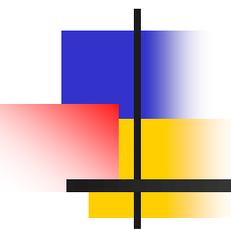


iMedia 2024

Funktionsweise eines Rechners mit interaktiven Tools erklären



Michael Becker

Sickingen-Gymnasium Landstuhl

michael.becker@sg-l.schule

DSimWeb: Schaltungen der Digitaltechnik aufbauen und simulieren

Vollversion: www.inf-schule.de/tools/dsim

Bedienungsanleitung: www.inf-schule.de/12.2.2.2

Schalterterme vereinfachen:

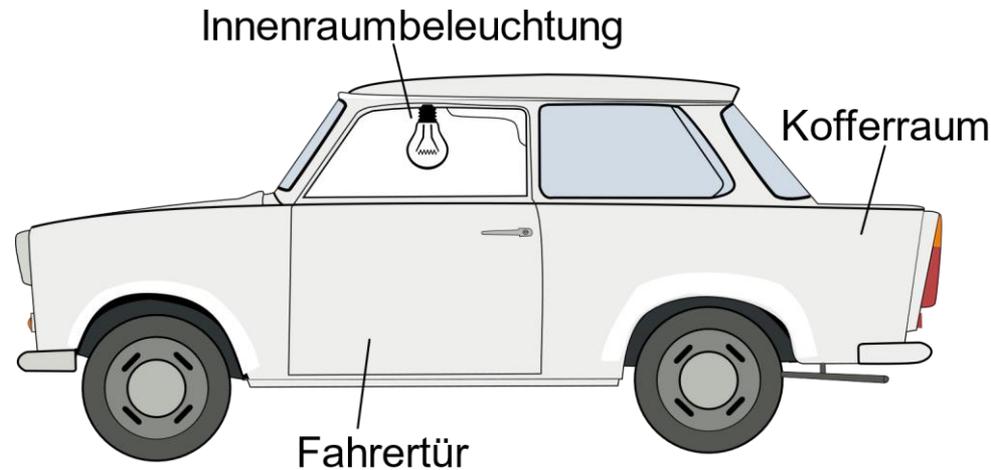
[www.inf-schule.de/tools/logic tool](http://www.inf-schule.de/tools/logic_tool)

Kapitel zur Digitaltechnik:

www.inf-schule.de/12.2

Beispiel: Lampe in einem Auto

Bsp.: Die Lampe im Innenraum eines Autos soll leuchten, wenn die Fahrertür oder der Kofferraum geöffnet ist.



gesucht: Schaltnetz, das die Beleuchtung im Auto steuert

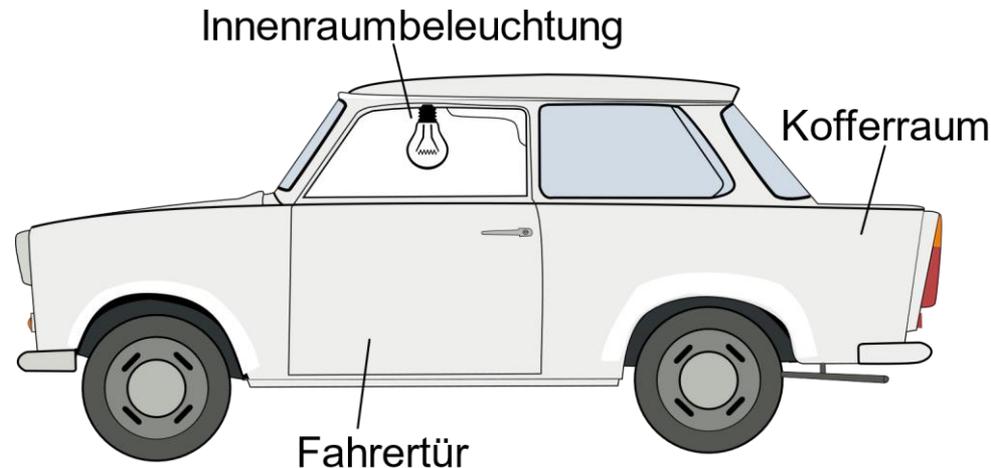
Einfache Schaltnetze können durch intuitives Verständnis konstruiert werden.

Bei komplexeren Schaltnetzen geht man oft in folgenden Schritten vor:

- 1) **Schalttabelle aufstellen**
- 2) Schaltterm aufstellen
- 3) optional (aber meist sinnvoll): Schaltterm vereinfachen
- 4) Schaltnetz aufbauen

Aufstellen einer Schalttafel

Bsp.: Die Lampe im Innenraum eines Autos soll leuchten, wenn die Fahrertür oder der Kofferraum geöffnet ist.



Schaltvariablen:

- **f**: Fahrertür geschlossen
- **k**: Kofferraum geschlossen
- **Y**: Licht im Auto ist an

Schalttafel:

f	k	Y
0	0	
0	1	
1	0	
1	1	

Einfache Schaltnetze können durch intuitives Verständnis konstruiert werden.

Bei komplexeren Schaltnetzen geht man oft in folgenden Schritten vor:

- 1) Schalttabelle aufstellen ✓
- 2) **Schalterm aufstellen**
- 3) optional (aber meist sinnvoll): Schalterm vereinfachen
- 4) Schaltnetz aufbauen

Schalterm aufstellen

Die disjunktive Normalform bietet ein Verfahren, mit dem man systematisch einen Schalterm zu einer Schalttable finden kann.

- Zu jeder Zeile, bei der das Ergebnis 1 ist, bildet man je einen Minterm. Dies ist ein Term, in dem alle Eingangsvariablen mit der UND-Verknüpfung verbunden sind. Hat eine Variable den Wert 0, steht sie negiert im Minterm.
- Die disjunktive Normalform erhält man, indem man alle Minterme durch ODER-Verknüpfungen verbindet.

f	k	Y	Minterm
0	0	1	$\bar{f} \wedge \bar{k}$
0	1	1	$\bar{f} \wedge k$
1	0	1	$f \wedge \bar{k}$
1	1	0	

disjunktive Normalform:

$$(\bar{f} \wedge \bar{k}) \vee (\bar{f} \wedge k) \vee (f \wedge \bar{k})$$

Einfache Schaltnetze können durch intuitives Verständnis konstruiert werden.

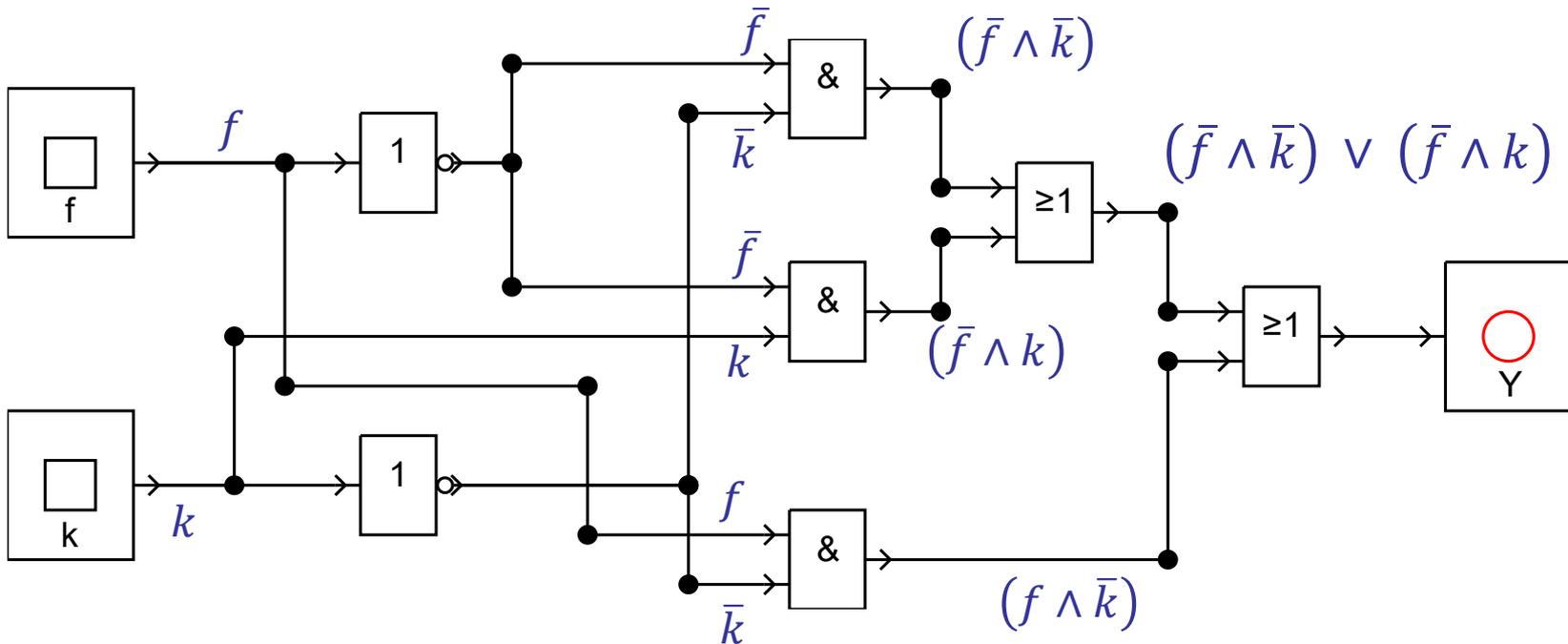
Bei komplexeren Schaltnetzen geht man oft in folgenden Schritten vor:

- 1) Schalttabelle aufstellen ✓
- 2) Schaltterm aufstellen ✓
- 3) optional (aber meist sinnvoll): Schaltterm vereinfachen
- 4) **Schaltnetz aufbauen**

machen wir später 😊

Übungen

$$(\bar{f} \wedge \bar{k}) \vee (\bar{f} \wedge k) \vee (f \wedge \bar{k})$$



Übung: Bauen Sie das Schaltnetz auf und testen Sie es.

<https://inf-schule.de/tools/dsim>

Weiterer Vorschlag zur Erkundung: www.inf-schule.de/12.2.3

Einfache Schaltnetze können durch intuitives Verständnis konstruiert werden.

Bei komplexeren Schaltnetzen geht man oft in folgenden Schritten vor:

- 1) Schalttabelle aufstellen ✓
- 2) Schaltterm aufstellen ✓
- 3) **optional (aber meist sinnvoll): Schaltterm vereinfachen**
- 4) Schaltnetz aufbauen

machen wir jetzt 😊

Schalterm vereinfachen

1. Möglichkeit: "von Hand" rechnen

$$\begin{aligned}
 & (\bar{f} \wedge \bar{k}) \vee (\bar{f} \wedge k) \vee (f \wedge \bar{k}) \\
 = & (\bar{f} \wedge \bar{k}) \vee (\bar{f} \wedge \bar{k}) \vee (\bar{f} \wedge k) \vee (f \wedge \bar{k}) \\
 = & (\bar{f} \wedge \bar{k}) \vee (\bar{f} \wedge k) \vee (\bar{f} \wedge \bar{k}) \vee (f \wedge \bar{k}) \\
 = & (\bar{f} \wedge \bar{k}) \vee (\bar{f} \wedge k) \vee (\bar{k} \wedge \bar{f}) \vee (\bar{k} \wedge f) \\
 = & \bar{f} \wedge (\bar{k} \vee k) \vee \bar{k} \wedge (\bar{f} \vee f) \\
 = & \bar{f} \wedge 1 \vee \bar{k} \wedge 1 \\
 = & \bar{f} \vee \bar{k}
 \end{aligned}$$

Probleme: aufwendig, benötigt Zeit

2. Möglichkeit: Tool zum Vereinfachen von Schaltermen verwenden

https://www.inf-schule.de/tools/logic_tool

$$(\bar{a} \wedge b \wedge \bar{c}) \vee (a \wedge b \wedge \bar{c})$$

kann eingegeben werden als: (not a and b and not c) or (a and b and not c)

oder: !a b !c or a b !c

Übung:

- Vereinfachen Sie den Schalterm $(\bar{f} \wedge \bar{k}) \vee (\bar{f} \wedge k) \vee (f \wedge \bar{k})$.
- Bauen Sie den vereinfachten Schalterm als Schaltung auf und testen Sie die Schaltung.
- Wie viele Gatter konnten durch die Vereinfachung des Schalterms eingespart werden?

Übung - Lösung

13

Schalterm: $(\bar{a} \wedge b \wedge \bar{c}) \vee (a \wedge b \wedge c)$

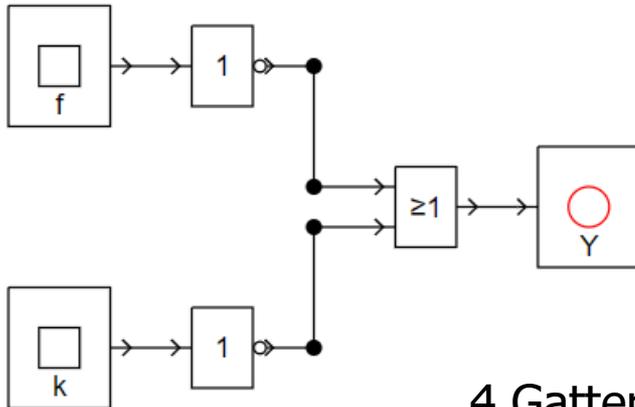
Eingabe (lang): (not f and not k) or (not f and k) or (f and not k)

Eingabe (kurz): !f !k or !f k or f !k

vereinfachter Schalterm: not f or not k

mit vereinfachtem Schalterm:

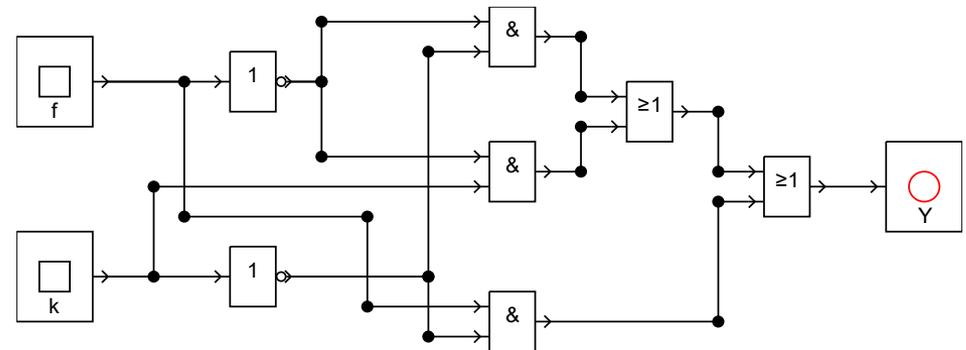
3 Gatter



4 Gatter eingespart!

vor der Vereinfachung:

7 Gatter

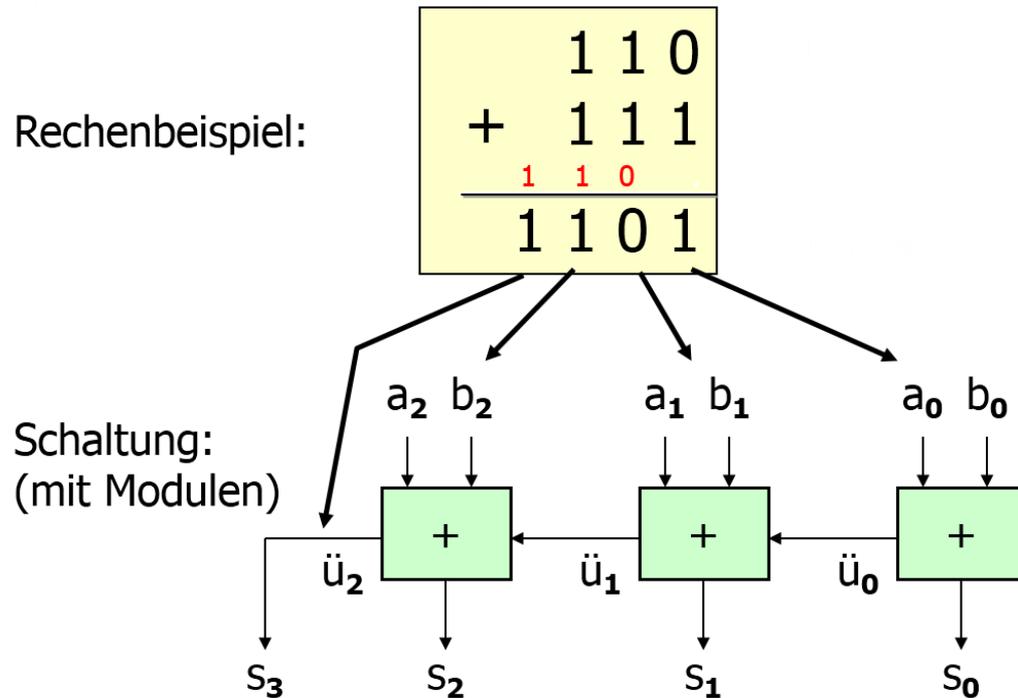


Weitere Vorschläge zur Erkundung:

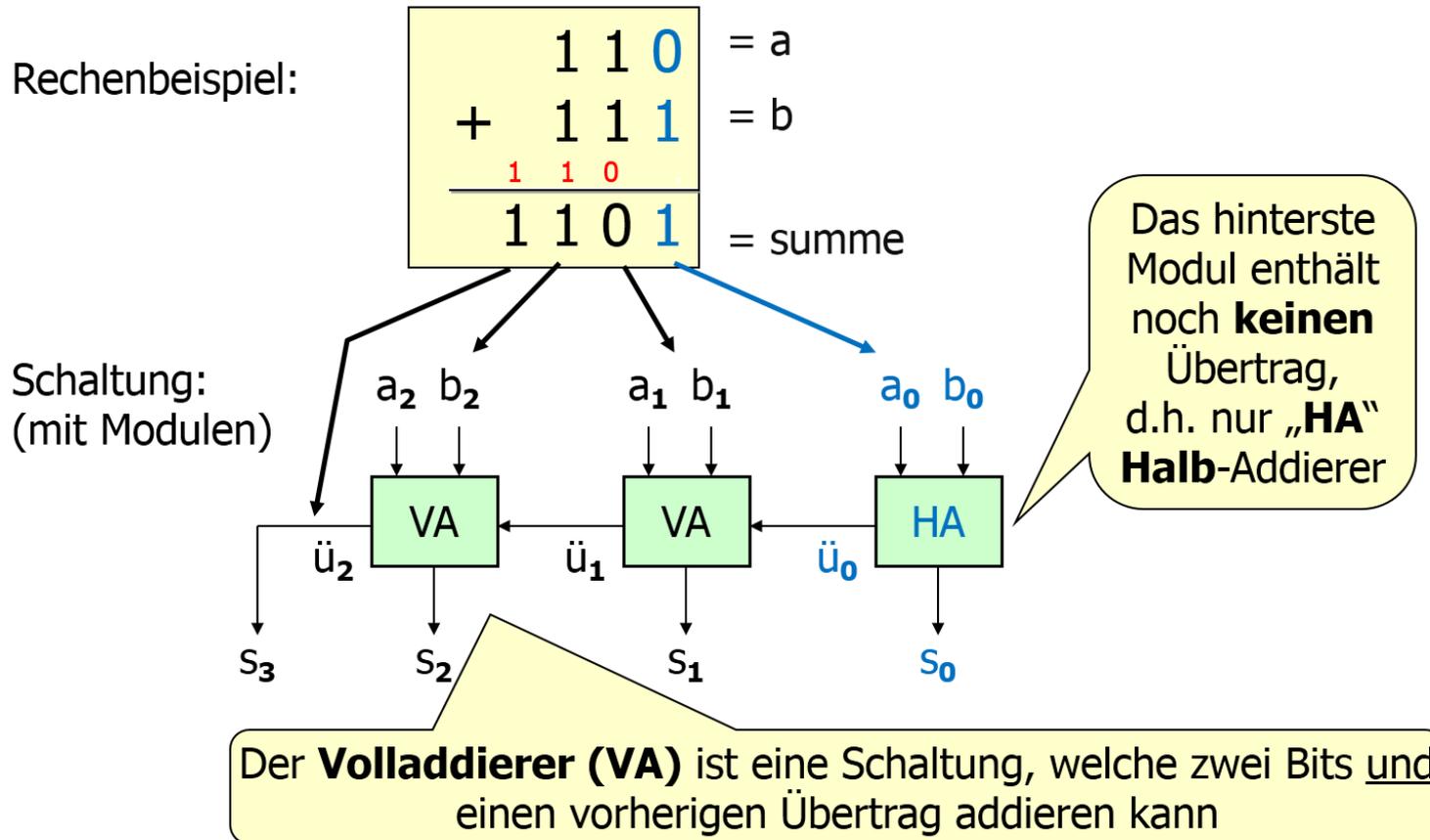
www.inf-schule.de/12.2.3

www.inf-schule.de/12.2.4

Bsp.: Addition zweier Zahlen im Binärsystem

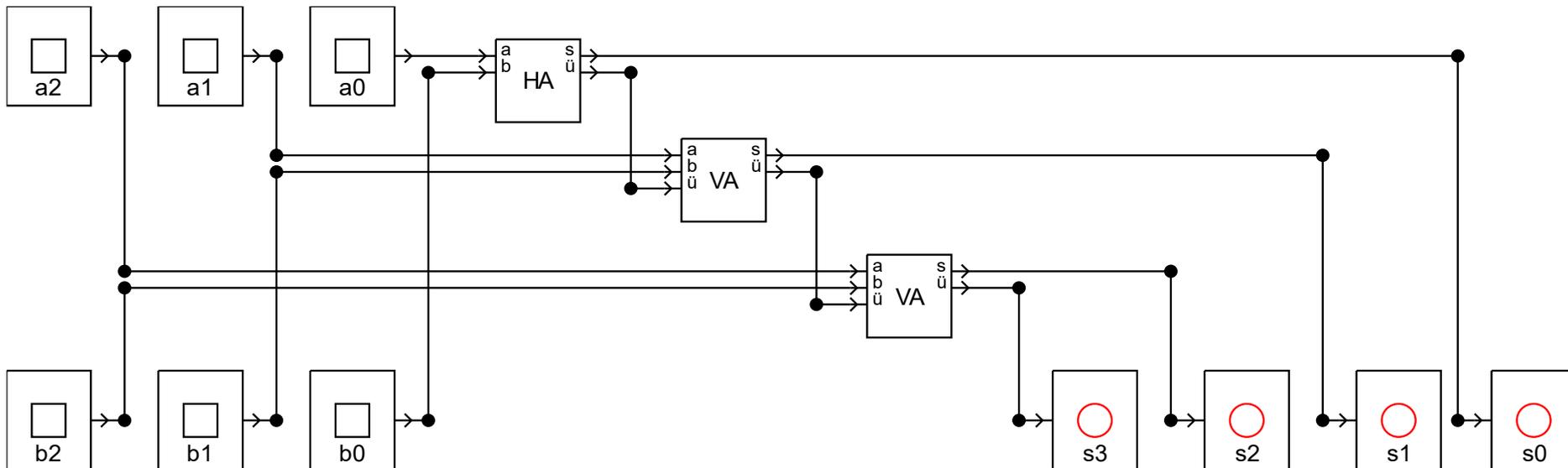


DsimWeb: Module



Übung:

- Öffnen Sie die Seite www.inf-schule.de/105.23
- In dem Modul Volladdierer: Machen Sie die Schalter zu Eingängen und die Lampen zu Ausgängen.
- Erstellen Sie ein neues Modul. Bauen Sie darin einen 3-Bit-Addierer auf:

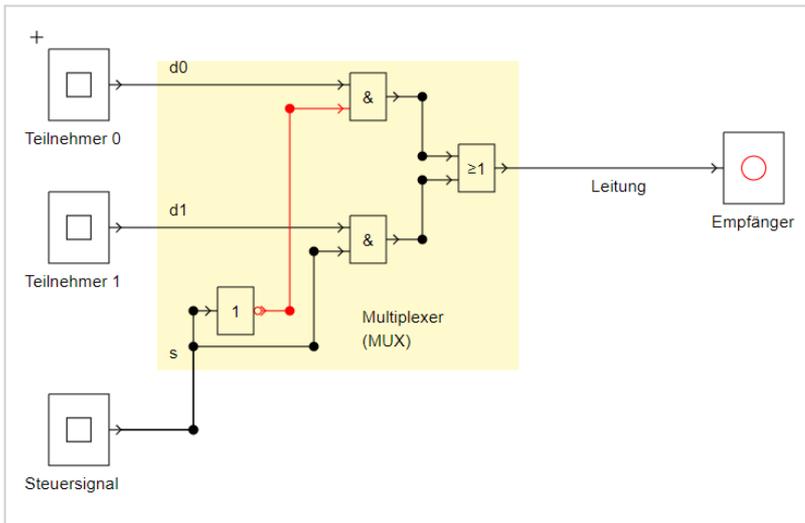


Weitere Vorschläge zur Erkundung:

www.inf-schule.de/12.2.6 (Projekt Spielstandsanzeige, verwendet Module)

z.B.: www.inf-schule.de/12.2.3.1

Simulation ohne Bearbeitung



eingeschränkte Auswahl an Bauteilen
keine Module

Interpreter für Bonsai-Assembler:

<https://inf-schule.de/tools/bonsai-assembler-interpreter>

Übersetzer für Bonsai-Assembler:

https://inf-schule.de/tools/bonsai_uebersetzer

Kapitel zu Modellrechnern:

Murmelrechner und Bonsai, verwenden wir heute:

www.inf-schule.de/12.3

Alternative: Johnny-Modellrechner

www.inf-schule.de/12.4

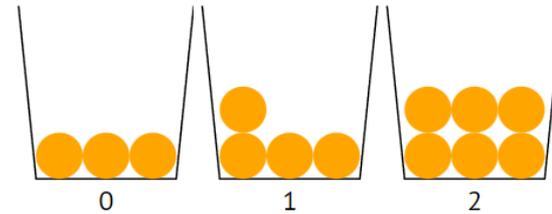
Modellrechner im Kapitel 12.3

Murmelrechner:

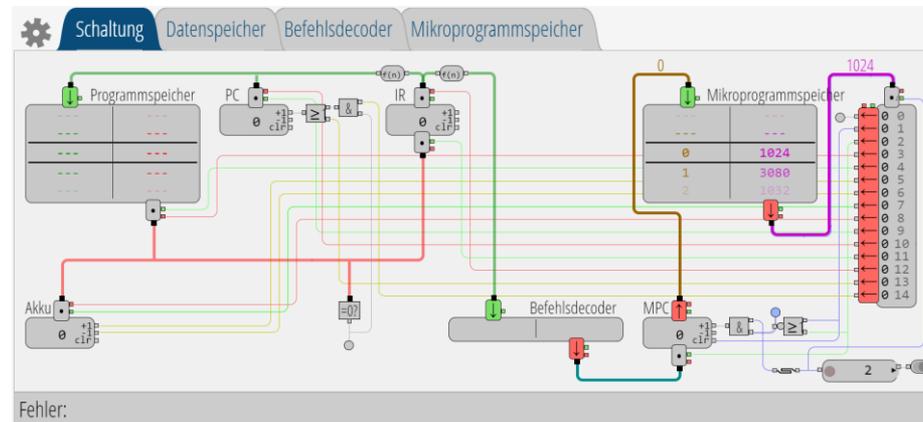
Stellt den Speicher mit
Bechern dar, in denen sich
Murmeln befinden

Daten:

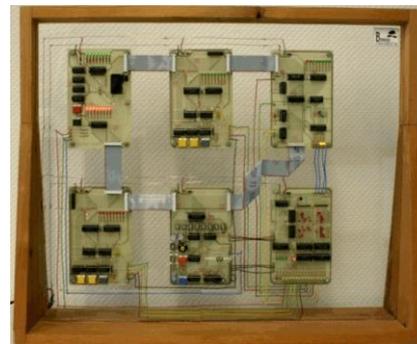
0	3
1	4
2	6



Bonsai-Modellrechner
(Simulator im Browser)



Bonsai-Modellrechner
(Hardware)



Öffnen Sie die Seite

<https://inf-schule.de/tools/bonsai-assembler-interpreter>

Führen Sie das Programm aus.

Beschreiben Sie die Bedeutung der Befehle `inc`, `dec`, `jmp`, `tst` und `hlt`.

Beschreiben Sie die Aufgabe, die das Programm erfüllt.

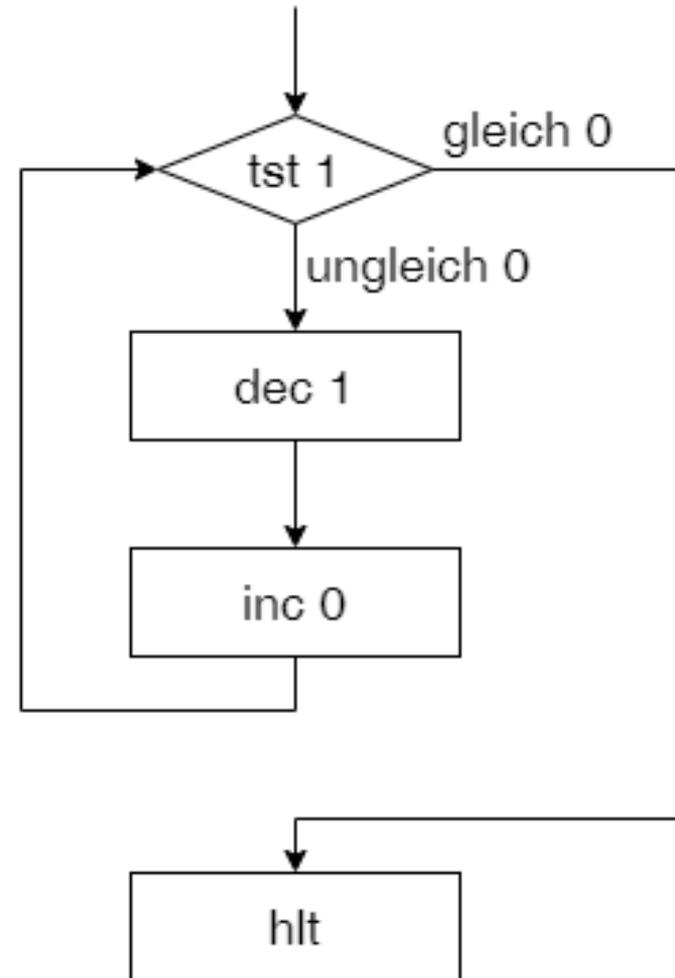
Murmelrechner, Bonsai-Simulator und Bonsai-Hardware verwenden die gleich Assemblersprache.

Operation	Bedeutung
<code>inc a</code>	Erhöhe den Wert an der Adresse <i>a</i> um 1. Weiter in der nächsten Programmzeile.
<code>dec a</code>	Erniedrige den Wert an der Adresse <i>a</i> um 1. Weiter in der nächsten Programmzeile.
<code>jmp z</code>	Weiter in der Programmzeile <i>z</i> .
<code>tst a</code>	Falls an der Adresse <i>a</i> der Wert 0 steht: weiter in der übernächsten Zeile sonst: weiter in der nächsten Zeile
<code>hlt</code>	Programm beenden

Programmablaufpläne

Ein Programmablaufplan erleichtert das Verständnis des Programms.

tst-Befehle dargestellt durch Rauten
jmp-Befehle dargestellt durch Pfeile



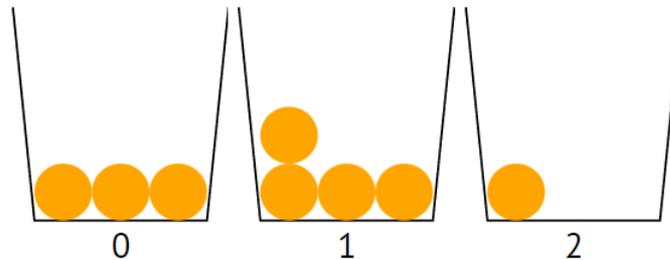
Programme bearbeiten

Gesucht: Programm, das die Zahl in Becher 1 zu den Zahlen in Becher 0 und Becher 2 addiert

vor der Ausführung des Programms:

Daten:

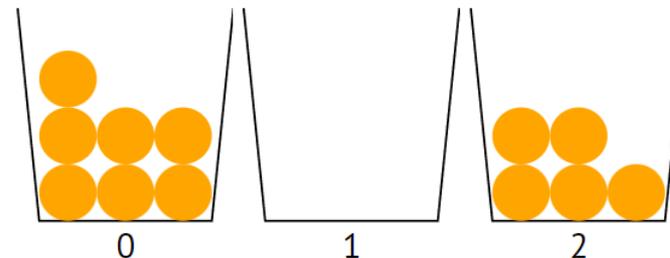
0	3
1	4
2	1



nach der Ausführung des Programms:

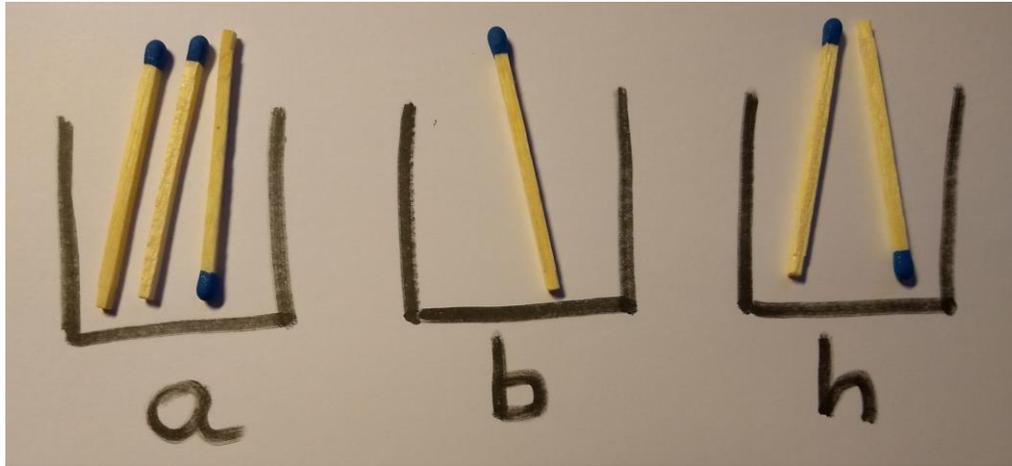
Daten:

0	7
1	0
2	5



Tipps zum Programmieren

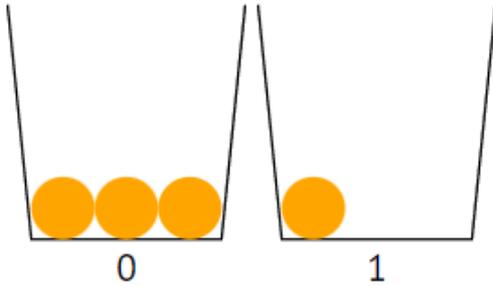
- Ideen zunächst mit Murmeln / Streichhölzern / ... durchspielen:



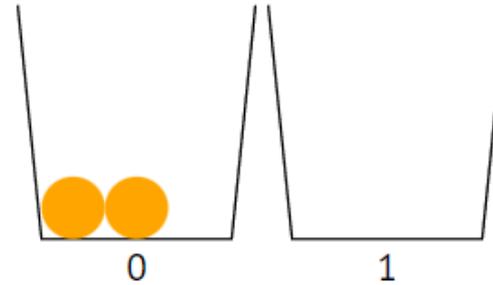
- Algorithmus zunächst in Pseudocode oder Python schreiben, dann übersetzen in Assembler
- zunächst einen Programmablaufplan zeichnen
- ein (evtl. unvollständiger) Programmablaufplan als Hilfestellung

1. Schreiben Sie ein Programm, das zwei Zahlen subtrahiert.

vorher:

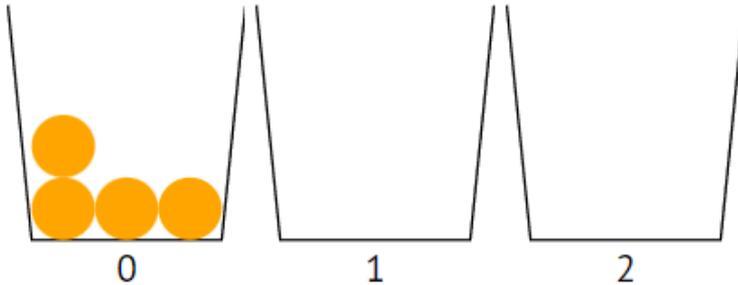


nachher:

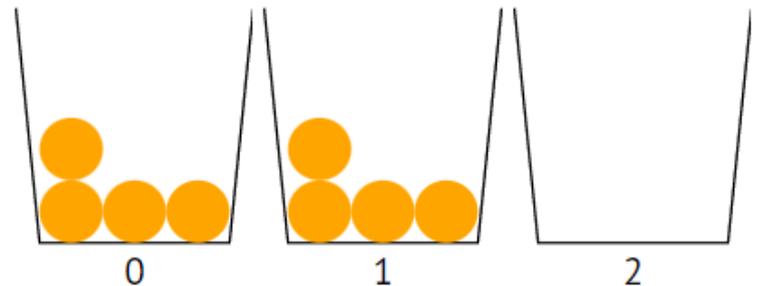


2. Schreiben Sie ein Programm, das eine Zahl kopiert.

vorher:



nachher:



Programme übersetzen

https://inf-schule.de/tools/bonsai_uebersetzer

Übersetzt Assembler-Programme in die Maschinensprache des Bonsai-Simulators

Programm:	Daten:	Zwischensprache	Maschinensprache
0 tst 1	0 3	0 tst 8	0 40008
1 jmp 3	1 4	1 jmp 3	1 30003
2 jmp 6		2 jmp 6	2 30006
3 dec 1		3 dec 8	3 20008
4 inc 0		4 inc 7	4 10007
5 jmp 0		5 jmp 0	5 30000
6 hlt		6 hlt	6 50000
		7 3	7 3
		8 4	8 4

Startadresse der Daten automatisch (direkt nach den Befehlen)

übersetzen

Übersetzung erfolgreich

Fragen?

Vorschläge?

Wünsche?

Vielen Dank für Ihre Aufmerksamkeit!