Logikgatter und Schaltnetze

Workshop zu den Grundlagen der Technischen Informatik in der Sekundarstufe II am Gymnasium in Sachsen

Peter Böttcher

27. April 2024

Alexander-von-Humboldt-Gymnasium Greifswald

Vorstellung

Wer bin ich?

Peter Böttcher
Informatik/Mathematik
Greifswald

Wer seid ihr?

Kennenlern-Bingo!
5-10 Minuten

Was wünscht ihr euch?

Was kann ich bieten?

Inhaltsverzeichnis

Motivation

Lehrplan - Sekundarstufe 1

Lernbereich 1: Algorithmen	**11 Ustd.**
Beherrschen der Implementierung der algorithmischen Grundstrukturen	grundlegende, einfache Algorithmen ➔ Kl. 8, LB 1 Syntax und Semantik ➔ LB 2
- Datentypen · Zahlen · Zeichenketten · Wahrheitswerte	
- Variablenzuweisungen	
- verknüpfte Bedingungen	Verkettung durch logische Operatoren
Kennen des Prinzips der Modularisierung	Nutzen von Unterprogrammen und Bibliotheken
Übertragen der Kenntnisse zu Algorithmen auf maschinelle Entscheidungsprozesse	autonomes Fahren, Gesichtserkennung, Wahlcomputer ➔ KL. 9, LB 2 ➔ ETH, Kl. 10, LB 1

Lehrplan - Sekundarstufe 2 - Grundkurs

Lernbereich 1: **Technische Informatik**	**10 Ustd.**
Kennen theoretischer Grundlagen	
- Binär- und Hexadezimalsystem	
- rechnerinterne Zahlenformate	Gleitkommazahlen, Zweierkomplement, Festkommazahlen
- Zeichenkodierung	ASCII-Code, Unicode
- Boolesche Algebra	Wahrheitstabellen, NOT, AND, OR und deren Verknüpfungen
Übertragen der theoretischen Grundlagen auf die Umsetzung in Schaltnetzen	Einsatz von Simulationssoftware
Schaltnetz-Analyse	Schaltnetz-Synthese
Einblick gewinnen in die Herstellung von Mikro-chips	
Übertragen der Kenntnisse zur Rechnerarchitektur auf aktuelle Hardware	Von-Neumann-Rechner, Entwicklungen und Standards beachten
- Prozessoren	Befehlssatz, Taktfrequenz, Energieeffizienz
- Arten von Speichersystemen	optisch, elektromagnetisch und elektronisch
Einblick gewinnen in die nachhaltige Nutzung von Hardware	Nutzungsdauer von Smartphones, Elektroschrott, Reparierbarkeit ⇒ Bildung für nachhaltige Entwicklung → LBW 4

Lehrplan - Sekundarstufe 2 - Leistungskurs

Lernbereich 1: Technische Informatik	18 Ustd.
Kennen theoretischer Grundlagen	
- Binär- und Hexadezimalsystem	
- rechnerinterne Zahlenformate	Gleitkommazahlen
· Zweierkomplement	
· Festkommazahlen	
- Zeichenkodierung	ASCII-Code, Unicode
- Boolesche Algebra	
· NOT, AND, OR und deren Verknüpfungen	
· Wahrheitstabellen	
· Doppelnegation, Kommutativ-, Assoziativ- und Distributivgesetz, De Morgansche Regeln	
· Vereinfachung boolescher Ausdrücke	
Übertragen der theoretischen Grundlagen auf die Umsetzung in Schaltnetzen	Einsatz von Simulationssoftware in praktischen Übungen
- Schaltnetz-Analyse	
- Schaltnetz-Synthese	Umsetzung einer booleschen Funktion in eine Hardware-Schaltung
- Standardschaltnetze Halb- und Volladdierer	
Einblick gewinnen in die Herstellung von Mikrochips	
Übertragen der Kenntnisse zur Rechnerarchitektur auf aktuelle Hardware	Von-Neumann-Rechner, Entwicklungen und Standards beachten
- Prozessoren	Befehlssatz, Taktfrequenz, Energieeffizienz, Multi-Threading

Andere Bundesländer

- Mecklenburg-Vorpommern
- Sachsen-Anhalt
- Bayern
- Folgerung: Sachsen ist vorne dabei (wenn nicht führend)!

Fachliche Annäherung

- Einstieg in Theoretische Informatik
- Grundlage der Aussagenlogik
  - Eines der ältesten Gebiete der Wissenschaft
- Grundlage der Technik und Datenverarbeitung

Hilfsmittel für die SuS und diesen Workshop

Tafelwerk
Auch im Abitur verfügbar!

Boolesche Algebra

Menge ...

Grundlage der Booleschen Algebra

Example

- „Draußen ist gutes Wetter.“
- „Morgen ist Samstag.“

Die beiden Wahrheitswerte „wahr“ und „falsch“, auch „true/false“ oder „1“ und „0.“

Wahrheitswerte kombinieren - Aussagenlogik

Example

- „Draußen ist gutes Wetter oder draußen ist nicht gutes Wetter.“
- „Morgen ist Samstag und morgen ist Donnerstag.“

Die beiden Wahrheitswerte lassen sich nach festen Regeln kombinieren. Dafür gibt es die Operatoren *NICHT*, *UND*, *ODER*.

Aufgabe 1.1

Bestimme den Wahrheitsgehalt der folgenden Aussagen

- „Draußen ist gutes Wetter **und** die Sonne scheint."
- „Draußen ist gutes Wetter **und** es ist bewölkt."
- „Draußen ist schlechtes Wetter **und** die Sonne scheint."
- „Draußen ist schlechtes Wetter **und** es ist bewölkt."
- „Morgen ist Samstag **oder** morgen ist Donnerstag."
- „Morgen ist Dienstag **oder** morgen ist Freitag."
- „Morgen ist Dienstag **oder** heute ist Mittwoch."
- „Morgen ist Donnerstag **oder** heute ist Mittwoch."

erste Aufgaben - Lösung

Verknüpfung mit UND

Teilaussage 1	Teilaussage 2	Ergebnis
wahr	wahr	wahr
wahr	falsch	falsch
falsch	wahr	falsch
falsch	falsch	falsch

Verknüpfung mit ODER

Teilaussage 1	Teilaussage 2	Ergebnis
wahr	wahr	wahr
wahr	falsch	wahr
falsch	wahr	wahr
falsch	falsch	falsch

erste Aufgaben - Lösung

Verknüpfung mit UND

A	B	$A \wedge B$
1	1	1
1	0	0
0	1	0
0	0	0

Verknüpfung mit ODER

A	B	$A \vee B$
1	1	1
1	0	1
0	1	1
0	0	0

Symbole

Boolesche Werte können mit folgenden Verknüpfungen zueinander ins Verhältnis gesetzt werden:

UND

A	B	$A \wedge B$
1	1	1
1	0	0
0	1	0
0	0	0

ODER

A	B	$A \vee B$
1	1	1
1	0	1
0	1	1
0	0	0

NICHT

A	$\neg A$
1	0
0	1

Definition
Eine Funktion $f : A \to B$ heißt **Boolesche Funktion**, wenn $A = \{0,1\}^n$ und $B = \{0,1\}^m$ für ein $n, m \in \mathbb{N}$ gilt.

Ist $m = 1$, so sprechen wir von einer einfachen Booleschen Funktion oder einer Booleschen Funktion mit einem Booleschen Ausgang, ansonsten von einer Booleschen Funktion mit mehreren Booleschen Ausgängen.

Rechnen mit Booleschen Funktionen

Mit einfachen Booleschen Funktionen kann man, ähnlich wie mit ganzen Zahlen, „rechnen“[1].

Wir nehmen uns dafür $a, b \in \{0, 1\}$ und definieren für alle a, b eine

- ODER-Verknüpfung
  $a \vee b = 0 \Leftrightarrow a = b = 0.$
- UND-Verknüpfung
  $a \wedge b = 1 \Leftrightarrow a = b = 1.$
- NEGATION / Komplement
  $\neg a = 1 \Leftrightarrow a = 0.$

[1] Achtung! Technisch gesehen kann man mit Booleschen Funktionen nicht rechnen! Das geht nur mit Booleschen Ausdrücken.

Rechnen mit Booleschen Funktionen

Es gibt Regeln...
$\forall f, g, h : \{0,1\}^n \to \{0,1\}$:

- **Kommutativität**: $f \vee g = g \vee f$, $f \wedge g = g \wedge f$
- **Assoziativität**: $f \vee (g \vee h) = (f \vee g) \vee h, \quad f \wedge (g \wedge h) = (f \wedge g) \wedge h$
- **Distributivität**:
  $(f \vee g) \wedge (f \vee h) = f \vee (g \wedge h), \quad (f \wedge g) \vee (f \wedge h) = f \wedge (g \vee h)$
- **Idempotenz**: $f \vee f = f, f \wedge f = f$

Rechnen mit Booleschen Funktionen

... und noch sooooo viel mehr Regeln.

$\forall f, g, h : \{0,1\}^n \to \{0,1\}$:

- **Absorption**: $f \vee (f \wedge g) = f$, $f \wedge (f \vee g) = f$
- **Regel der 0 und 1**: $f \vee (g \wedge \neg g) = f$, $f \wedge (g \vee \neg g) = f$
- **Doppeltes Komplement**: $\neg(\neg f) = f$
- **de Morgan'sche Regel**: $(\neg f) \wedge (\neg g) = \neg(f \vee g)$,
  $(\neg f) \vee (\neg g) = \neg(f \wedge g)$
- **Consensus-Regel**: $(f \wedge g) \vee (\neg f \wedge h) \vee (g \wedge h) = (f \wedge g) \vee (\neg f \wedge h)$
  $(f \vee g) \wedge (\neg f \vee h) \wedge (g \vee h) = (f \vee g) \wedge (\neg f \vee h)$

Korrektheit beweisen - Teil 1

- **Absorption**: $f \vee (f \wedge g) = f$, $f \wedge (f \vee g) = f$

Logisch: „f oder (f und g)“
Es gilt: $f \wedge g$ ist nur 1, wenn f und g beide 1 sind $\Rightarrow$ Wenn f 1 ist, ist $1 \vee 1 = 1$, ansonsten ist $f \vee 0 = f$.

Wie formalisieren?

Korrektheit beweisen - Teil 1

- **Absorption**: $f \vee (f \wedge g) = f$, $f \wedge (f \vee g) = f$

Wahrheitstabelle!

f	g	$f \wedge g$	$f \vee (f \wedge g)$
0	0		
0	1		
1	0		
1	1		

Korrektheit beweisen - Teil 1

- **Absorption** - $f \vee (f \wedge g) = f$, $f \wedge (f \vee g) = f$

Wahrheitstabelle!

f	g	$f \wedge g$	$f \vee (f \wedge g)$
0	0	0	0
0	1	0	0
1	0	0	1
1	1	1	1

Korrektheit beweisen - Teil 2

Aufgabe 1.2

Beweise die Gültigkeit der de Morgan'schen Regel mithilfe einer Wahrheitstabelle.

de Morgan'sche Regel: $(\neg f) \wedge (\neg g) = \neg(f \vee g)$,

[Bonus: $(\neg f) \vee (\neg g) = \neg(f \wedge g)$]

Korrektheit beweisen - Teil 2

Aufgabe 1.2

Beweise die Gültigkeit der de Morgan'schen Regel mithilfe einer Wahrheitstabelle.

de Morgan'sche Regel: $(\neg f) \land (\neg g) = \neg(f \lor g)$,

[Bonus: $(\neg f) \lor (\neg g) = \neg(f \land g)$]

f	g	$\neg f$	$\neg g$	$f \lor g$	$(\neg f) \land (\neg g)$	$\neg(f \lor g)$
0	0	1	1	0	1	1
0	1	1	0	1	0	0
1	0	0	1	1	0	0
1	1	0	0	1	0	0

Mini-Logik-Test!

https://l.boettcher.schule/l/BoolTest

Vereinbarung: Ab sofort schreiben wir...

... statt „$\vee$“ nur noch „$+$“

... statt „$\wedge$“ nur noch „$\cdot$“

... statt „$\neg x$“ nur noch „$\bar{x}$“

Vereinfachen von Ausdrücken

Example

Vereinfache den folgenden Ausdruck:

$(x_1 \cdot x_2) + (x_2 \cdot x_3)$

Vereinfachen von Ausdrücken

Example

Vereinfache den folgenden Ausdruck:

$(x_1 \cdot x_2) + (x_2 \cdot x_3)$

$$\Rightarrow x_2(x_1 + x_3)$$

Aufgabe 1.3

Vereinfache den folgenden Ausdruck:

$((\bar{x}_1 \cdot \bar{x}_2) + x_3 + (\bar{x}_1 \cdot \bar{x}_3 \cdot x_4) + (x_2 \cdot \bar{x}_3 \cdot x_4))$

Vereinfachen von Ausdrücken

Aufgabe 1.3

Vereinfache den folgenden Ausdruck:

$((\bar{x_1} \cdot \bar{x_2}) + x_3 + (\bar{x_1} \cdot \bar{x_3} \cdot x_4) + (x_2 \cdot \bar{x_3} \cdot x_4))$

1. Schritt - Absorption: $((\bar{x_1} \cdot \bar{x_2}) + x_3 + (\bar{x_1} \cdot x_4) + (x_2 \cdot \bar{x_3} \cdot x_4))$

Vereinfachen von Ausdrücken

Aufgabe 1.3

Vereinfache den folgenden Ausdruck:

$((\bar{x_1} \cdot \bar{x_2}) + x_3 + (\bar{x_1} \cdot \bar{x_3} \cdot x_4) + (x_2 \cdot \bar{x_3} \cdot x_4))$

1. Schritt - Absorption: $((\bar{x_1} \cdot \bar{x_2}) + x_3 + (\bar{x_1} \cdot x_4) + (x_2 \cdot \bar{x_3} \cdot x_4))$
2. Schritt - Absorption: $((\bar{x_1} \cdot \bar{x_2}) + x_3 + (\bar{x_1} \cdot x_4) + (x_2 \cdot x_4))$

Vereinfachen von Ausdrücken

Aufgabe 1.3

Vereinfache den folgenden Ausdruck:
$((\bar{x_1} \cdot \bar{x_2}) + x_3 + (\bar{x_1} \cdot \bar{x_3} \cdot x_4) + (x_2 \cdot \bar{x_3} \cdot x_4))$

1. Schritt - Absorption: $((\bar{x_1} \cdot \bar{x_2}) + x_3 + (\bar{x_1} \cdot x_4) + (x_2 \cdot \bar{x_3} \cdot x_4))$
2. Schritt - Absorption: $((\bar{x_1} \cdot \bar{x_2}) + x_3 + (\bar{x_1} \cdot x_4) + (x_2 \cdot x_4))$
3. Schritt - Consensus: $((\bar{x_1} \cdot \bar{x_2}) + x_3 + (x_2 \cdot x_4))$

Vereinfachen von Ausdrücken - fakultativ

Aufgabe 1.Bonus

Vereinfache den folgenden Ausdruck:

$\overline{(\overline{(x_1 + x_2)} \cdot x_3)} \cdot (x_1 \cdot x_3)$

Tipp: de Morgan

Vereinfachen von Ausdrücken - fakultativ

Aufgabe 1.Bonus

Vereinfache den folgenden Ausdruck:

$\overline{(\overline{(x_1 + x_2)} \cdot x_3)} \cdot (x_1 \cdot x_3)$

1. Schritt - de Morgan: $\overline{(\overline{(x_1 + x_2)}} + \bar{x}_3) \cdot (x_1 \cdot x_3)$

Aufgabe 1.Bonus

Vereinfache den folgenden Ausdruck:

$\overline{(\overline{(x_1 + x_2)} \cdot x_3)} \cdot (x_1 \cdot x_3)$

1. Schritt - de Morgan: $\overline{(\overline{(x_1 + x_2)}} + \bar{x}_3) \cdot (x_1 \cdot x_3)$
2. Schritt - Doppeltes Komplement: $(x_1 + x_2 + \bar{x}_3) \cdot (x_1 \cdot x_3)$

Vereinfachen von Ausdrücken - fakultativ

Aufgabe 1.Bonus

Vereinfache den folgenden Ausdruck:

$\overline{(\overline{(x_1 + x_2)} \cdot x_3)} \cdot (x_1 \cdot x_3)$

1. Schritt - de Morgan: $(\overline{\overline{(x_1 + x_2)}} + \bar{x}_3) \cdot (x_1 \cdot x_3)$
2. Schritt - Doppeltes Komplement: $(x_1 + x_2 + \bar{x}_3) \cdot (x_1 \cdot x_3)$
3. Schritt - Distributivgesetz: $(x_1 \cdot x_3 \cdot x_1) + (x_1 \cdot x_3 \cdot x_2) + (x_1 \cdot x_3 \cdot \bar{x}_3)$

Vereinfachen von Ausdrücken - fakultativ

Aufgabe 1.Bonus

Vereinfache den folgenden Ausdruck:

$\overline{(\overline{(x_1 + x_2)} \cdot x_3)} \cdot (x_1 \cdot x_3)$

1. Schritt - de Morgan: $(\overline{\overline{(x_1 + x_2)}} + \bar{x}_3) \cdot (x_1 \cdot x_3)$
2. Schritt - Doppeltes Komplement: $(x_1 + x_2 + \bar{x}_3) \cdot (x_1 \cdot x_3)$
3. Schritt - Distributivgesetz: $(x_1 \cdot x_3 \cdot x_1) + (x_1 \cdot x_3 \cdot x_2) + (x_1 \cdot x_3 \cdot \bar{x}_3)$
4. Schritt - Idempotenz: $(x_1 \cdot x_3) + (x_1 \cdot x_3 \cdot x_2) + (x_1 \cdot x_3 \cdot \bar{x}_3)$

Aufgabe 1.Bonus

Vereinfache den folgenden Ausdruck:

$\overline{(\overline{(x_1 + x_2)} \cdot x_3)} \cdot (x_1 \cdot x_3)$

1. Schritt - de Morgan: $(\overline{\overline{(x_1 + x_2)}} + \bar{x}_3) \cdot (x_1 \cdot x_3)$
2. Schritt - Doppeltes Komplement: $(x_1 + x_2 + \bar{x}_3) \cdot (x_1 \cdot x_3)$
3. Schritt - Distributivgesetz: $(x_1 \cdot x_3 \cdot x_1) + (x_1 \cdot x_3 \cdot x_2) + (x_1 \cdot x_3 \cdot \bar{x}_3)$
4. Schritt - Idempotenz: $(x_1 \cdot x_3) + (x_1 \cdot x_3 \cdot x_2) + (x_1 \cdot x_3 \cdot \bar{x}_3)$
5. Schritt - 0 und 1: $(x_1 \cdot x_3) + (x_1 \cdot x_3 \cdot x_2) + 0$

Aufgabe 1.Bonus

Vereinfache den folgenden Ausdruck:

$\overline{(\overline{(x_1 + x_2)} \cdot x_3)} \cdot (x_1 \cdot x_3)$

1. Schritt - de Morgan: $(\overline{\overline{(x_1 + x_2)}} + \bar{x}_3) \cdot (x_1 \cdot x_3)$
2. Schritt - Doppeltes Komplement: $(x_1 + x_2 + \bar{x}_3) \cdot (x_1 \cdot x_3)$
3. Schritt - Distributivgesetz: $(x_1 \cdot x_3 \cdot x_1) + (x_1 \cdot x_3 \cdot x_2) + (x_1 \cdot x_3 \cdot \bar{x}_3)$
4. Schritt - Idempotenz: $(x_1 \cdot x_3) + (x_1 \cdot x_3 \cdot x_2) + (x_1 \cdot x_3 \cdot \bar{x}_3)$
5. Schritt - 0 und 1: $(x_1 \cdot x_3) + (x_1 \cdot x_3 \cdot x_2) + 0$
6. Schritt - Absorption: $(x_1 \cdot x_3)$

Aufgabe 1.4

Eine Klingelanlage soll mit einem Booleschen Ausdruck modelliert werden. Die Variable x_1 steht für das Drücken der Klingel (1 wenn gedrückt). Die Variable x_2 soll das Betätigen des Türöffners symbolisieren (1 wenn gedrückt). Die Tür hat zudem einen Öffnungssensor x_3 (1 wenn offen).
Die Tür soll sich öffnen wenn die Klingel und der Türöffner betätigt wurden, allerdings nicht, wenn sie bereits offen ist.

Aufgabe 1.4

Eine Klingelanlage soll mit einem Booleschen Ausdruck modelliert werden. Die Variable x_1 steht für das Drücken der Klingel (1 wenn gedrückt). Die Variable x_2 soll das Betätigen des Türöffners symbolisieren (1 wenn gedrückt). Die Tür hat zudem einen Öffnungssensor x_3 (1 wenn offen).
Die Tür soll sich öffnen wenn die Klingel und der Türöffner betätigt wurden, allerdings nicht, wenn sie bereits offen ist.

$$(x_1 \cdot x_2 \cdot \bar{x}_3)$$

Aufgabe 1.5

Erstelle einen Booleschen Ausdruck, der eine 1 ausgibt, falls zwei der drei Eingänge x_1, x_2, x_3 aktiv sind.

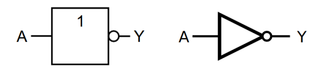
Aufgabe 1.5

Erstelle einen Booleschen Ausdruck, der eine 1 ausgibt, falls zwei der drei Eingänge x_1, x_2, x_3 aktiv sind.

$$(x_1 \cdot x_2) + (x_1 \cdot x_3) + (x_2 \cdot x_3)$$

Logikgatter

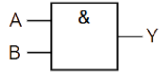
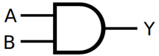
Grundlagen - NOT-Gatter

Name	Bezeichnung	Symbole
NOT	Nicht	

Wertetabelle

A	Y
0	1
1	0

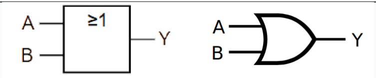
Grundlagen - AND-Gatter

Name	Bezeichnung	Symbole
AND	Und	 

Wertetabelle

A	B	Y
0	0	0
1	0	0
0	1	0
1	1	1

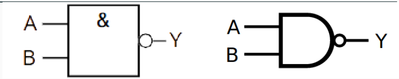
Grundlagen - OR-Gatter

Name	Bezeichnung	Symbole
OR	Oder	

Wertetabelle

A	B	Y
0	0	0
1	0	1
0	1	1
1	1	1

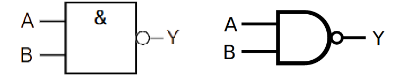
Grundlagen - NAND-Gatter

Name	Bezeichnung	Symbole
NAND	Nicht-Und	

Wertetabelle

A	B	Y
0	0	
1	0	
0	1	
1	1	

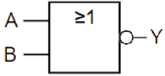
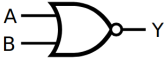
Grundlagen - NAND-Gatter

Name	Bezeichnung	Symbole
NAND	Nicht-Und	

Wertetabelle

A	B	Y
0	0	1
1	0	1
0	1	1
1	1	0

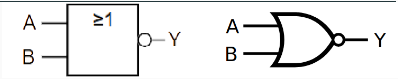
Grundlagen - NOR-Gatter

Name	Bezeichnung	Symbole
NOR	Nicht-Oder	 

Wertetabelle

A	B	Y
0	0	
1	0	
0	1	
1	1	

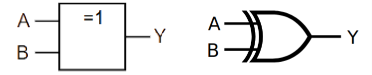
Grundlagen - NOR-Gatter

Name	Bezeichnung	Symbole
NOR	Nicht-Oder	

Wertetabelle

A	B	Y
0	0	1
1	0	0
0	1	0
1	1	0

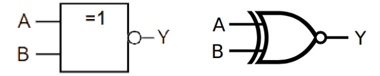
Grundlagen - XOR-Gatter

Name	Bezeichnung	Symbole
XOR	Exklusiv-Oder	

Wertetabelle

A	B	Y
0	0	0
1	0	1
0	1	1
1	1	0

Grundlagen - XNOR-Gatter

Name	Bezeichnung	Symbole
XNOR	Exklusiv-Nicht-Oder	

Wertetabelle

A	B	Y
0	0	1
1	0	0
0	1	0
1	1	1

Anwendung: Calliope

Aufgabe

Beschreibe die Funktionsweise des Gatters in eigenen Worten. Nutze dazu Sätze wie: „`Die Lampe von Gatter 1 ist nur dann aus, wenn...`“

Anwendung: Calliope

Mein Gatterprogramm

`https://l.boettcher.schule/f/GatterCalliope`

Gatter verknüpfen

`https://l.boettcher.schule/l/TinoCalliope`

Spiel: Gates of Logic

https://l.boettcher.schule/l/GoL

Gartenbewässerung (1)

Eine Bewässerungsanlage soll nur wässern, wenn der Wettersensor gutes Wetter meldet (gekennzeichnet mit „s“) und der Bewegungssensor niemanden im Garten erkennt (gekennzeichnet mit „g“). Die beiden Signale werden jeweils mit 0 (Schlechtes Wetter/Garten besetzt) oder 1 (Gutes Wetter/Garten frei) kodiert.

Aufgabe 2.1

Interpretiere folgende Zustände:

- `s=0, g=1`
- `s=0, g=0`
- `s=1, g=1`

Gartenbewässerung (2)

Aufgabe 2.2

Ergänze die Wertetabelle so, dass in der letzten Spalte das in der Beschreibung gewünschte Verhalten der Bewässerungsanlage abgebildet wird (W=1 - Bewässerung, W=0 - Keine Bewässerung).

s	g	W
0	0	
1	0	
0	1	
1	1	

Aufgabe 2.3

Welches Logikgatter würdest du wählen, um das gewünschte Verhalten abzubilden? Begründe deine Auswahl.

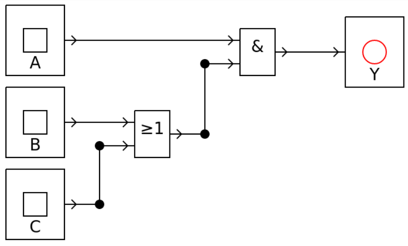
Eigentlich müssten spätestens jetzt folgende Begriffe geklärt werden:

- Schaltung
- Eingang
- Ausgang
- Verzweigungen im Schaltplan

Schaltnetze

Aufgabe 3.1

Betrachte die folgende Schaltung und fülle die dazugehörige Wahrheitstabelle aus. Bei welcher Belegung entsteht welches Ergebnis?

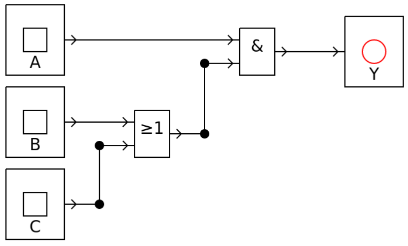


A	B	C	Y
1	1	1	
1	1	0	
1	0	1	
1	0	0	
0	1	1	
0	1	0	
0	0	1	
0	0	0	

Verknüpfung von Gattern

Aufgabe 3.1

Betrachte die folgende Schaltung und fülle die dazugehörige Wahrheitstabelle aus. Bei welcher Belegung entsteht welches Ergebnis?



A	B	C	Y
1	1	1	1
1	1	0	1
1	0	1	1
1	0	0	0
0	1	1	0
0	1	0	0
0	0	1	0
0	0	0	0

Verknüpfung von Gattern (2)

Aufgabe 3.2

Überlege dir einen Anwendungsfall aus der realen Welt, der mit einer solchen Schaltung abgedeckt werden könnte.

Aufgabe 3.3

Erstelle einen Booleschen Ausdruck, der das Verhalten der Schaltung beschreibt.

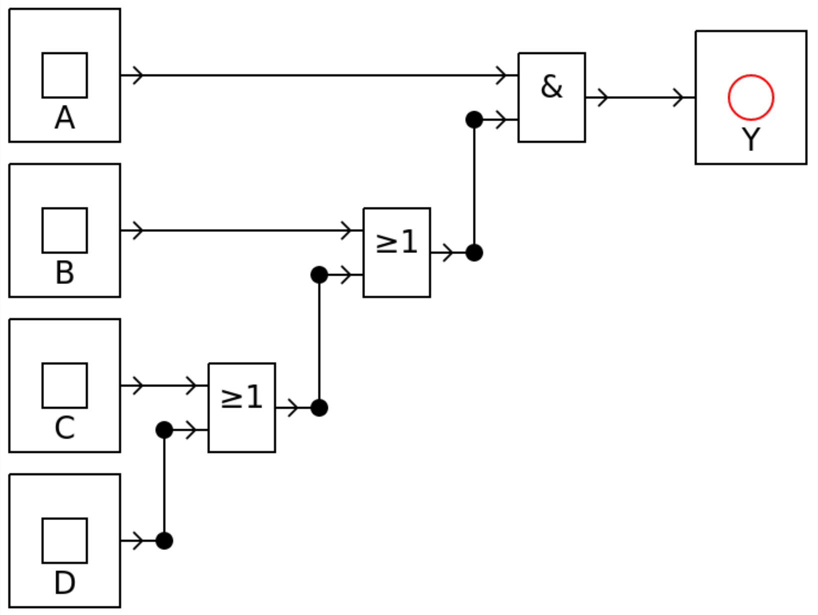
Verknüpfung von Gattern (3)

Angenommen, diese Schaltung würde eine Alarmanlage mit einen Hauptschalter (A) und zwei Sensoren (B und C) abbilden. Wenn der Hauptschalter gedrückt ist, und einer der Sensoren aktiviert wird, schlägt die Anlage Alarm.

Aufgabe 3.4

Füge einen dritten Sensor D hinzu, der gleichberechtigt zu den Sensoren B und C anschlagen kann.

Verknüpfung von Gattern (3)



Verknüpfung von Gattern (3)

Aufgabe 3.5

Welches allgemeine Vorgehen müsste man verfolgen, um aus einer Verknüpfung von Logikgattern einen Booleschen Ausdruck zu erhalten?

Verknüpfung von Gattern (3)

Aufgabe 3.5

Welches allgemeine Vorgehen müsste man verfolgen, um aus einer Verknüpfung von Logikgattern einen Booleschen Ausdruck zu erhalten?

- vom Output zum Input gehen (Operatorbaum)

[2] Das Gegenteil (jeden Weg zu einem falschen Output verfolgen) nennt sich konjunktive Normalform KNF.

Aufgabe 3.5

Welches allgemeine Vorgehen müsste man verfolgen, um aus einer Verknüpfung von Logikgattern einen Booleschen Ausdruck zu erhalten?

- Jeden Weg zu einem wahren Output verfolgen (DNF - Disjunctive Normal Form)[2]

[2]Das Gegenteil (jeden Weg zu einem falschen Output verfolgen) nennt sich konjunktive Normalform KNF.

Aufgabe 3.5

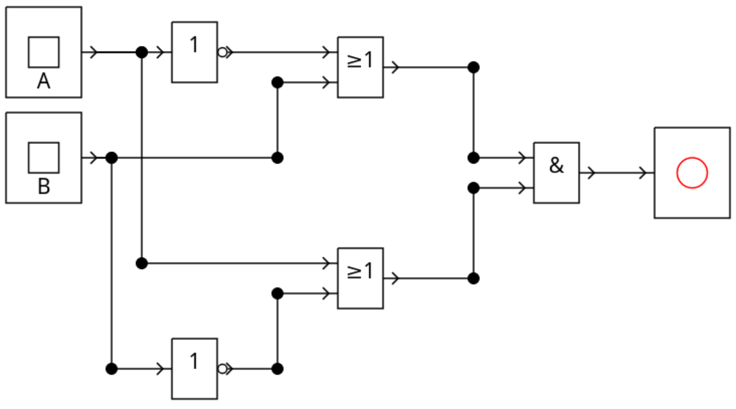
Welches allgemeine Vorgehen müsste man verfolgen, um aus einer Verknüpfung von Logikgattern einen Booleschen Ausdruck zu erhalten?

- Eine Variablenreihenfolge festlegen und einen Entscheidungsbaum zeichnen (BDD - Binary Decision Diagram)

[2]Das Gegenteil (jeden Weg zu einem falschen Output verfolgen) nennt sich konjunktive Normalform KNF.

Aufgabe 3.6

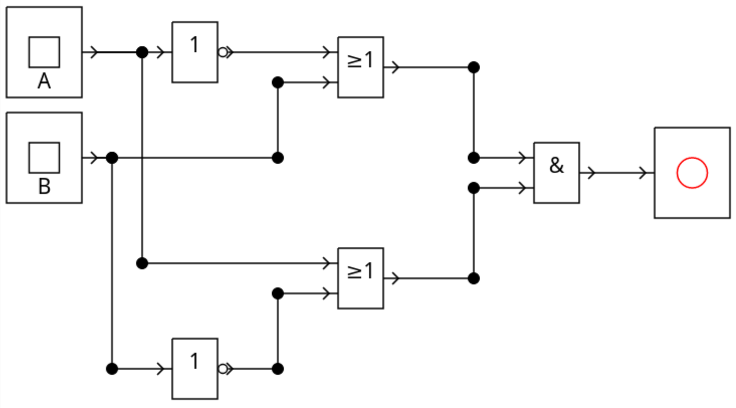
Finde einen Booleschen Ausdruck, der die folgende Schaltung darstellt. Gehe vom Output zum Input.



Operatorbaum

Aufgabe 3.6

Finde einen Booleschen Ausdruck, der die folgende Schaltung darstellt. Gehe vom Output zum Input.



$$((\bar{A}+B)\cdot(A+\bar{B}))$$

Operatorbaum

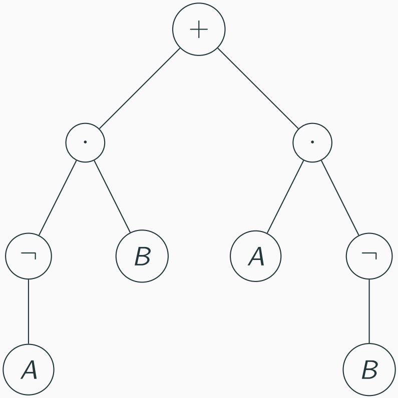


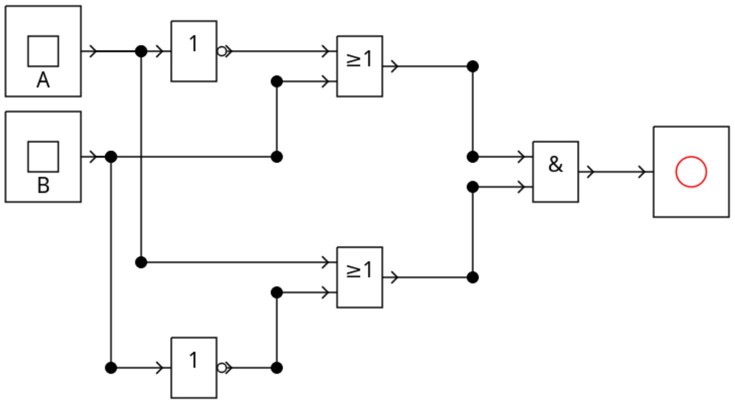
Abbildung 1: Operatorbaum zu $((\bar{A} + B) \cdot (A + \bar{B}))$

- Kann direkt in Schaltungen umgesetzt werden.
- Nicht gut für Optimierungen geeignet.

DNF - Disjunctive Normal Form

Aufgabe 3.6

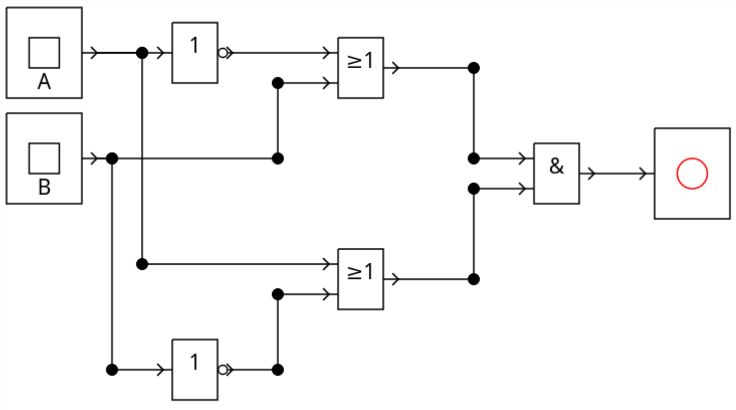
Finde einen Booleschen Ausdruck, der die folgende Schaltung darstellt. Betrachte dafür alle Bedingungen, unter denen die LED leuchten kann.



DNF - Disjunctive Normal Form

Aufgabe 3.6

Finde einen Booleschen Ausdruck, der die folgende Schaltung darstellt. Betrachte dafür alle Bedingungen, unter denen die LED leuchten kann.



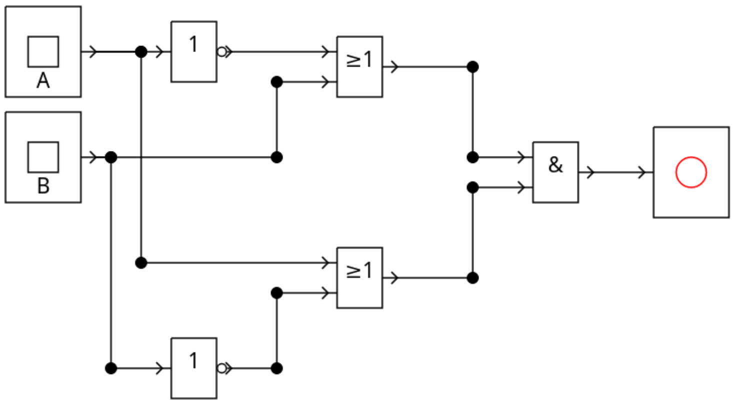
$$((\bar{A} \cdot \bar{B}) + (A \cdot B))$$

BDD - Binary Decision Diagram

Aufgabe 3.6

Finde einen Booleschen Ausdruck, der die folgende Schaltung darstellt. Erstelle dafür eine Belegungstabelle, aus der du einen Baum formst, den du anschließend optimieren kannst.

Reihenfolge: A $<$ B



BDD - Binary Decision Diagram

A	B	Ergebnis
1	1	1
1	0	0
0	1	0
0	0	1

BDD - Binary Decision Diagram

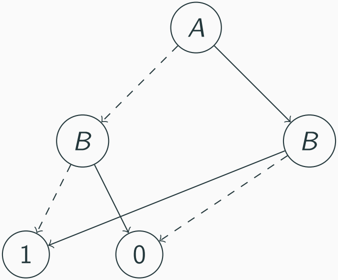


Abbildung 2: Binary Decision Diagram für $A < B$

- Erst sinnvoll bei größeren Ausdrücken/Tabellen
- Sehr gut für Optimierungen geeignet
- Hängt ab von gewählter Variablenreihenfolge
- Direkt in Hardware realisierbar $\rightarrow$ jeder Knoten ein Multiplexer

Spiel: Digitalo

Digitalo ist ein Kartenspiel, bei welchem zwei Spieler*innen versuchen, aus Gatter-Karten eine pyramidenförmige logische Schaltung aufzubauen. Kommerziell ist es als „Booleo“ bekannt.

Anleitung

`https://l.boettcher.schule/l/DigitaloAnleitung`

selber Drucken

`https://l.boettcher.schule/l/DigitaloKarten`

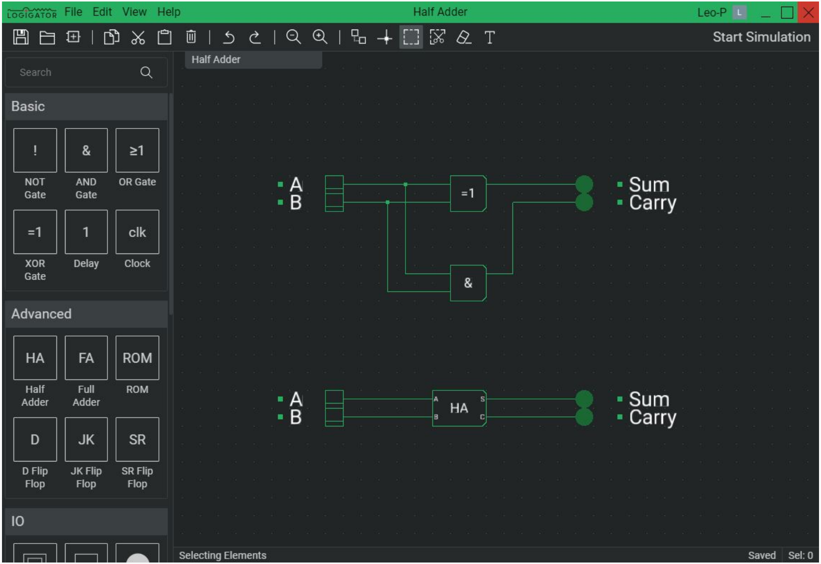
Warum simulieren?

- Stärkt Verständnis
- Lebensweltbezug
- Änderungen werden direkt sichtbar
- (Wird im Abitur benötigt)

Logigator

- Auf IO-Stick verfügbar
- Wird aktiv entwickelt
- Soll LogicSim ersetzen
- `https://logigator.com/` als Web-App
- `https://logigator.com/download/` als Download für Linux/Windows
- **Achtung:** Nicht kompatibel mit LogicSim!

Logigator-Interface

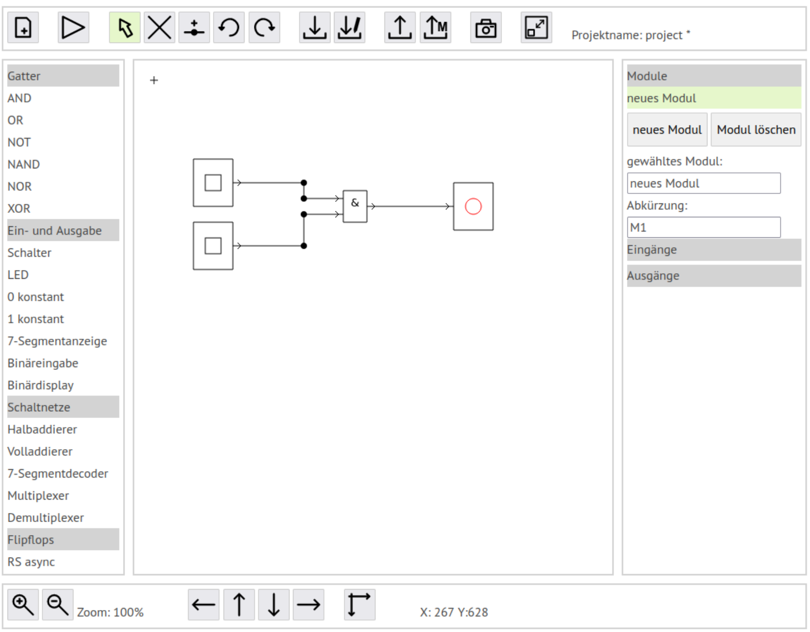


DSimWeb

- Noch nicht auf IO-Stick
- Wird aktiv entwickelt
- Sehr intuitive Bedienung
- Web-App auch offline verfügbar

`https://l.boettcher.schule/l/DSimWeb`

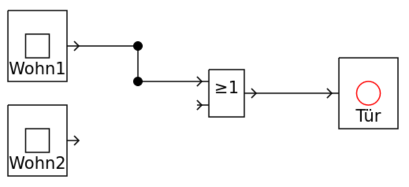
DSimWeb-Interface



Der Türöffner in einem Mehrfamilienhaus muss reagieren, sobald der entsprechende Knopf in einer der Wohnungen gedrückt wird.[3]

Aufgabe 3.1

Überprüfe die gegebene Schaltung in DSimWeb für ein Zweifamilienhaus auf Korrektheit und korrigiere ggf. Fehler, indem du Signalleitungen von einem Ausgang an einen Eingang legst.



[1]Nach Inf-Schule.de - Grundgatter sowie T. Hempel, CC BY-SA

Türöffner (2)

Aufgabe 3.2

Vervollständige folgenden Satz mit den Begriffen „und“, „oder“, „entweder-oder“ bzw. „nicht“:

`Wenn der Türknopf in Wohnung 1 ______ der Türknopf in Wohnung 2 gedrückt wird, dann ___________.`

Aufgabe 3.3

Erweitere die Schaltung so, dass sie auch in einen Haus mit 4 Wohnungen genutzt werden kann.

[1]Nach Inf-Schule.de - Grundgatter sowie T. Hempel, CC BY-SA

Raketenstart

Zum Start einer Rakete benötigt ein Raketenkontrollsystem eine Starterlaubnis. Wenn die Starterlaubnis erteilt ist, kann einer von zwei Startknöpfen gedrückt werden, um die Rakete zu starten.

Aufgabe 4.1

Erstelle in DSimWeb eine Schaltung mit zwei Logikgattern, die diesen Sachverhalt abbildet. Das Abheben der Rakete soll durch das Leuchten einer Lampe symbolisiert werden.

Aufgabe 4.2

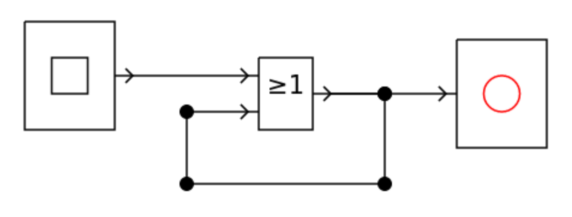
Füge der Schaltung einen „Präsidentenknopf“ hinzu. Dieser kann sämtliche Regeln überschreiben und die Rakete auch ohne Starterlaubnis starten lassen.

Aufgabe 4.3

Zur Wartung der Rakete ist ein Testmodus nötig. Wird der Schalter für den Testmodus betätigt werden sämtliche Eingaben ignoriert (auch der Präsidentenknopf) und die Rakete kann unter keinen Umständen starten.

Aufgabe 5.1

Baue folgende Schaltung in DSimWeb nach. Was passiert, wenn du den Schalter betätigst?



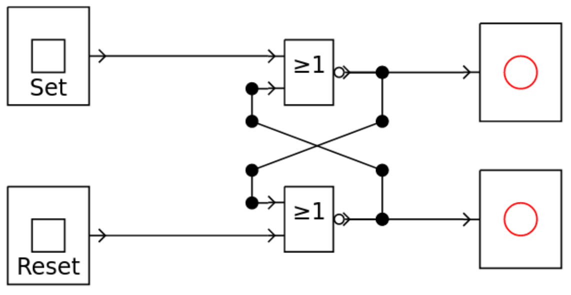
Aufgabe 5.2

Wo könnte eine solche Schaltung nützlich sein? Welche Möglichkeiten siehst du zur Erweiterung?
Baue eine Alarmanlage, die manuell zurückgesetzt werden muss, damit der Alarm aufhört.

Speichern von Zuständen - RS-FlipFlop

Aufgabe 5.3

Diese Schaltung nennt sich „R(eset)-S(et)-FlipFlop“. Baue sie in DSimWeb nach. Wie verhalten sich die beiden Ausgänge zueinander?



Man könnte an dieser Stelle thematisieren, warum ein FlipFlop besser ist als die Schaltung auf der vergangenen Folie (NAND-Gatter lassen sich leichter produzieren), aber das ist im Rahmenplan nicht vorgesehen. Außerdem: Prima Einstieg in Automatentheorie!

Module - FlipFlop

DSimWeb bietet (wie fast alle Logiksimulatoren) die Verwendung von Modulen an.

Die Realisierung von Addition erfolgt im Rechenwerk des von-Neumann-Rechners mittels Logikgattern. Aber wie?[4]

Aufgabe 6.1

Für das Addieren zweier Bits gilt folgende Wertetabelle:

A		B	Y	Ü
0	+	0	0	0
1	+	0	1	0
0	+	1	1	0
1	+	1	0	1

Wofür steht das Ü? Kann es ein einzelnes Logikgatter geben, dass uns diese Wertetabelle erzeugt? Gibt es ein Gatter, das unter Berücksichtigung der Eingänge A und B den Ausgang Y erzeugt?

[2]Nach T. Hempel 2023, CC BY-SA 4.0

Halbaddierer (2)

Aufgabe 6.2

Baue eine Schaltung, die die folgende Wertetabelle implementiert.

A		B	Y	Ü
0	+	0	0	0
1	+	0	1	0
0	+	1	1	0
1	+	1	0	1

Aufgabe 6.2 (alternative)

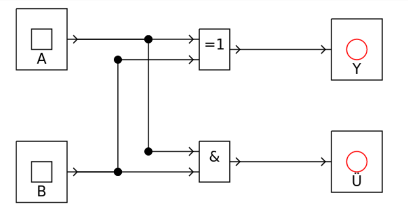
Baue eine Schaltung, die die folgende Wertetabelle implementiert, ohne ein XOR-Gatter zu verwenden.

[2] Nach T. Hempel 2023, CC BY-SA 4.0

Halbaddierer (2)

Aufgabe 6.2

Baue eine Schaltung, die die folgende Wertetabelle implementiert.



Aufgabe 6.2 (alternative)

Baue eine Schaltung, die die folgende Wertetabelle implementiert, ohne ein XOR-Gatter zu verwenden.

Volladdierer (1)

Ein Volladdierer arbeitet wie ein Halbaddierer, berücksichtigt aber zusätzlich noch einen ggf. vorhandenen Übertrag C.

Aufgabe 7.1

Ergänze die Wertetabelle des Halbaddierers um einen weiteren Input „C“ und passe die Ausgaben entsprechend an.

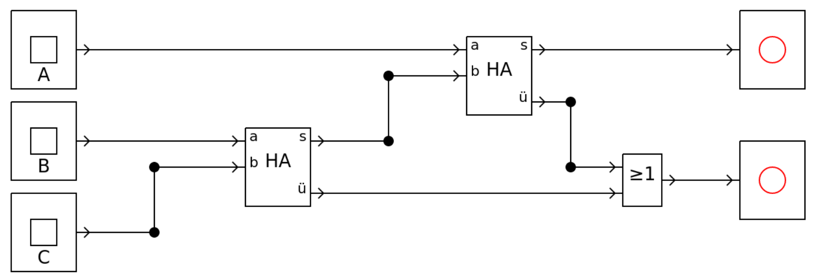
`Tipp: Deine Tabelle muss 5 Zeilen und 8 Spalten besitzen.`

[2] Nach T. Hempel 2023, CC BY-SA 4.0

Volladdierer (2)

Aufgabe 7.2

Überprüfe die Korrektheit der Schaltung für einen Volladdierer anhand deiner Wertetabelle.



[2]Nach T. Hempel 2023, CC BY-SA 4.0

Fragen?

Multiplexer (1)

Ein Multiplexer (kurz: MUX oder Mux) ist eine Selektionsschaltung, mit der aus einer Anzahl von Eingangssignalen eines ausgewählt und an den Ausgang durchgeschaltet werden kann.

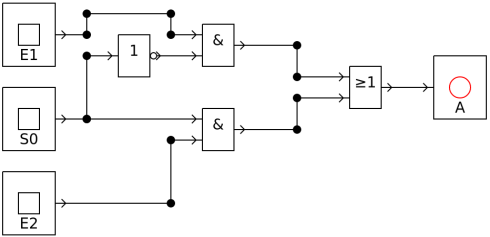
Aufgabe 8.1

Beschreibe einen möglichen Anwendungsbereich für einen Multiplexer. Informiere dich mittels einer Internetrecherche über weitere Anwendungsbereiche.

Multiplexer (2)

Aufgabe 8.2

Unten siehst du das Schaltbild eines Multiplexers. Erstelle eine Wahrheitstafel für die 3 Eingänge und den Ausgang.



Aufgabe 8.3

Erweitere den Multiplexer um 4 weitere Eingänge ($E_3 - E_6$). Wie viele Schalteingänge S werden dafür benötigt?

BDD - Binary Decision Diagram

Aufgabe 8.4

Finde ein BDD, der den Booleschen Ausdruck

$$((x_1 + x_3) \cdot (x_2 + x_4))$$

darstellt.

Variablenordnung: $x_1 < x_2 < x_3 < x_4$

Aufgabe 8.5

Finde ein BDD, der den Booleschen Ausdruck

$$((x_1 + x_3) \cdot (x_2 + x_4))$$

darstellt.

Variablenordnung: $x_1 < x_3 < x_2 < x_4$

Aufgabe 8.4

Finde ein BDD, der den Booleschen Ausdruck

$$((x_1 + x_3) \cdot (x_2 + x_4))$$

darstellt.

Variablenordnung: $x_1 < x_2 < x_3 < x_4$

Aufgabe 8.5

Finde ein BDD, der den Booleschen Ausdruck

$$((x_1 + x_3) \cdot (x_2 + x_4))$$

darstellt.

Variablenordnung: $x_1 < x_3 < x_2 < x_4$

Zusammenfassung

Rückblick Lehrplan

Leistungskurs (10 Ustd.)

- Boolesche Algebra
  - NOT, AND, OR und deren Verknüpfungen
  - Wahrheitstabellen
  - Doppelnegation, Kommutativ-, Assoziativ- und Distributivgesetz, DeMorgansche Regeln
  - Vereinfachung boolescher Ausdrücke
- Übertragen der theoretischen Grundlagen auf die Umsetzung in Schaltnetzen
  - Schaltnetz-Analyse
  - Schaltnetz-Synthese
  - Schaltnetz-Simulation

Praktische Links

- https://www.inf-schule.de/rechner/digitaltechnik
- https://www.inf-schule.de/rechner/digitaltechnik/Simulatoren/DSimWeb/DSimWeb_Vollversion
- https://www.inf-schule.de/rechner/digitaltechnik/gatter/digitalo
- https://logigator.com/
- https://nandgame.com/
- https://calliopemini.info/