

# Grundlagen der Digitalisierung

17. März 2021

## 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 Maschinensprache.
Eignung binärer Signale für die maschinelle Verarbeitung erläutern <ul style="list-style-type: none"> <li>• Bits logisch verknüpfen</li> <li>• binäre Addition</li> </ul>	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.
Auswirkung der Digitalisierung erkennen <ul style="list-style-type: none"> <li>• Chancen und Risiken der Nutzung von konkreten alltäglichen Informatiksystemen erkennen und bewerten</li> <li>• aus den Folgen der Digitalisierung Rückschlüsse für das eigene Verhalten ziehen</li> <li>• gesellschaftliche Folgen der Digitalisierung beurteilen und bewerten</li> </ul>	Die Schülerinnen und Schüler diskutieren aktuelle Fragestellungen zur Digitalisierung. Mögliche Themenfelder sind <ul style="list-style-type: none"> <li>• Ausfall- und Datensicherheit im privaten, beruflichen und wirtschaftlich-technischen Umfeld,</li> <li>• digitale Identität und digitaler Nachlass,</li> <li>• Nachhaltigkeit digitaler Daten,</li> <li>• Grenzen komplexer Informatiksysteme.</li> </ul>

# Codierung – Binär-, Dezimal- und Hexadezimalsystem

The screenshot shows the Microsoft MakeCode IDE for the Calliope mini. The top navigation bar includes 'CALLIOPE mini', 'Startseite', 'Teilen', 'Blöcke', 'JavaScript', and the Microsoft logo. The left sidebar contains a search bar and a list of component categories: Grundlagen, Eingabe, Musik, LED, Funk, Schleifen, Logik, Variablen, Mathematik, Motoren, and Fortgeschritten. Below the sidebar is a 'Herunterladen' button and a 'Nachfolger' field.

The main workspace displays two event-driven code blocks:

```

wenn Knopf A gedrückt
  setze zahl auf wähle eine zufällige Zahl von 0 bis 9
  zeige Zahl zahl

```

```

wenn Knopf B gedrückt
  ändere zahl um 1
  zeige Zahl zahl

```

At the bottom of the workspace, there are navigation and execution controls, including a 'Nachfolger' field and buttons for undo, redo, and execution.

# Codierung – Binär-, Dezimal- und Hexadezimalsystem

```

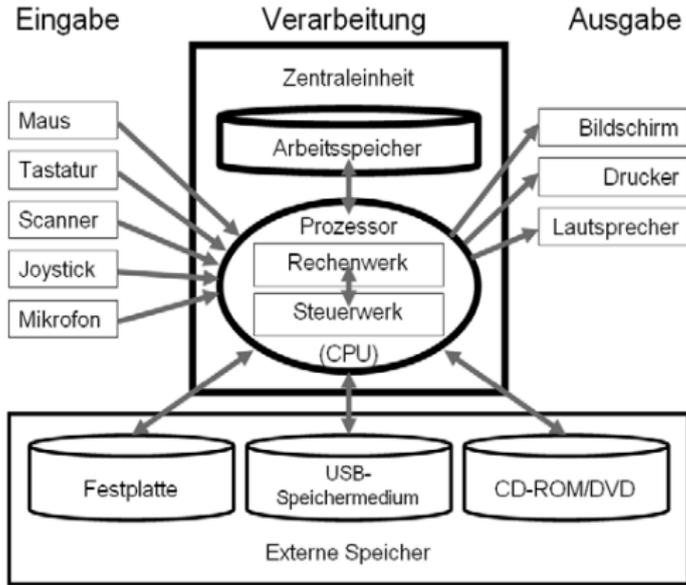
1  ☐:020000040000FA
2  :10000000C0070000D1060000D1000000B1060000CA
3  :10001000000000000000000000000000000000E0
4  :10002000000000000000000000000000005107000078
5  :100030000000000000000000DB000000E50000000
6  :10004000EF000000F900000003010000D010000B6
7  :1000500017010000210100002B0100003501000004
8  :100060003F01000049010000530100005D01000054
9  :1000700067010000710100007B01000085010000A4
10 :100080008F01000099010000A3010000AD010000F4
11 :10009000B7010000C1010000CB010000D501000044
12 :1000A000DF010000E9010000F3010000FD01000094
13 :1000B00007020000110200001B02000025020000E0
14 :1000C0001FB5C046C04600F0EFFA04B00FB41FBD24
15 :1000D00008205A49096809580847382057490968CB
16 :1000E000095808473C2055490968095808474020E5
17 :1000F0005249096809580847442050490968095875
18 :10010000084748204D490968095808474C204B4981
19 :10011000096809580847502048490968095808479C
20 :100120005420464909680958084758204349096836

```

# von-Neumann-Modell

Verbindliche Ziele und Inhalte	Hinweise und Anregungen
<p>den Aufbau und die grundlegende Funktionsweise eines Informatiksystems nach dem von-Neumann-Modell beschreiben</p>	<p>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 Maschinensprache.</p>

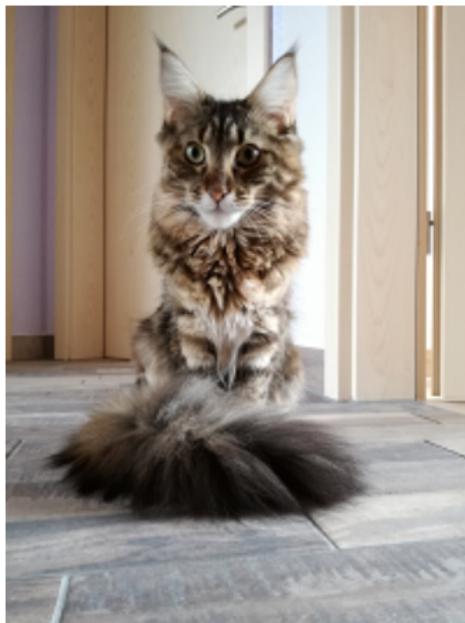
# EVAS-Prinzip



1

<sup>1</sup>Dauscher (2013): Aufbau und Funktion eines von-Neumann-Rechners – Ein möglicher Unterrichtsgang mit dem Open-Source-Simulator Johnny.

# EVAS-Prinzip



# EVAS-Prinzip



# Lochkarten



# von-Neumann-Modell

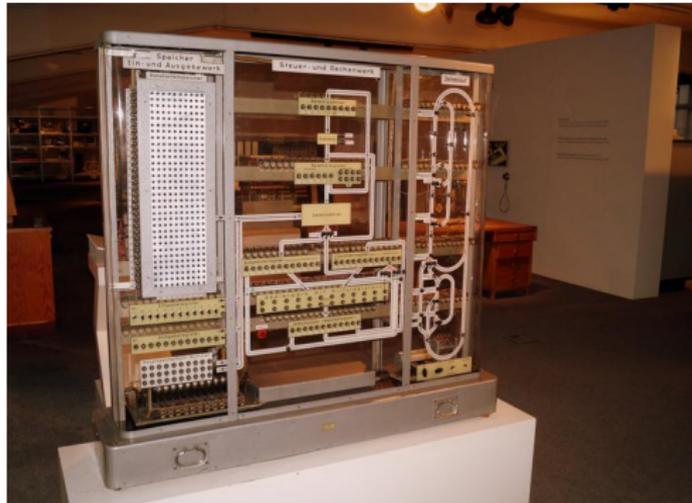


2

---

<sup>2</sup>John von Neumann (1903 - 1957) (<https://upload.wikimedia.org/wikipedia/commons/5/5e/JohnvonNeumann-LosAlamos.gif>)

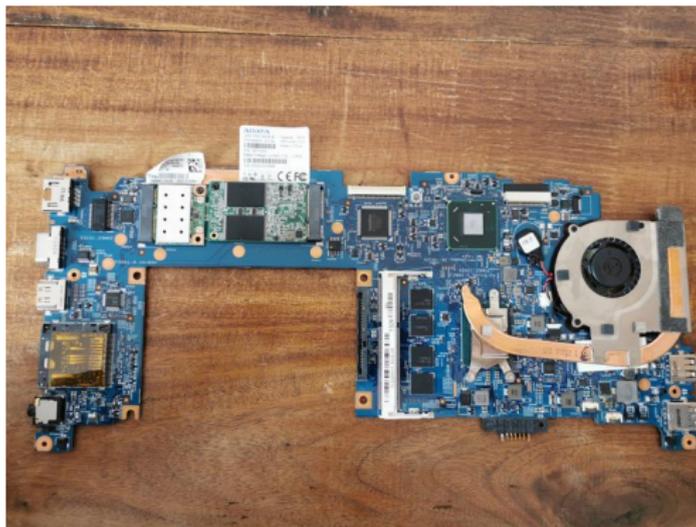
# Modellrechner



3

<sup>3</sup>Modellrechnerautomat (1958) mit von-Neumann-Architektur in den  
nationalen Sammlungen Dresden  
([https://commons.wikimedia.org/wiki/File:  
Modellrechenautomat\\_TH\\_Dresen\\_1958\\_TSD.jpg](https://commons.wikimedia.org/wiki/File:Modellrechenautomat_TH_Dresen_1958_TSD.jpg) )

# Mainboard

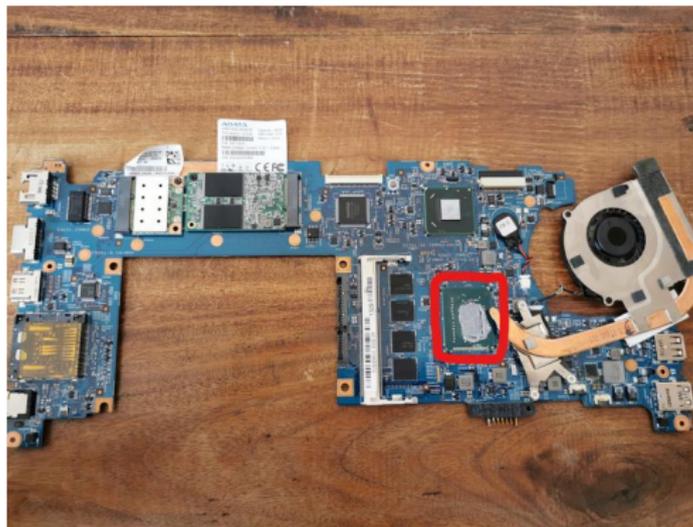


4

<sup>4</sup>Mainboard eines Sony Vaio SV-T1313L1ES (2013)



# Mainboard



5

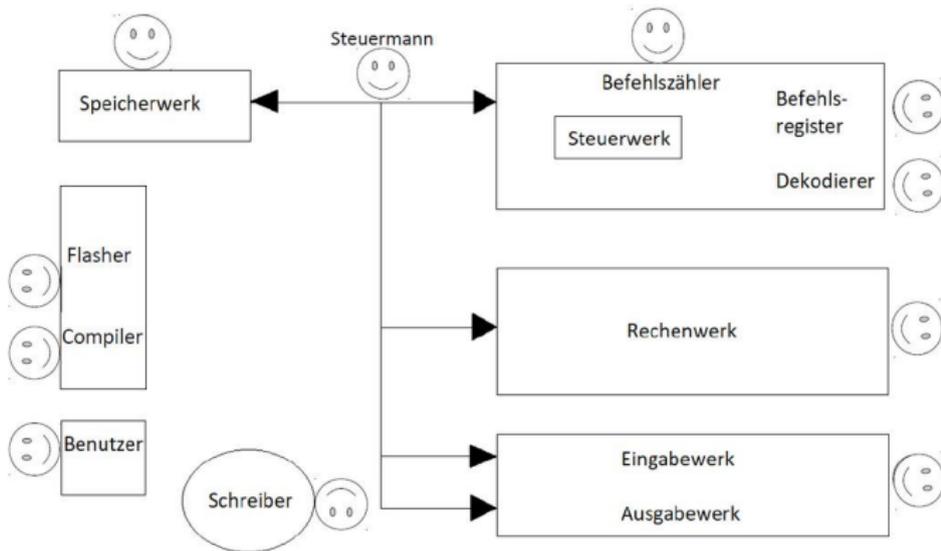
<sup>5</sup>Mainboard eines Sony Vaio SV-T1313L1ES (2013)



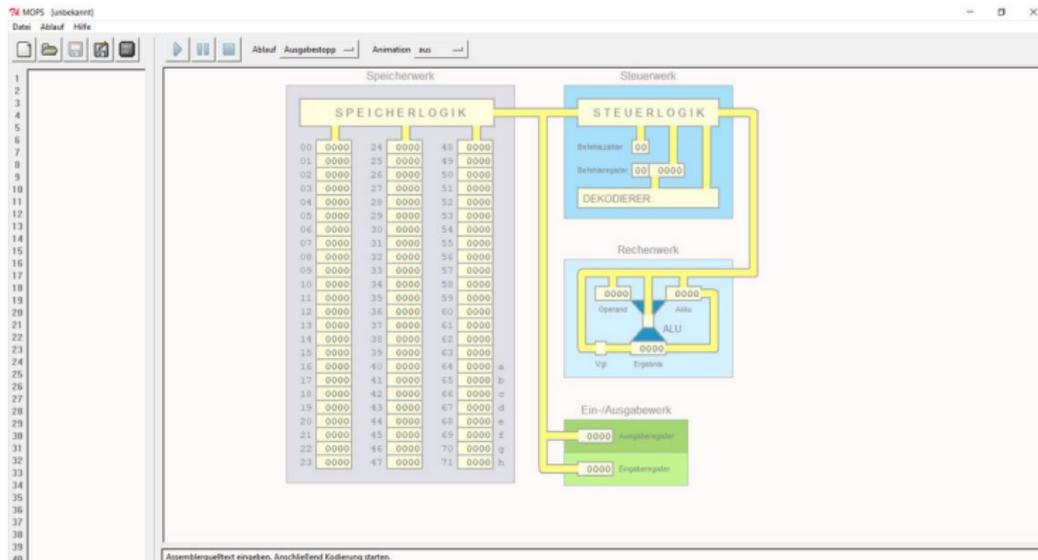
# von-Neumann-Modell

Verbindliche Ziele und Inhalte	Hinweise und Anregungen
<p>den Aufbau und die grundlegende Funktionsweise eines Informatiksystems nach dem von-Neumann-Modell beschreiben</p>	<p>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.</p>

# von-Neumann-Modell



# M0dellrechner mit PSeudoassembler



6

<sup>6</sup>Haase (2013): Modellrechner MOPS.

<http://www.viktorianer.de/info/mops.html>



# Codierung – Binär-, Dezimal- und Hexadezimalsystem

CALLIOPE mini Startseite Teilen Blöcke JavaScript Microsoft

Suche...

- Grundlagen
- Eingabe
- Musik
- LED
- Funk
- Schleifen
- Logik
- Variablen
- Mathematik
- Motoren
- Fortgeschritten

Herunterladen Nachfolger

```
wenn Knopf A gedrückt  
  setze zahl auf wähle eine zufällige Zahl von 0 bis 9  
  zeige Zahl zahl  
  
wenn Knopf B gedrückt  
  ändere zahl um 1  
  zeige Zahl zahl
```

# Codierung – Binär-, Dezimal- und Hexadezimalsystem

```
1  ☐:020000040000FA
2  :10000000C0070000D1060000D1000000B1060000CA
3  :10001000000000000000000000000000000000E0
4  :10002000000000000000000000000000005107000078
5  :100030000000000000000000DB000000E500000000
6  :10004000EF000000F900000003010000D010000B6
7  :1000500017010000210100002B0100003501000004
8  :100060003F01000049010000530100005D01000054
9  :1000700067010000710100007B01000085010000A4
10 :100080008F01000099010000A3010000AD010000F4
11 :10009000B7010000C1010000CB010000D501000044
12 :1000A000DF010000E9010000F3010000FD01000094
13 :1000B00007020000110200001B02000025020000E0
14 :1000C0001FB5C046C04600F0EFFA04B00FB41FBD24
15 :1000D00008205A49096809580847382057490968CB
16 :1000E000095808473C2055490968095808474020E5
17 :1000F0005249096809580847442050490968095875
18 :10010000084748204D490968095808474C204B4981
19 :10011000096809580847502048490968095808479C
20 :100120005420464909680958084758204349096836
```

# Codierung – Binär-, Dezimal- und Hexadezimalsystem

```
:020000021000EC
:10010000214601360121470136007EFE09D2190140
:100110002146017EB7C20001FF5F16002148011988
:10012000194E79234623965778239EDA3F01B2CA7
:100130003F0156702B5E712B722B732146013421C7
:00000001FF
```

- Startcode
- Byte count
- Adresse
- Typ
- Datenfeld
- Prüfsumme

Die Prüfsumme für den ersten Beispiel-Datensatz berechnet sich wie folgt:  $10 + 01 + 00 + 00 + 21 + 46 + 01 + 36 + 01 + 21 + 47 + 01 + 36 + 00 + 7E + FE + 09 + D2 + 19 + 01 = 3C0 \rightarrow C0 \rightarrow -C0 + 1 = 40$ .

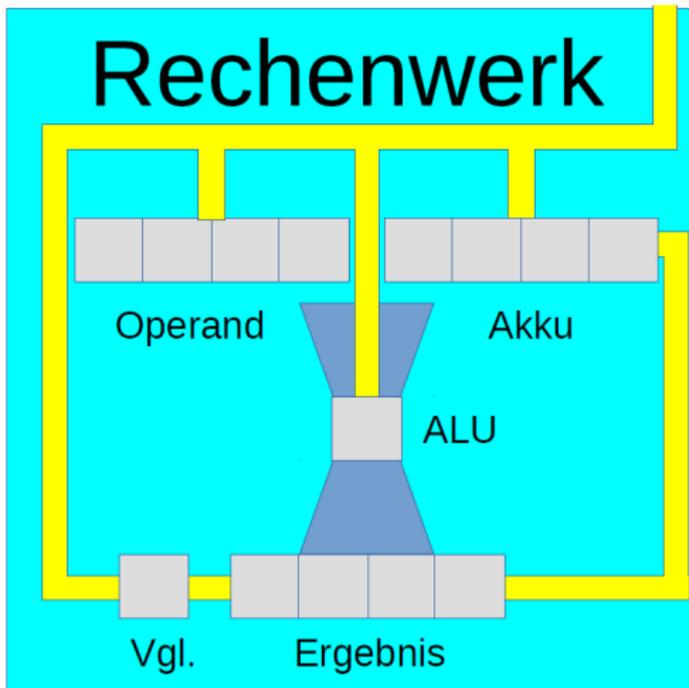
7

---

<sup>7</sup>[https://de.wikipedia.org/wiki/Intel\\_HEX](https://de.wikipedia.org/wiki/Intel_HEX)



# binäre Signale



# binäre Signale

Eignung binärer Signale für die maschinelle Verarbeitung erläutern

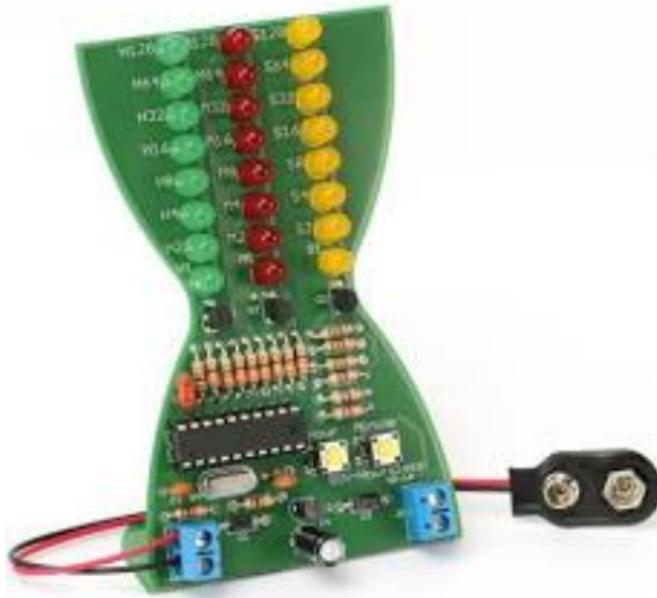
- Bits logisch verknüpfen
- binäre Addition

eine Grenze der binären Zahlendarstellung erklären

Die Prinzipien sind anschaulich mit einem Verweis für Möglichkeiten einer technischen Umsetzung zu vermitteln.

Die Grenzen können z. B. anhand der Wertebereiche ganzer Zahlen oder der eingeschränkten Abbildbarkeit von Dezimalbrüchen im Binärsystem verdeutlicht werden.

# Codierung – Binär-, Dezimal- und Hexadezimalsystem



# Codierung – Binär-, Dezimal- und Hexadezimalsystem

Binär	Dezimal	Hexadezimal
00000000	0	0
00000001	1	1
00000010	2	2
00000011	3	3
00000100	4	4
00000101	5	5
00000110	6	6
00000111	7	7
00001000	8	8
00001001	9	9
00001010	10	A
00001011	11	B
00001100	12	C
00001101	13	D
00001110	14	E
00001111	15	F
00010000	16	10
00010001	17	11
00010010	18	12
00010011	19	13
00010100	20	14
00010101	21	15
00010110	22	16
00010111	23	17
00011000	24	18
00011001	25	19
00011010	26	1A
00011011	27	1B
00011100	28	1C
00011101	29	1D
00011110	30	1E
00011111	31	1F
00100000	32	20
00100001	33	21
00100010	34	22
00100011	35	23
00100100	36	24
00100101	37	25
00100110	38	26
00100111	39	27
00101000	40	28
00101001	41	29
00101010	42	2A
00101011	43	2B
00101100	44	2C
00101101	45	2D
00101110	46	2E
00101111	47	2F
00110000	48	30
00110001	49	31
00110010	50	32
00110011	51	33
00110100	52	34
00110101	53	35
00110110	54	36
00110111	55	37
00111000	56	38
00111001	57	39
00111010	58	3A
00111011	59	3B
00111100	60	3C
00111101	61	3D
00111110	62	3E
00111111	63	3F
01000000	64	40
01000001	65	41
01000010	66	42
01000011	67	43
01000100	68	44
01000101	69	45
01000110	70	46
01000111	71	47
01001000	72	48
01001001	73	49
01001010	74	4A
01001011	75	4B
01001100	76	4C
01001101	77	4D
01001110	78	4E
01001111	79	4F
01010000	80	50
01010001	81	51
01010010	82	52
01010011	83	53
01010100	84	54
01010101	85	55
01010110	86	56
01010111	87	57
01011000	88	58
01011001	89	59
01011010	90	5A
01011011	91	5B
01011100	92	5C
01011101	93	5D
01011110	94	5E
01011111	95	5F
01100000	96	60
01100001	97	61
01100010	98	62
01100011	99	63
01100100	100	64

<sup>8</sup><http://www.abenteuer-informatik.de/schulmaterial.html>



# Codierung – Binär-, Dezimal- und Hexadezimalsystem



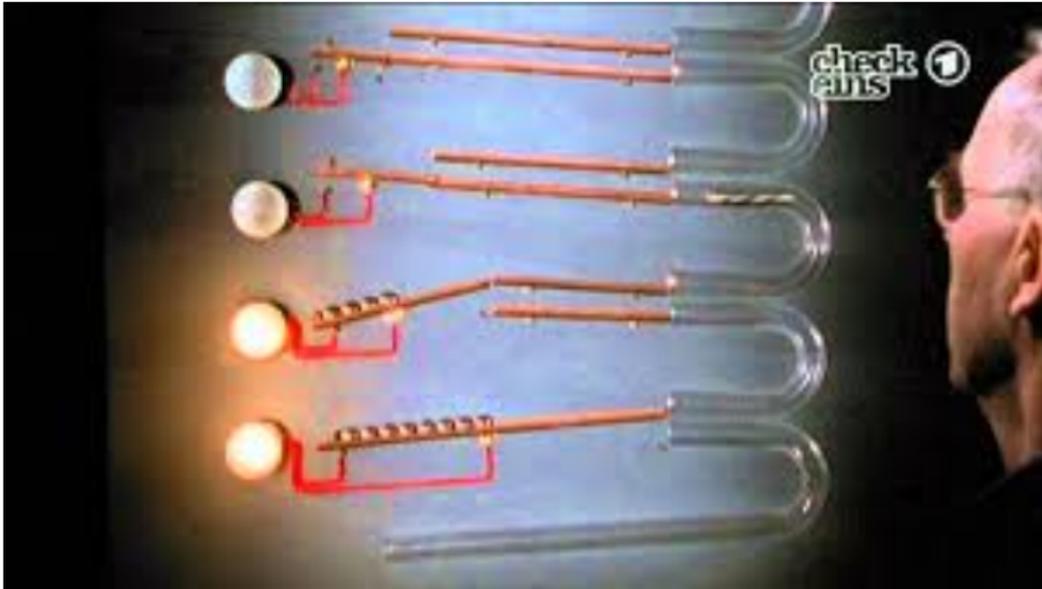
# Codierung – Binär-, Dezimal- und Hexadezimalsystem



9

<sup>9</sup><https://www.youtube.com/watch?v=GcDshWmhF4A>

# Codierung – Binär-, Dezimal- und Hexadezimalsystem



10

<sup>10</sup>[https://www.youtube.com/watch?v=91-1\\_dD6q-PQ](https://www.youtube.com/watch?v=91-1_dD6q-PQ)

# binäre Signale

LogicSim - Addition.Isim

Datei Module Einstellungen Hilfe

Simulieren Reset

Schalter  
LED  
AND  
NAND  
OR  
NOR  
NOT (Negator)  
XOR (Antivalenz)  
<-> (Äquivalenz)  
...  
Normaler Eingang  
Eingangsnegator  
Eingang High  
Eingang Low  
...  
RS FlipFlop  
D FlipFlop  
D FlipFlop & Reset  
JK FF (Flankengesteuert)  
JKMS FlipFlop  
T FlipFlop  
...  
Clock  
Monoflop

Praktische Aufgabe 1

Aufgabe: Teste das folgende System. Notiere deine Ergebnisse auf dem Arbeitsblatt.

$2^2 = 4$        $2^1 = 2$        $2^0 = 1$

+

# binäre Signale

LogicSim - Halbaddierer.Isim

Datei Module Einstellungen Hilfe

Simulieren Reset

Schalter  
LED  
AND  
NAND  
OR  
NOR  
NOT (Negator)  
XOR (Antivalenz)  
<-> (Äquivalenz)  
...  
Normaler Eingang  
Eingangsnegator  
Eingang High  
Eingang Low  
...  
RS FlipFlop  
D FlipFlop  
D FlipFlop & Reset  
JK FF (Flankengesteuert)  
JKMS FlipFlop  
T FlipFlop  
...  
Clock  
Monoflop  
Einschaltverzögerung  
Ausschaltverzögerung

Praktische Aufgabe 2

Aufgabe: Konstruiere eine Schaltung, die richtig addiert. Nutze die vorgegebenen Bauteile.

$2^0 = 1$

0

+

0

=1

&

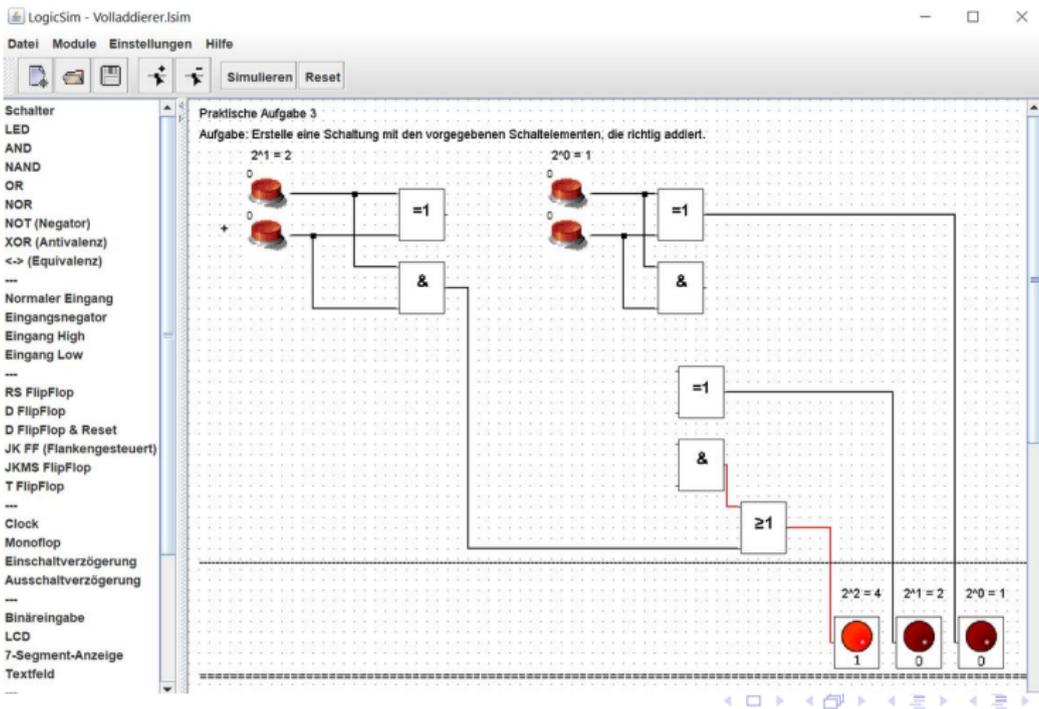
Übertrag

0

Summe

0

# binäre Signale



# binäre Signale

Suche...

CALLIOPE mini Startseite Teilen Blöcke JavaScript Microsoft

Grundlagen  
Eingabe  
Musik  
LED  
Funk  
Schleifen  
Logik  
Variablen  
Mathematik  
Motoren  
Fortgeschritten

beginne mit

setze Modus = auf 1  
setze Logikfunktion = auf 2  
Aufref zeigeGatter

Funktion zeigeGatter

Bildschirmtext löschen

wenn Logikfunktion == 1 dann  
Zeichne x 2 y 4 Rest von Logikfunktion + 1

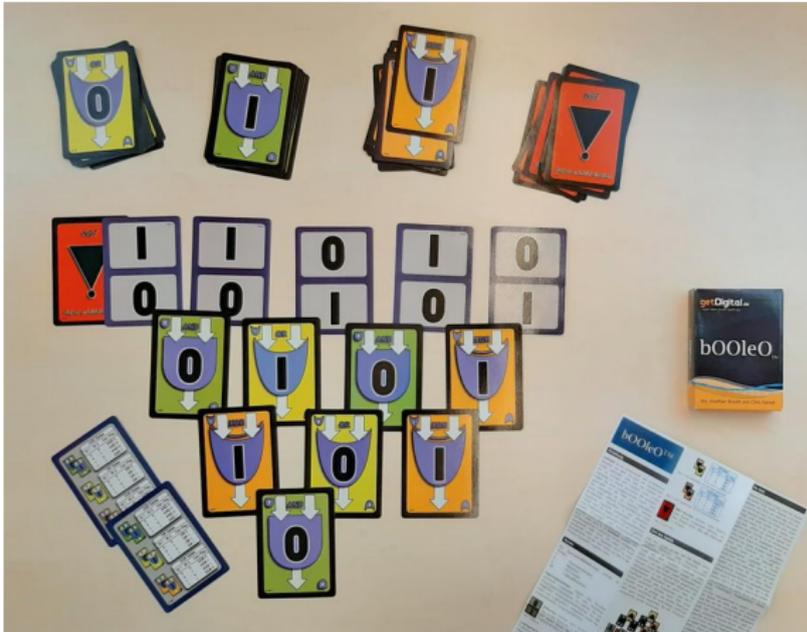
wenn Logikfunktion == 2 und Logikfunktion == 3 dann  
Zeichne x 2 y 1

wenn Logikfunktion == 4 dann  
Zeichne x 2 y 8

stoppe mit

Herunterladen Logikgatter\_SuS

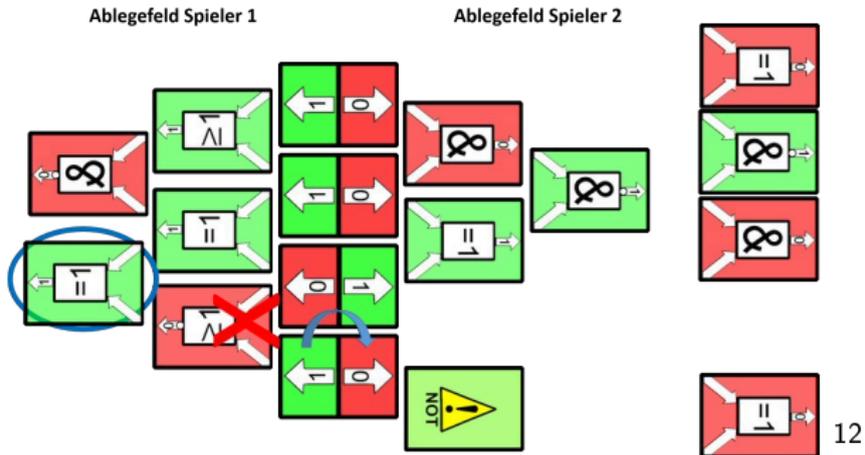
# binäre Signale



11

<sup>11</sup><https://www.getdigital.de/b00le0-Kartenspiel.html>

# binäre Signale



<sup>12</sup><https://www.inf-schule.de/rechner/digitaltechnik/gatter/digitalo>

# Auswirkungen der Digitalisierung

## Auswirkung der Digitalisierung erkennen

- Chancen und Risiken der Nutzung von konkreten alltäglichen Informatiksystemen erkennen und bewerten
- aus den Folgen der Digitalisierung Rückschlüsse für das eigene Verhalten ziehen
- gesellschaftliche Folgen der Digitalisierung beurteilen und bewerten

Die Schülerinnen und Schüler diskutieren aktuelle Fragestellungen zur Digitalisierung. Mögliche Themenfelder sind

- Ausfall- und Datensicherheit im privaten, beruflichen und wirtschaftlich-technischen Umfeld,
- digitale Identität und digitaler Nachlass,
- Nachhaltigkeit digitaler Daten,
- Grenzen komplexer Informatiksysteme.

# Auswirkungen der Digitalisierung

