichergestellt

werden. Um bei der Verdrahtung nicht den Überblick zu verlieren, sollten einfache Beispiele mit wenigen Bauteilen gewählt werden. Neben falscher Verdrahtung kommt hier als Fehlerquelle ein defektes Bauteil hinzu. Daher sollten die Schüler\*innen zunächst lernen, wie sie die Bauteile testen können. Steht im Unterricht ausreichend Zeit zur Verfügung, kann der Produktstolz der Schüler\*innen bei einer realen Schaltung ungleich höher sein als bei einer simulierten Schaltung.

## Lernziele und didaktische Hinweise

### Leitfaden Schaltnetze

Im niedersächsischen Kerncurriculum Informatik für die Sek I kann das vorliegende Material der Entwicklung der Kompetenz „Die Schülerinnen und Schüler entwerfen und implementieren Schaltnetze unter Verwendung von Logikbausteinen.“ zugeordnet werden. Diese ist im Modul „technische Realisierung automatisierter Prozesse“ im Lernfeld *Automatisierte Prozesse* verortet (s. [6]).

Grundlage der technischen Schaltungen sind die logischen Verknüpfungen UND, ODER sowie NICHT. Diese sind den Schüler\*innen eventuell schon aus der Algorithmik bekannt. Dort werden sie jedoch häufig eher intuitiv und an die Alltagssprache angelehnt eingesetzt. Die Identifikation von *wahr* mit 1 und *falsch* mit 0 sowie die entsprechende Auswertung logischer Verknüpfungen wird daher einiger Übung bedürfen.

Im Rahmen des Leitfadens verwenden die Schüler\*innen die Darstellungsformen *Schalterm*, *Wahrheitstabelle* und *Schaltung aus Schaltsymbolen* und lernen, wie sie die Darstellungsformen systematisch ineinander transformieren können. Da bei den Aufgaben nach Möglichkeit ein Kontextbezug zu Systemen aus der Lebenswelt der Schüler\*innen hergestellt wird, ist dabei vor allem der Weg von der Wahrheitstabelle zum Schalterm und schließlich zur Schaltung von Bedeutung. Die Wahrheitstabelle ergibt sich aus dem Kontext der Aufgabe. Bei den Einstiegsaufgaben lässt sich auch der passende Schalterm direkt aus der Aufgabe ableiten. Bei den zunehmend komplexeren Aufgaben wird dies nur Schüler\*innen gelingen, die gerne logisch denken und knobeln. Die Schüler\*innen lernen daher auch, wie sie einen Schalterm in der disjunktiven Normalform (DNF) systematisch und nach einem klaren Rezept aus der Wahrheitstabelle herleiten können. Da die Schaltterme in der DNF lang werden können, wird zur Vereinfachung der Schaltterme die Regel  $(x_1 \wedge x_2 \wedge \dots \wedge x_n) \vee (\bar{x}_1 \wedge x_2 \wedge \dots \wedge x_n) = (x_2 \wedge \dots \wedge x_n)$  eingeführt. Diese können sich die Schüler\*innen im Rahmen von Aufgabe 22 selbständig durch logische Überlegungen erschließen. Die Schüler\*innen werden die Regel dabei eher umgangssprachlich als mathematisch formulieren.

Als Hintergrundinformation sei hier ergänzt, dass sich die Regel formal mithilfe der Axiome von Huntington herleiten lässt:

$$(x_1 \wedge x_2 \wedge \dots \wedge x_n) \vee (\bar{x}_1 \wedge x_2 \wedge \dots \wedge x_n) = (x_1 \vee \bar{x}_1) \wedge (x_2 \wedge \dots \wedge x_n) \text{ Distributivgesetz}$$

$$(x_1 \vee \bar{x}_1) \wedge (x_2 \wedge \dots \wedge x_n) = 1 \wedge (x_2 \wedge \dots \wedge x_n) \quad \text{Inverse Elemente } (x_1 \vee \bar{x}_1) = 1$$

$$1 \wedge (x_2 \wedge \dots \wedge x_n) = (x_2 \wedge \dots \wedge x_n) \quad \text{Neutrale Elemente } a \wedge 1 = a$$

Auch die Umsetzung des Schaltterms in DNF in eine Schaltung folgt einer Systematik. In einer ersten Ebene wird für jeden Minterm ein UND-Baustein mit entsprechend vielen Eingängen benötigt. Die Schaltvariablen müssen gemäß dem Schalterm ggf. verneint werden, bevor sie in den UND-Baustein geleitet werden. In einer zweiten Ebene folgt ein ODER-Baustein. Die Anzahl der Eingänge muss hier der Anzahl der Minterme entsprechen. Die Ausgänge der UND-Bausteine (Minterme) werden mit den

Eingängen des ODER-Bausteins verbunden. Der Ausgang des ODER-Bausteins liefert das Ergebnis und kann mit einer Lampe verbunden werden. Insbesondere für leistungsschwächere Schüler\*innen ist dieser Zusammenhang nicht offensichtlich und muss ggf. explizit thematisiert werden.

Den Schüler\*innen wird also mit der DNF und der Vereinfachungsregel zum Zusammenfassen der Minterme ein systematischer Weg von der Wahrheitstabelle zur Schaltfunktion und schließlich zur Schaltung an die Hand gegeben. Dieser Weg ist jedoch als Hilfsangebot zu verstehen, wenn die logischen Zusammenhänge nicht auf Anhieb erkannt werden. Leistungsstarke Schüler\*innen werden durch logisches Argumentieren häufig schneller zum Ziel kommen. Daher wird im Leitfaden auch der alternative Zugang angeboten einen kürzeren Schalterm als den in DNF durch gezielte logische Überlegungen direkt aus der Wahrheitstabelle abzuleiten.

Bei Wahrheitstabellen, die sehr viele Zeilen mit 1en aufweisen, ist das Erzeugen der konjunktiven Normalform (KNF) aus den Maxtermen einfacher. Da sich für die Aufgaben im Leitfaden jedoch keine entsprechenden Wahrheitstabellen ergeben, wird die KNF an dieser Stelle nicht thematisiert. Für interessierte Schüler\*innen liegt eine Erläuterung bei, die bei Bedarf verwendet werden kann. Einen geeigneten Anwendungsfall stellen z. B. die Schaltfunktionen zur Ansteuerung einer Siebensegmentanzeige dar, die im abschließenden Gruppenpuzzle konstruiert werden (s. unten).

Ein allgemeingültiges Konzept der Informatik, das sich im Rahmen der Konstruktion von Schaltungen gut veranschaulichen lässt, ist das Zusammensetzen komplexer Lösungen aus einfacheren Teillösungen. Dazu werden die Teillösungen zur Wiederverwendung in einem Baustein zusammengefasst. So werden hier zunächst nur die Grundbausteine UND-, ODER- und NICHT eingeführt. Für komplexere Schaltungen werden häufig verwendete Schaltungen wie z. B. XOR, Halbaddierer oder Volladdierer zu einem Baustein zusammengefasst. Aus diesen lassen sich dann komplexere Schaltungen wie ein Schaltnetz zur Addition mehrstelliger Binärzahlen konstruieren. Wichtig ist, dass diese zusätzlichen Bausteine immer auf eine Konstruktion aus den Grundbausteinen zurückgeführt werden, damit deutlich wird, dass sich grundsätzlich auch jede komplexe Schaltung aus diesen drei Grundbausteinen aufbauen lässt. Dieser Aspekt kann in entsprechend leistungsstarken Lerngruppen mit dem abschließenden Gruppenpuzzle zur Konstruktion von Standardschaltnetzen, wie dem Multiplexer, dem Demultiplexer und dem Komparator noch vertieft werden. Das Gruppenpuzzle bietet damit auch eine Überleitung zum Aufbau eines vereinfachten Modellrechners.

### Gruppenpuzzle zu Standardschaltungen

In den Expertengruppen werden Multiplexer, Demultiplexer und Komparator zunächst aus Grundgattern konstruiert. Hier kann ggf. die DNF angewendet und das Verfahren gefestigt werden. In den weiterführenden Aufgaben werden größere Varianten der Bausteine, für mehr Ein- bzw. Ausgabe-Bits aus der kleinsten Version des Bausteins konstruiert. Dieser Ansatz greift das Vorgehen zur Konstruktion der Schaltnetzes zur Addition aus Halb- und Volladdierer-Bausteinen auf.

In der Stammgruppe sollen die Ansätze *Konstruktion aus Grundbausteinen mittels DNF* und *Aufbau aus kleineren Bausteinen der gleichen Funktionalität* in Aufgabe 3b verglichen werden. Die Schüler\*innen werden hier vermutlich argumentieren, welche Vorgehensweise ihnen leichter fällt, welche Schaltung übersichtlicher ist oder sich ihrer Ansicht nach leichter konstruieren lässt.

Für reale Schaltungen würden hier eher die Faktoren Schaltzeit und Größe der Schaltung eine Rolle spielen. Die Schaltzeit lässt sich anhand der Anzahl der Bausteine, die ein Signal vom Eingang bis zum Ausgang passieren muss, gut erfassen. Die Größe der Schaltung ist in der Simulation hingegen nicht

ersichtlich. Ein UND-Baustein ist in der Simulation unabhängig von der Anzahl der Eingänge immer gleich groß. In der Realität wächst die entsprechende Schaltung hingegen mit jedem Eingang. Dies wird anschaulich, wenn man sich den UND-Baustein als Kettenschaltung entsprechend vieler Schalter vorstellt. Daher kann man für einen Multiplexer mit vielen Eingängen abschätzen, dass die Schaltung, die der DNF entspricht, zwar schneller ist, aber auch mehr Fläche beansprucht (vgl. [4]). Eine detaillierte Betrachtung führt in der Sek I zu weit. Es kann aber ggf. ein Ausblick erfolgen, dass logisch gleichwertige Schaltungen in der Realisierung durchaus unterschiedliche Vor- und Nachteile haben, die es bei dem Versuch, die Rechenchips für technische Geräte so zu konstruieren, dass sie sowohl klein als auch schnell sind, zu berücksichtigen und gegeneinander abzuwägen gilt.

Eine vierte Expertengruppe beschäftigt sich im Rahmen des Gruppenpuzzles mit der Siebensegmentanzeige. Der Baustein selbst lässt sich wie eine Lampe oder LED in die Kategorie Ausgabe einordnen. Die Schaltung, die benötigt wird, um eine Binärzahl als Dezimalzahl auf der Siebensegmentanzeige anzeigen zu lassen, kann jedoch ebenfalls zu den Standardschaltungen gezählt werden, so dass die Betrachtung in den Kontext des Gruppenpuzzles passt. Der Digitalsimulator stellt auch hierfür einen fertigen Baustein zur Verfügung (s. Abbildung 1).

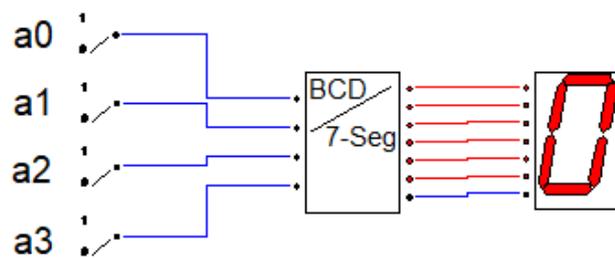


Abbildung 1: Baustein zur Umwandlung von Binärzahlen in 7-Segmentanzeige im Digitalsimulator (s. [2])

Die Konstruktion einer Schaltung zur Anzeige von dreistelligen Binärzahlen als Dezimalzahlen auf der Siebensegmentanzeige kann auch als einzelne, arbeitsteilige Gruppenarbeit umgesetzt werden. Das arbeitsteilige Arbeiten bietet sich hier besonders an, da die Arbeitsschritte zur Konstruktion der Schaltungen für die einzelnen Segmente sehr ähnlich sind. Hier bietet sich einerseits die Verwendung der DNF ohne Vereinfachung an, da die einzelnen Minterme für jedes Segment wiederverwendet werden können. Da hier einige Segmente fast nur Einsen und wenige Nullen in der Wahrheitstabelle aufweisen, bietet sich hier aber auch die Thematisierung der konjunktiven Normalform (KNF) an. Je nachdem wie sicher die Schüler\*innen bereits im Umgang mit der DNF sind, kann der Exkurs zur KNF ergänzend oder als Ersatz für Aufgabe 2 und 3 zur Siebensegmentanzeige eingesetzt werden.

### Ausblick

Betrachtet man einen Rechner von innen, so wird man einige der Funktionen, die hier mithilfe von Schaltnetzen konstruiert wurden, wiederfinden. Etwa die Multiplexer und Demultiplexer in Form des Bussystems oder das Addieren als Teilfunktion der Alu. Um tatsächlich einen einfachen Rechner zu konstruieren, fehlt jedoch die Möglichkeit Daten zu speichern. Dies erfolgt über Bausteine mit Rückkopplung, die Flipflops. Damit wäre dann auch eine technische Realisierung von Mealy-Automaten, mit denen ggf. bereits technische Geräte modelliert wurden, möglich. Mit Schaltnetzen lassen sich nur Mealy-Automaten mit einem Zustand realisieren. Unterschiedliche Zustände bilden das Gedächtnis des Systems und eine Realisierung erfordert daher Speicherbausteine.

Materialien, die einen Einblick in die Zusammenhänge geben, sind in die Arbeit. Diese richten sich jedoch primär an Schüler\*innen der Sekundarstufe II und gehen über das Kerncurriculum für die Sekundarstufe I hinaus.

## Literaturverzeichnis

- [1] Calliope gGmbH (2020). *Calliope*. <https://calliope.cc> [Datum des Zugriffs: 23.03.2020]
- [2] FreeGroup Andreas Herz (2003). DigitalSimulator Version 5.57. [www.FreeGroup.de](http://www.FreeGroup.de)<sup>1</sup>
- [3] Gabriel, P. (2020). LogicSim Version 3.4.  
<https://github.com/codepiet/LogicSim3/blob/master/release/LogicSim-3.4-20200611.zip> [Datum des Zugriffs: 11.11.2020]
- [4] Hoffmann, D. (2007). Grundlagen der Technischen Informatik. München: Hanser Verlag.
- [5] Lifelong-Kindergarten-Group am Media-Lab des MIT. *Scratch*. <https://scratch.mit.edu/> [Datum des Zugriffs: 03.04.2020]
- [6] Niedersächsisches Kultusministerium (2014). *Kerncurriculum für die Schulformen des Sekundarbereichs I Schuljahrgänge 5 – 10. Informatik*. Hannover: Unidruck
- [7] Niedersächsisches Kultusministerium (Hrsg.) (2017) Kerncurriculum für das Gymnasium – gymnasiale Oberstufe, die Gesamtschule – gymnasiale Oberstufe, das Kolleg. Informatik. Hannover: unidruck
- [8] Tetzl, A. (2009). LogicSim Version 2.4. [www.tetzl.de](http://www.tetzl.de)

## Lizenz

Dieses Werk ist lizenziert unter einer [Creative Commons Namensnennung - Nicht kommerziell - Keine Bearbeitungen 4.0 International Lizenz](#). Sie erlaubt Download und Weiterverteilung des vollständigen Werkes unter Nennung meines Namens, jedoch keinerlei Bearbeitung oder kommerzielle Nutzung.

Die beiliegenden Schülermaterialien sind lizenziert unter einer [Creative Commons Namensnennung - Nicht-kommerziell - Weitergabe unter gleichen Bedingungen 4.0 International Lizenz](#). Sie erlaubt Bearbeitungen und Weiterverteilung des Werks unter Nennung meines Namens und unter gleichen Bedingungen, jedoch keinerlei kommerzielle Nutzung.

---

<sup>1</sup> Die Version, die auf dieser Seite verlinkt ist, ist leider veraltet. Der Digitalsimulator in der Version 5.57 ist z. B. unter dem Link <https://www.wintotal.de/download/digital-simulator/> [Datum des Zugriffs: 11.11.2020] erhältlich.