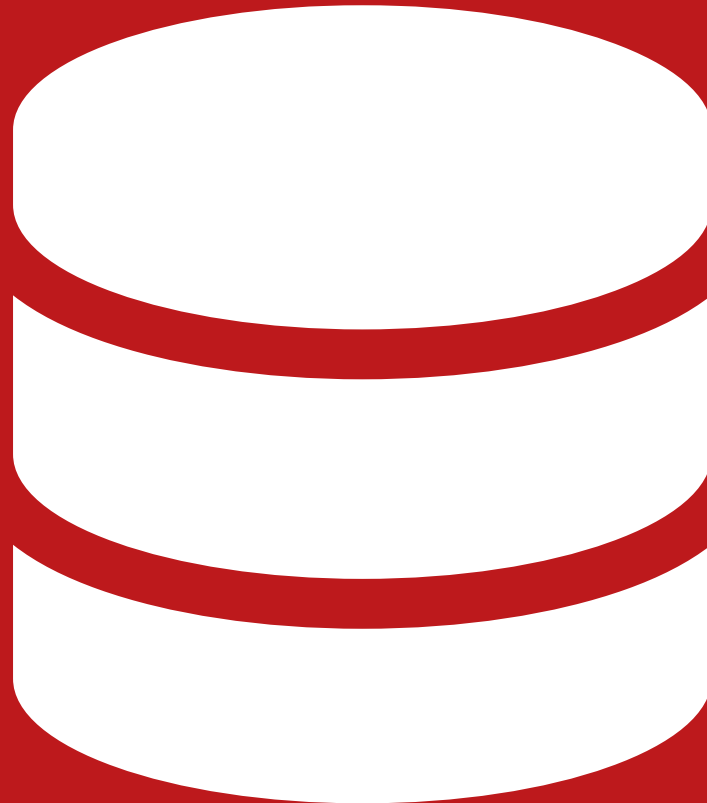




GRUNDLAGEN DER DIGITALISIERUNG

Digitale Schaltungen spielerisch
und praktisch erlebbar machen



FAHRPLAN

1. Verortung im Rahmenplan
2. Fachwissenschaftlicher Abriss
3. Spiele
4. Simulationstools
5. Spielen

VERORTUNG IM RAHMENPLAN



KLASSE 10 REGIO

Klasse 10

ca. 20 Unterrichtsstunden

Digitalisierung in meiner Umgebung untersuchen
[MD] [BO] [BNE] [DRF] [PG]

ca. 20 Unterrichtsstunden

Die Schülerinnen und Schüler untersuchen anhand eines geeigneten Kontexts Auswirkungen der Digitalisierung in ihrem unmittelbaren persönlichen Umfeld. Sie betrachten in diesem Zusammenhang auf Basis der zugrundeliegenden informatischen Konzepte Fragen der Sicherheit sowie soziale, rechtliche, ethische, ökologische und ökonomische Aspekte.

ca. 12 Unterrichtsstunden

Verbindliche Ziele und Inhalte	Hinweise und Anregungen
Aspekte der Digitalisierung verstehen	Die Aspekte sollten anschaulich und enaktiv-haptisch thematisiert werden.
- Eignung binärer Signale für die maschinelle Verarbeitung begründen - Bits logisch verknüpfen - binäre Addition	
Prinzip der asymmetrischen Verschlüsselung hinsichtlich Integrität, Vertraulichkeit und Authentizität untersuchen	Die Schülerinnen und Schüler erkennen, dass die Verwendung eines Paares aus privatem und öffentlichem Schlüssel den Nachteil der symmetrischen Verschlüsselung – den geheimen Austausch eines gemeinsamen Schlüssels über eine öffentliche Verbindung – aufhebt.



KLASSE 10 GYM

Grundlagen der Digitalisierung [MD] [BO] [BNE] [DRF] [PG]

ca. 8 Unterrichtsstunden

Mit dem von-Neumann-Modell lernen die Schülerinnen und Schüler den Aufbau und die Funktionsweise von Informatiksystemen kennen und verstehen die Programmierbarkeit als zentrales Wirkprinzip von Informatiksystemen.

Verbindliche Ziele und Inhalte	Hinweise und Anregungen
den Aufbau und die grundlegende Funktionsweise eines Informatiksystems nach dem von-Neumann-Modell beschreiben	Die Schülerinnen und Schüler nutzen ein Werkzeug zur Simulation maschinennaher Programmierung mit Hilfe eines vereinfachten von-Neumann-Modells. Sie erkennen die Zweckmäßigkeit der Verwendung höherer Programmiersprachen im Vergleich zur Maschinsprache.
Eignung binärer Signale für die maschinelle Verarbeitung erläutern • Bits logisch verknüpfen • binäre Addition	Die Prinzipien sind anschaulich mit einem Verweis für Möglichkeiten einer technischen Umsetzung zu vermitteln.
eine Grenze der binären Zahlendarstellung erklären	Die Grenzen können z. B. anhand der Wertebereiche ganzer Zahlen oder der eingeschränkten Abbildbarkeit von Dezimalbrüchen im Binärsystem verdeutlicht werden.

33



SEK II

Rechnerarchitektur
von-Neumann-Rechner [MD5] [Geschichte]

ca. 6/10 Unterrichtsstunden
ca. 6/10 Unterrichtsstunden

Verbindliche Ziele und Inhalte	Hinweise und Anregungen
von-Neumann-Rechner als Struktur- und Arbeitsmodell • die Struktur mit Zentraleinheit, bestehend aus Steuerwerk mit Befehlszähler und Befehlsregister, Rechenwerk mit Akkumulator, Speicherwerk, Ein- und Ausgabewerk sowie Bus beschreiben • die Arbeitsprinzipien beschreiben • den Zyklus der Befehlsabarbeitung erläutern • den von-Neumann-Flaschenhals als Grenze des von-Neumann-Rechners erklären • maschinennahe Programme in einer Simulation analysieren und das Systemverhalten erklären	Der Zyklus besteht aus den Phasen FETCH, DECODE, FETCH-OPERANDS, EXECUTE und WRITE BACK.
<i>zusätzlich für den Leistungskurs</i> • maschinennahe Programme in einer Simulation erweitern und entwickeln • Gemeinsamkeiten und Unterschiede zur Harvard-Architektur beschreiben	
• elementare, binär arbeitende Hardware-Komponenten auf Basis von Logik-Gattern mithilfe einer Simulation in Aufbau und Funktionsweise beschreiben - Halbaddierer - Flip-Flop	Der innere Aufbau eines Logik-Gatters ist nicht zu thematisieren.

FACHWISSENSCHAFTLICHER ABRISS

BINÄRCODIERUNG

- Beschränkung auf die Zeichen 0 und 1
- Ermöglicht Verarbeitung mittels Bauteilen, die als Schalter fungieren
- Binärzahl ist Zeichenkette bestehend aus Bits $c_i \in \{0,1\}$

$$Z_2 = c_{n-1}c_{n-2} \dots c_1c_0, c_{-1}c_{-2} \dots c_{-m+2}c_{-m+1}c_{-m}$$

- Dezimaldarstellung wird durch Gewichtung der Bits
 $z_{10} = g(z_2) = c_{n-1}2^{n-1} + c_{n-2}2^{n-2} + \dots + c_12^1 + c_02^0, c_{-1}2^{-1} + \dots + c_{-m}2^{-m}$



GANZZAHLIGE ADDITION

- Stellenweise Addition der Zahlen A und B mit entsprechenden Bits und Überträgen c_i

		0	1	1	1	1	1	1	0
+		0	0	1	1	0	1	0	1
Übertrag		1	1	1	1	1	0	0	
=		1	0	1	1	0	0	1	1

a_n	b_n	c_{n-1}	c_n	s
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

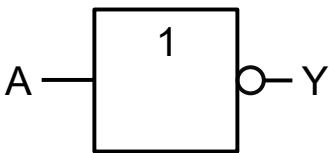
- Einerkomplement: Bits umdrehen
- Zweierkomplement: 1 addieren
- dann addieren
- letzter Übertrag wird ignoriert

$$\begin{array}{rcccccl}
& & 0 & 1 & 1 & 1 & 7_{10} \\
+ & & 1 & 1 & 0 & 1 & -3_{10} \\
\hline
= & 1 & 0 & 1 & 0 & 0 & \\
- & 1 & 0 & 0 & 0 & 0 & \\
\hline
= & & 0 & 1 & 0 & 0 & 4_{10}
\end{array}$$

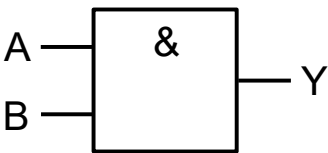


LOGIKGATTER

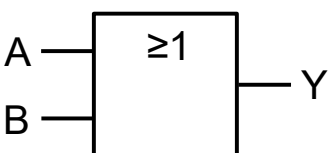
— Logikgatter realisieren logische Funktionen in elektronischen Schaltungen



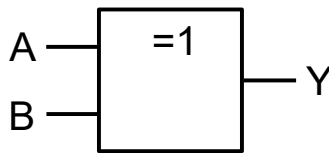
A	Y
0	1
1	0



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



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

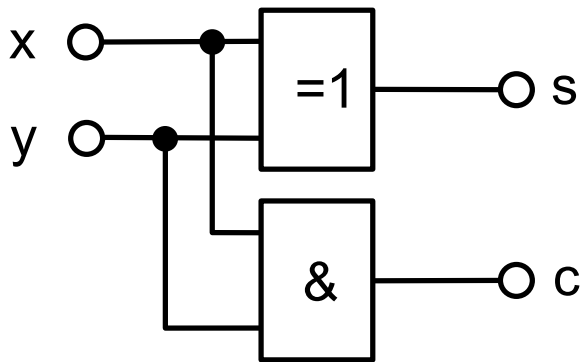


A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0



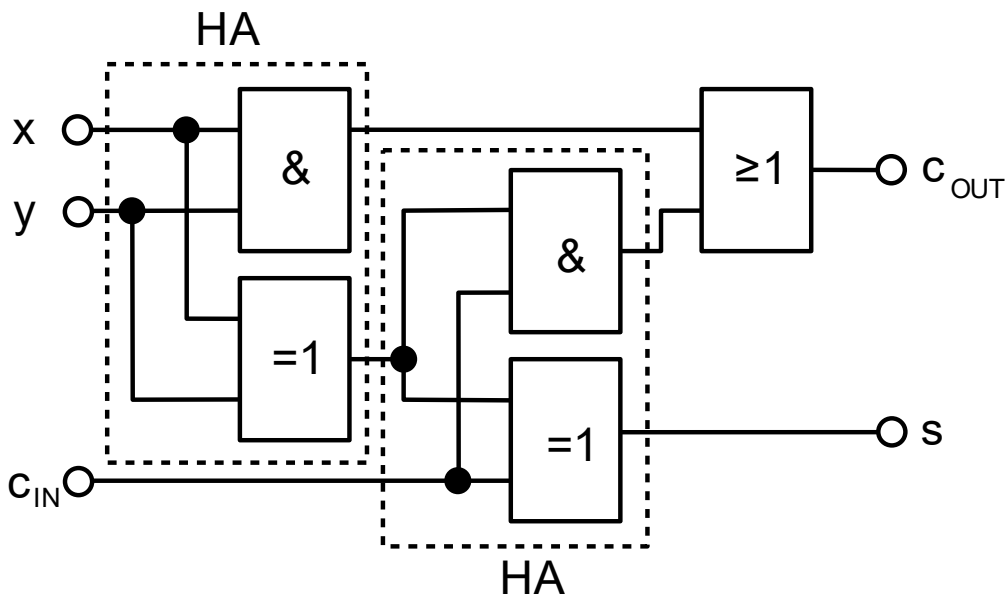
ADDIERER

Halbaddierer



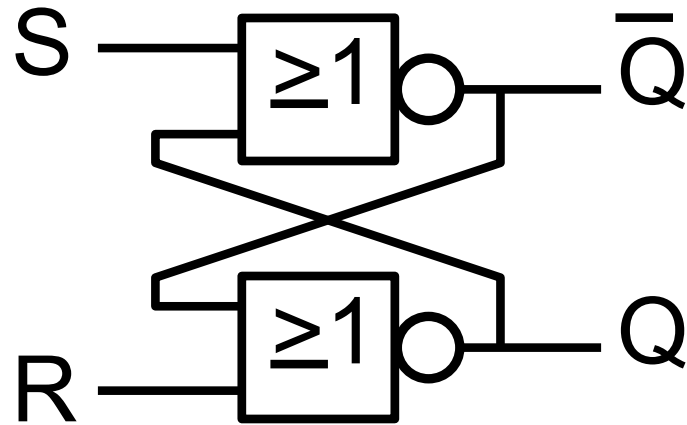
x	y	Übertrag c	Summe s
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Volladdierer



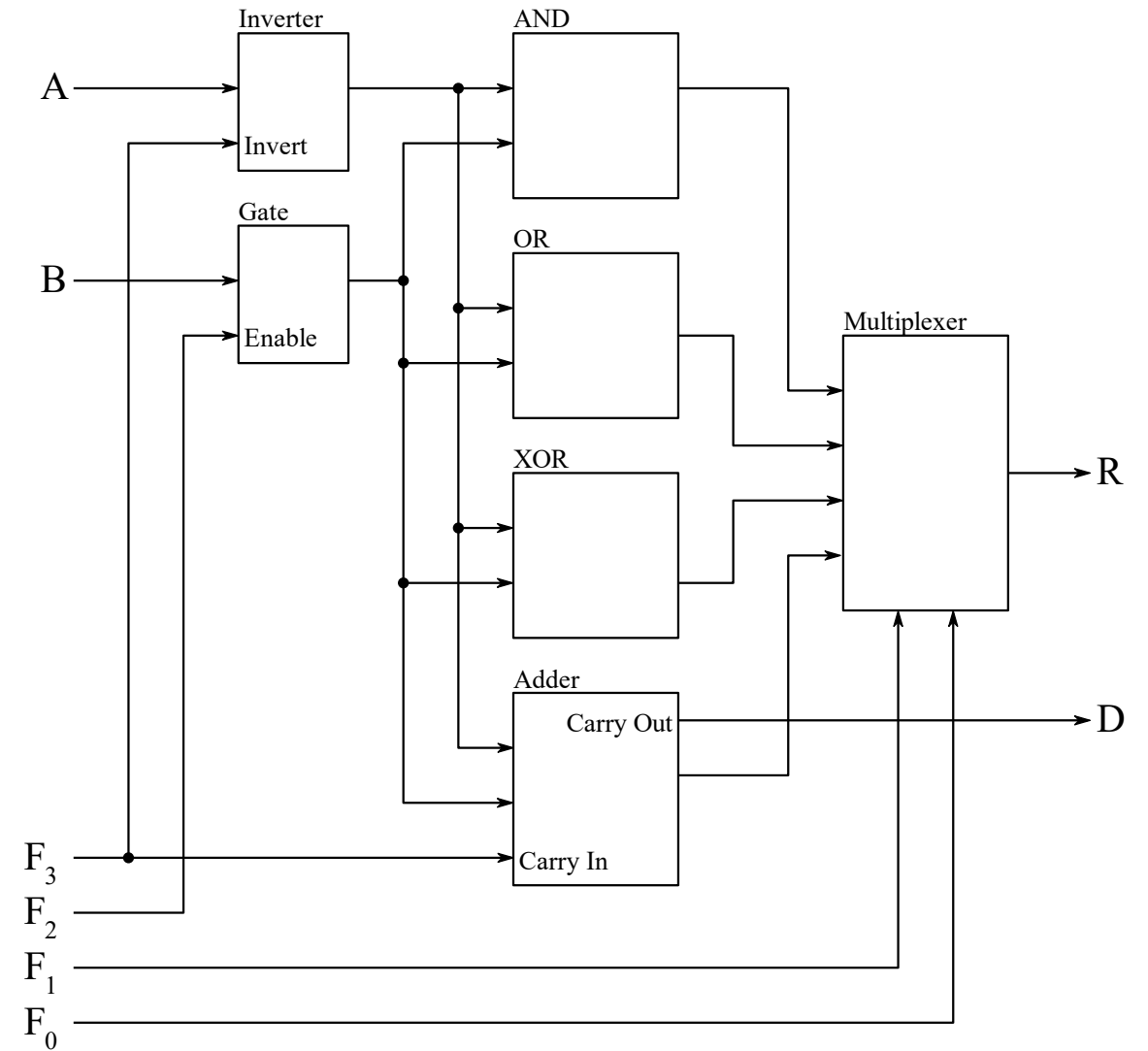
SR FLIP-FLOPS

- Zwei Stabile Ausgangszustände zur Speicherung eines Bits



ALU

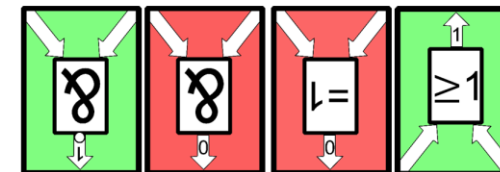
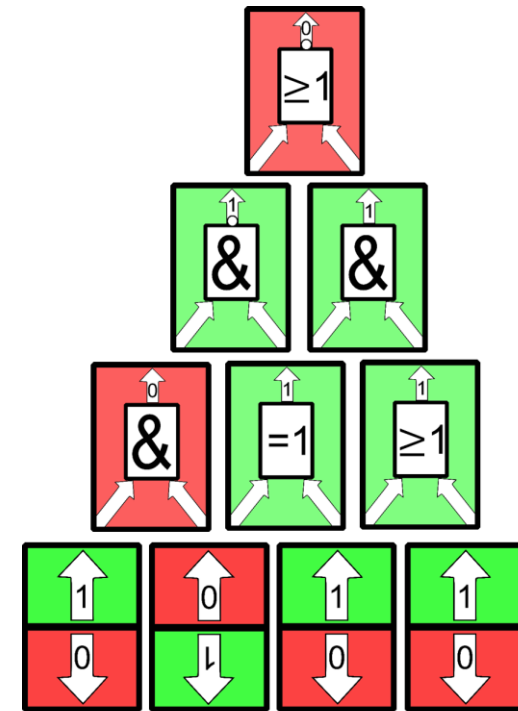
- Arithmetisch-Logische Einheit
- Bietet verschiedene Funktionen durch Auswahl mit Steuereingängen



SPIELE

DIGITALO

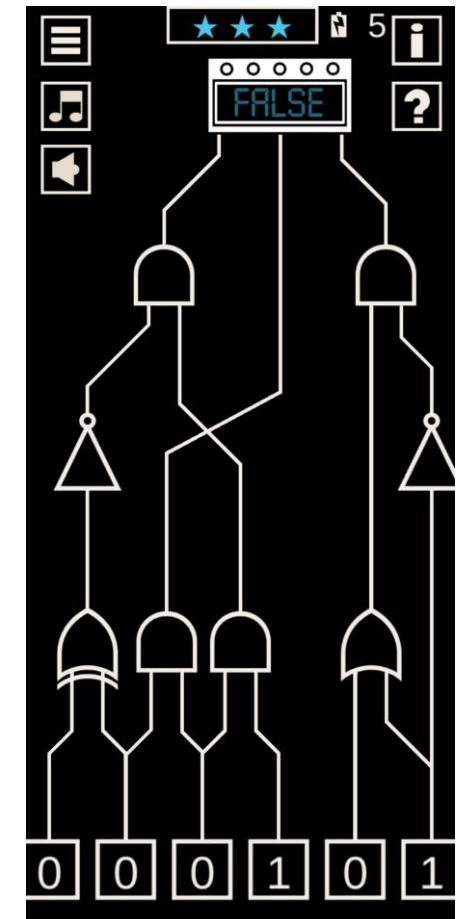
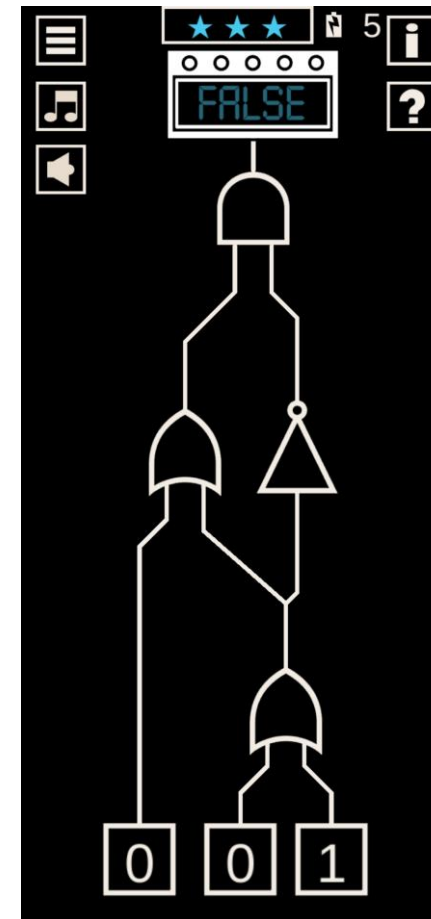
- Kartenspiel von inf-schule
- Frei download- und druckbar



<https://inf-schule.de/rechner/digitaltechnik/gatter/digitalo>

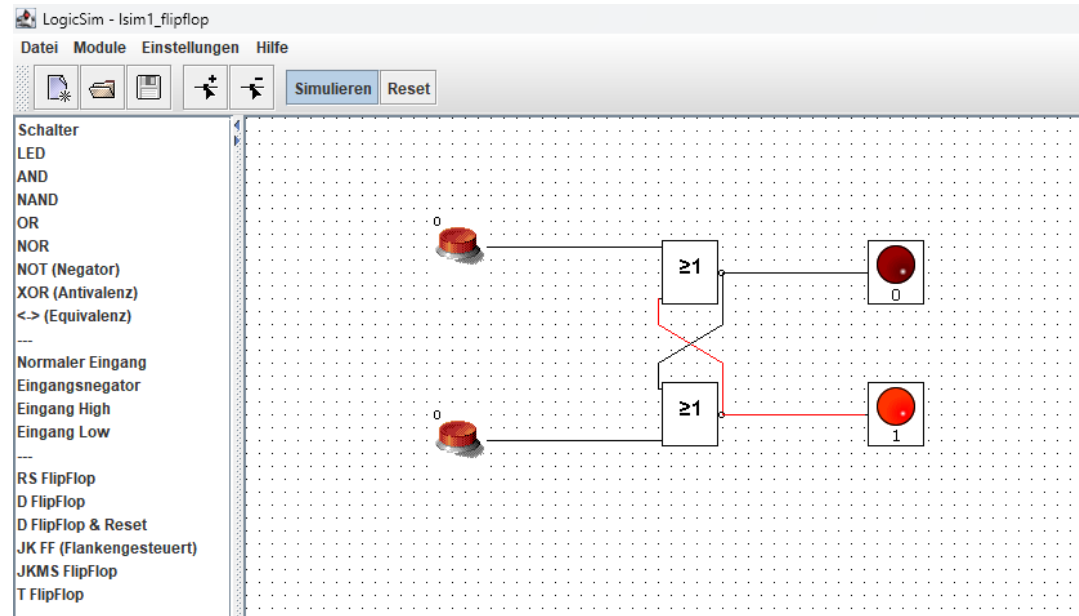
MAKE IT TRUE

- Mobile Game mit Logik-Puzzeln
 - <https://apps.apple.com/us/app/make-it-true-solve-circuits/id1536287319>
 - <https://play.google.com/store/apps/details?id=com.ViacheslavRud.Circuit&hl=gs>
- Problematisch: Werbung
- Ergo: Nachbauen in anderer Software



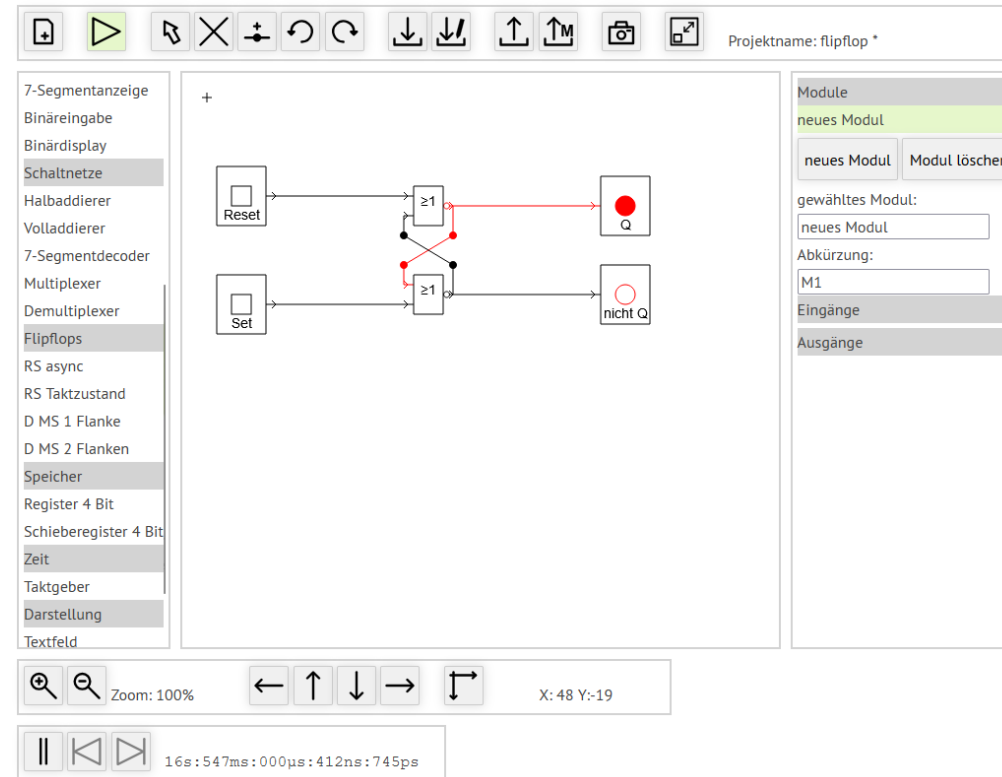
SIMULATIONSTOOLS

LOGICSIM



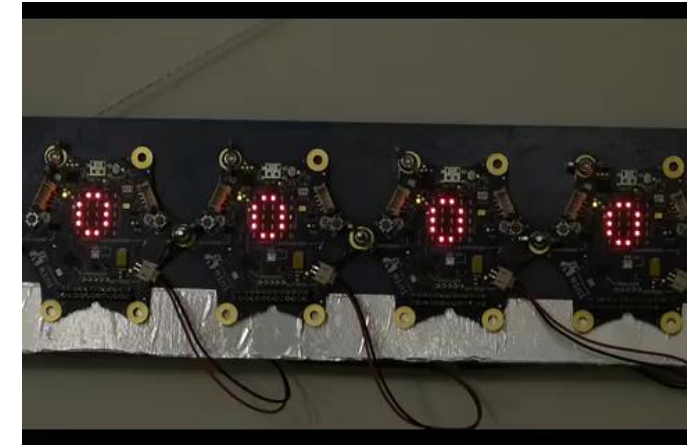
- LogicSim http://www.tetzi.de/java_logic_simulator_de.html
- LogicSim3 <https://github.com/codepiet/LogicSim3/tree/master/release>
- Java Installation benötigt

DSIMWEB



— https://inf-schule.de/rechner/digitaltechnik/Simulatoren/DSimWeb/DSimWeb_Vollversion

CALLIOPE

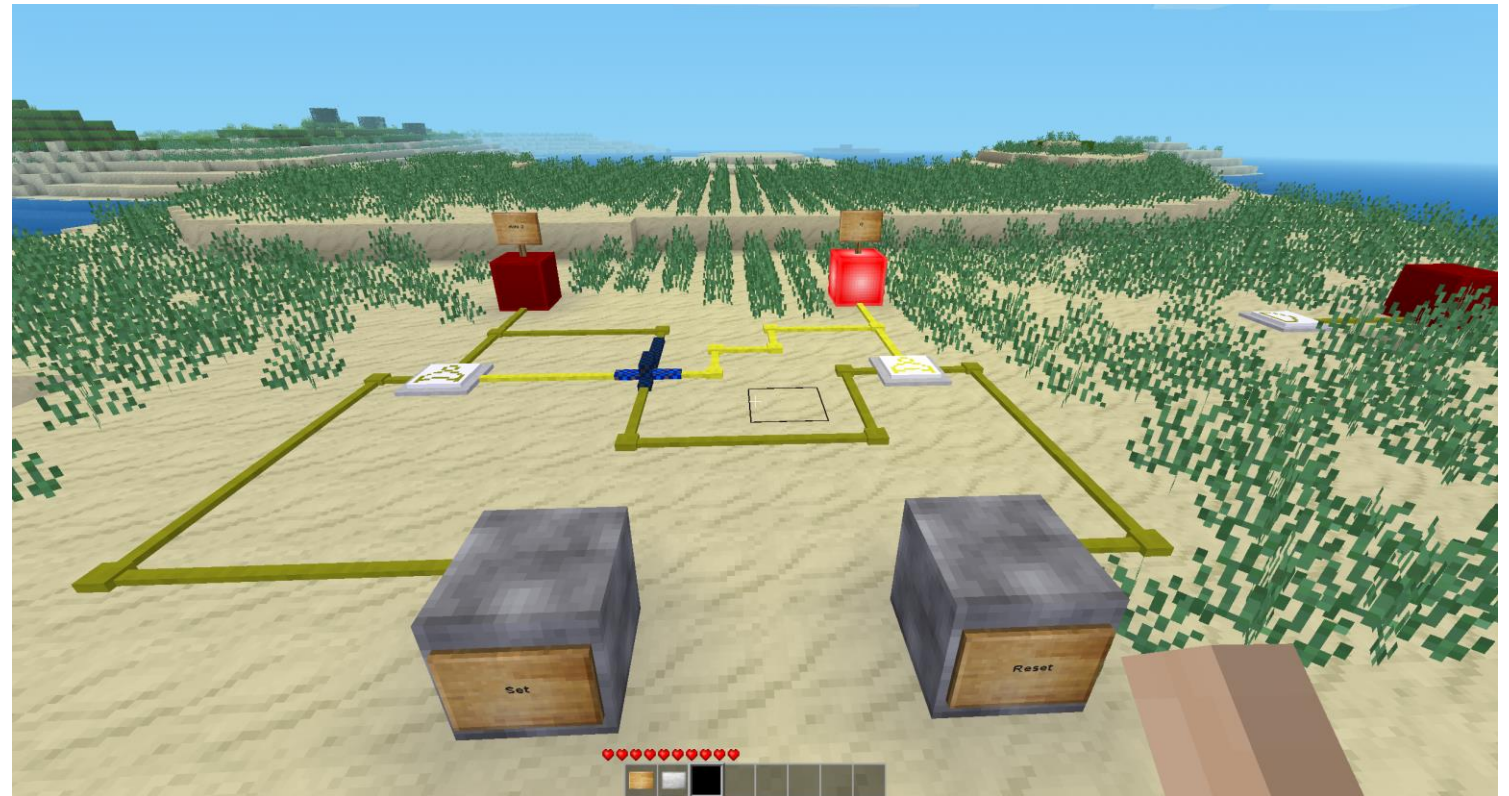


https://calliopemini.info/ca_69.php

- Möglichkeiten zur Programmierung einzelner Gatter oder
- Programmierung eines kompletten Gattersimulators
- Kombination mehrerer Calliope-Gatter durch Krokodilklemmen
- durch SchülerInnen oder vorgegeben
- Fertige Ideen und Programme beispielsweise unter
 - <https://calliopemini.info/> → logische Schaltungen
 - <https://www.hackster.io/ToniTaste/logik-gatter-und-schaltungen-simulieren-ea2e17>

MINETEST

- OpenSource Minecraft-Klon
<https://www.minetest.net/>
- mesecons Mod bringt Gatter und Strom
<https://mesecons.net/>





QUELLEN

- Die verwendeten Icons von Font Awesome stehen unter CC BY 4.0 license