FH JOANNEUM
University of Applied Sciences

Cloud Computing Lab

FFG
Forschung wirkt.

Das Internet – Folien.pptx
© 2024 by Mathias Knoll is licensed under CC BY-NC-ND 4.0.

Einführung in die Informatik

Mathias Knoll

Ein Vortrag des

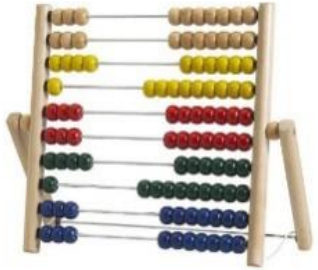
Cloud Computing Lab

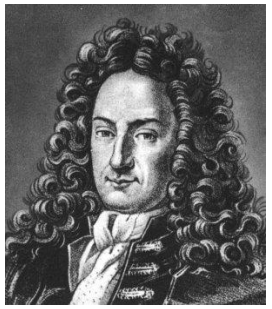
Gefördert von der

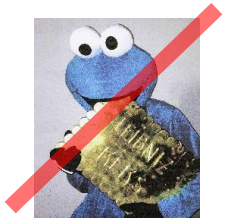
FFG
Forschung wirkt.

Anfänge: Informatik & Rechenmaschinen (1)

- Altertum: der **Abakus** wird als Rechenhilfe verwendet.
- 5. Jahrhundert: Entwicklung des **Dezimalsystems** in Indien.
- 1524: Adam Ries veröffentlicht ein **Rechenbuch über Rechengesetze** im Dezimalsystem.
- 1641: Blaise Pascal konstruiert eine **Maschine zu Addition sechsstelliger Zahlen**.
- 1674: Gottfried Wilhelm Leibniz konstruiert eine **Rechenmaschine für die vier Grundrechenarten**.



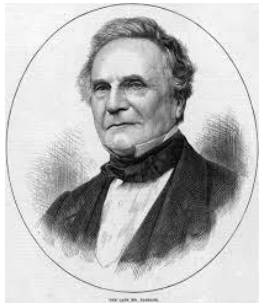




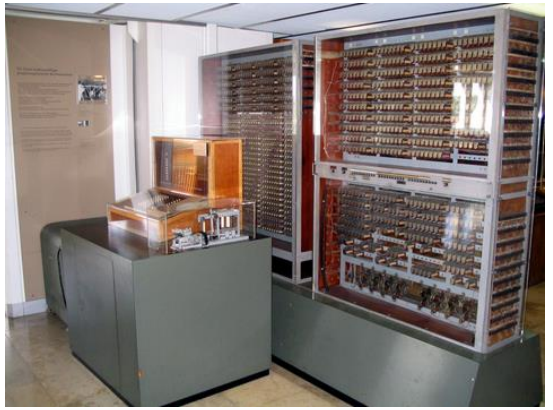
Bildquellen: ikea.com (abacus), wikipedia.org (adam ries), mathematik.ch (pascal, leibniz)

Anfänge: Informatik & Rechenmaschinen (2)

- 1838: Charles Babbage plant eine durch Lochkarten gesteuerte Rechenmaschine, die er „**Analytical Engine**“ nennt.



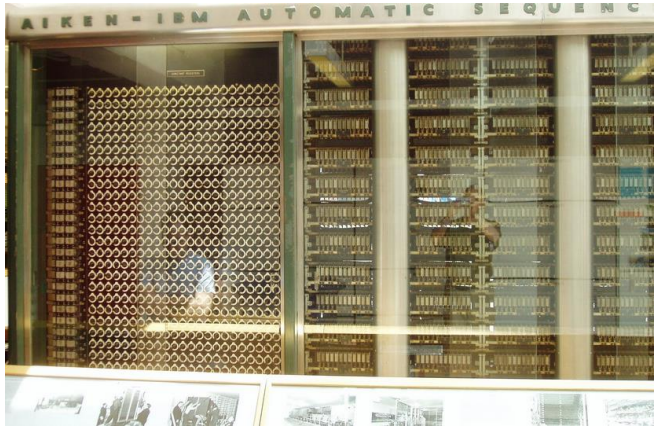
- 1941: Konrad Zuse stellt mit dem **elektro-mechanischem Z3 den ersten programm-gesteuerten Rechenautomat** vor
  - Er besitzt eine Speicherkapazität von 64 Worten a 22 Bit (=176 Byte) und kann
  - in 3 Sekunden multiplizieren, dividieren und Quadratwurzel ziehen.
  - Das Original ist zerstört, ein funktionsfähiger Nachbau von 1962 ist im Deutschen Museum

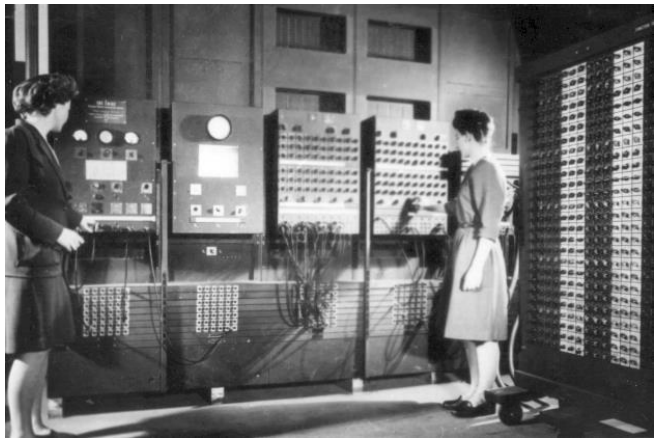


Bildquellen: wikipedia.org

Anfänge: Informatik & Rechenmaschinen (3)

- 1944: Howard Aiken erstellt in Zusammenarbeit mit der Harvard-University und der Firma IBM die teilweise **programmgesteuerte Rechenanlage MARK I**. Die Maschine ist 5 Tonnen schwer und benötigt 0,3 Sekunden für eine Addition.
- 1946: J.P. Eckert und J.W. Mauchly stellen den **ENIAC (electronic numerical integrator and calculator),** den ersten vollelektronischen Rechner, fertig.
- 1946-1952: Auf den Ideen **John von Neumanns** und weiterer Wissenschaftler des Institute of Advanced Study at Princeton, werden in Universitätslabors weitere Computer entwickelt.





Bildquellen: wikipedia.org

Anfänge: Informatik & Rechenmaschinen (4)

- 1951: **UNIVAC I (Universal Automatic Computer)**
  - Erster kommerziell erhältlicher Computer
  - Benutzt ein **Magnetband** als externen Speicher
  - Hersteller „Remington Rand“ (jetzt „Unisys“) verkauft 46 Geräte zu je 1Million USD
- 1953: IBM 650 Magnetic Drum Data-Processing Machine
  - Erster Computer, der in **Massenfertigung** hergestellt wird (~2000 Einheiten)
  - Adressen und Daten wurden dezimal dargestellt
  - Rotierenden Magnettrommel als Speicher
  - Sehr populär an Universitäten

https://commons.wikimedia.org/wiki/Category:UNIVAC_I?uselang=de#/media/File:Univac_I_Census_dedication.jpg

https://www.historyofinformation.com/image.php?id=4426

Anfänge: Informatik & Rechenmaschinen (5)

- 1957: Binär dezimaler Volltransistor Rechenautomat („**Mailüfterl**“)
  - Von Heinz Zemanek an der TU Wien gebaut
  - Erste vollständig mit Transistoren arbeitende Rechner auf dem europäischem Festland
  - **Taktfrequenz** von 132 kHz
- 1960: **PDP-1 (Programmed Data Processor)**
  - Erster Computer von DEC (Digital Equipment Corporation)
  - Speicher: 4096 Wörter zu 18-Bit
  - Eine der ersten Maschinen, für die ein Texteditor und **Spiele („Space Travel“)** entwickelt wurde

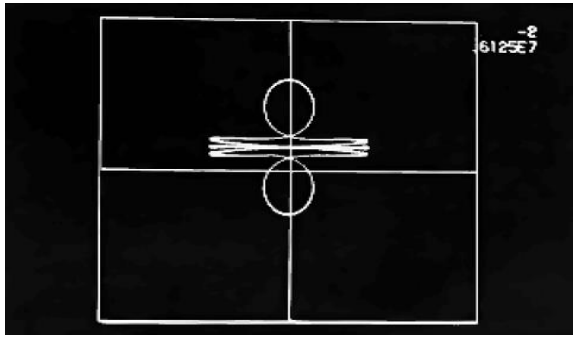
https://futurezone.at/science/mailuefterl-erbauer-wahr-ist-was-funktioniert/29.143.594

http://scihi.org/dennis-ritchie-unix/

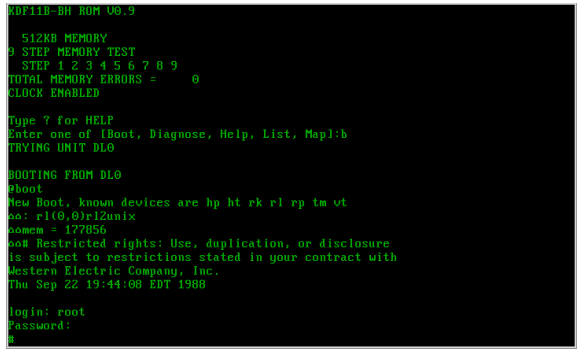
Anfänge: Informatik & Rechenmaschinen (6)

1970: **Unics (Uniplexed Information and Computing Service)**

- Von **Ken Thompson und Dennis Ritchie** in den Bell Labs auf einer DEC PDP-11 implementiert
- Eigenschaften: hierarchisches Dateisystem, Prozesse, Device Files, Command Line Interpreter
- Später zu **UNIX** umbenannt



https://en.wikipedia.org/wiki/Space_Travel_(video_game)



Anfänge: Informatik & Rechenmaschinen (7)

- 1976/77: **Steve Jobs und Steven Wozniak**
  - gründen Apple Computer und bauen den
  - ersten **Heim-Computer**. Von Steve Wozniak entwickelt und Steve Jobs vermarktet
  - MOS 6502 8-Bit Prozessor
  - Apple II: erweiterbar und offen, Preis ca. 1300 USD
- 1981: IBM führt den PC ein
  - **Betriebssystem DOS von Microsoft**
  - Erweiterbares und offenes System
  - Intel 8088 16-Bit Prozessor
  - Einstandspreis bei ca. 3000 USD

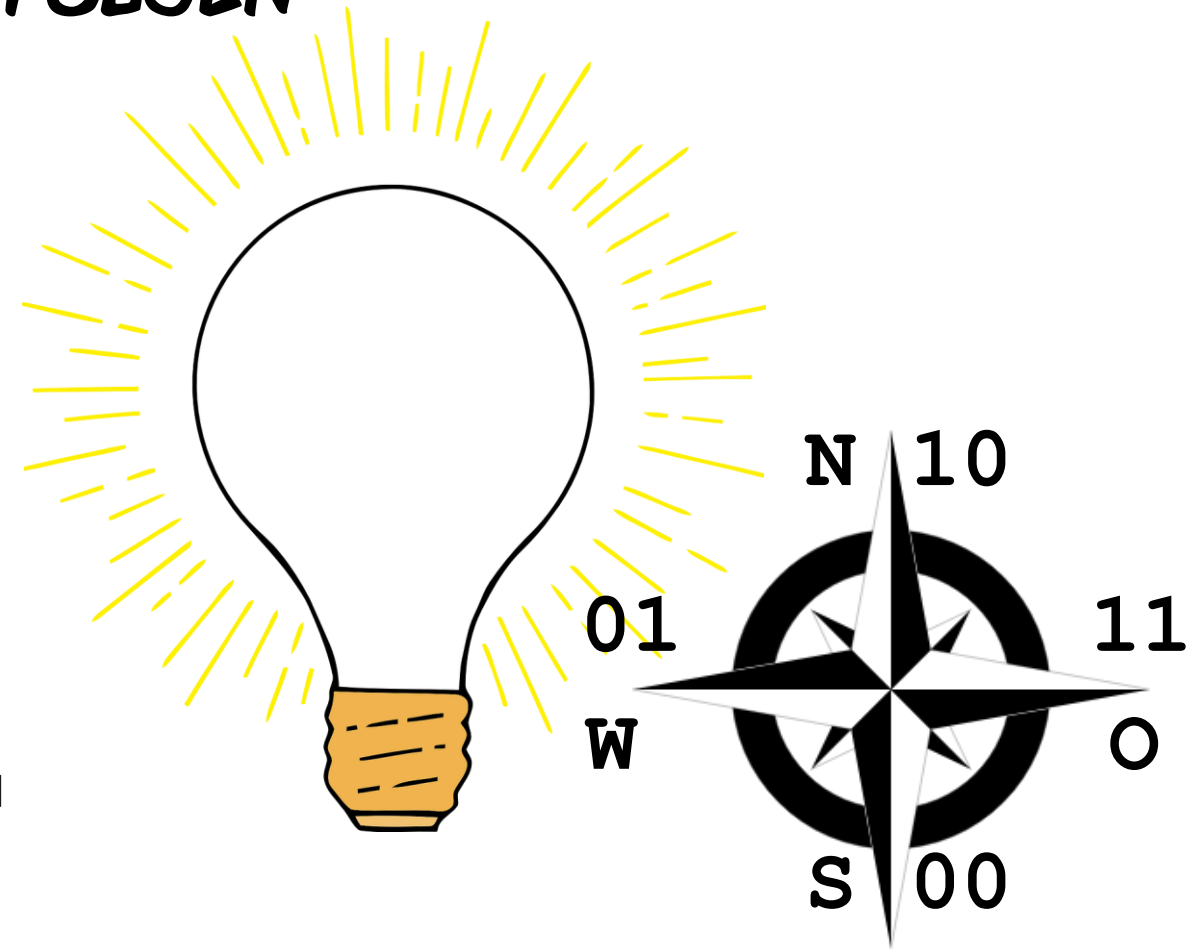
Bildquellen: wikipedia.org

Bits (binary digits) & Bitfolgen

- Ein Bit ist wie ein kleiner **elektronischer Lichtschalter**.
  Es kann entweder **ein oder aus** sein, also **entweder an oder ausgeschaltet**.
  Wie eine **Glühbirne**: Wenn sie leuchtet, ist der Schalter an, und wenn sie nicht leuchtet, ist der Schalter aus. In der Computerwelt repräsentiert ein Bit genau diese zwei Zustände: an oder aus, **1 oder 0**.

- Mehrere Bits beinhalten **komplexere Informationen**.
  Zum Beispiel: Mit 2 Bits kann man 4 verschiedene Kombinationen von 0en und 1en machen- und zum Beispiel für einen Kompass verwenden

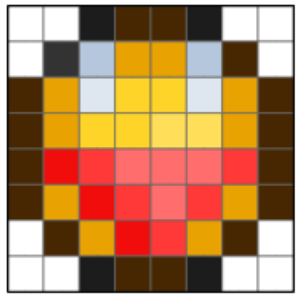
- Bits sind die **kleinsten Bausteine der digitalen Welt** und werden verwendet, um alle Arten **von Informationen in Computern zu speichern und zu verarbeiten**, von Texten und Bildern bis hin zu Videospielen und Internetseiten. Sie sind die Grundlage dafür, wie Computer arbeiten und Informationen verarbeiten können.



Information und Daten

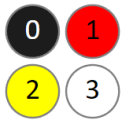
Decimal	Binary	Hexadecimal
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

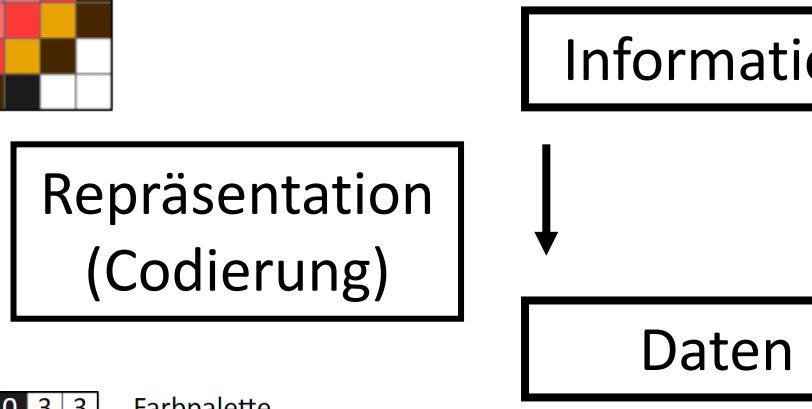
Vorschrift = Codierung



3	3	0	0	0	0	3	3
3	0	3	2	2	3	0	3
0	2	3	2	2	3	2	0
0	2	2	2	2	2	2	0
0	1	1	1	1	1	1	0
0	2	1	1	1	1	2	0
3	0	2	1	1	2	0	3
3	3	0	0	0	0	3	3

Farbpalette





https://weiterbildung-informatik.wollw.de/chapter2/part1/sec2/

Bitfolgen für Zeichen, Zahlen und Buchstaben

Was wird hier geschrieben?

01001000 01100001 01101100
01101100 01101111 00100001

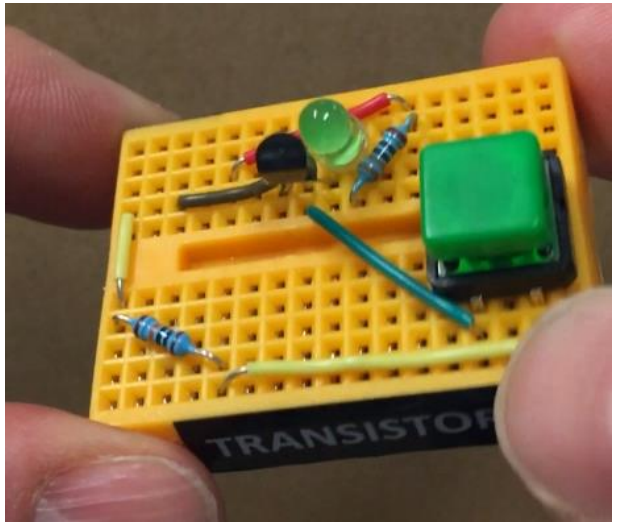
Decimal	Binary	Octal	Hex	ASCII	Decimal	Binary	Octal	Hex	ASCII	Decimal	Binary	Octal	Hex	ASCII	Decimal	Binary	Octal	Hex	ASCII
0	00000000	000	00	NUL	32	00100000	040	20	SP	64	01000000	100	40	@	96	01100000	140	60	`
1	00000001	001	01	SOH	33	00100001	041	21	!	65	01000001	101	41	A	97	01100001	141	61	a
2	00000010	002	02	STX	34	00100010	042	22	"	66	01000010	102	42	B	98	01100010	142	62	b
3	00000011	003	03	ETX	35	00100011	043	23	#	67	01000011	103	43	C	99	01100011	143	63	c
4	00000100	004	04	EOT	36	00100100	044	24	$	68	01000100	104	44	D	100	01100100	144	64	d
5	00000101	005	05	ENQ	37	00100101	045	25	%	69	01000101	105	45	E	101	01100101	145	65	e
6	00000110	006	06	ACK	38	00100110	046	26	&	70	01000110	106	46	F	102	01100110	146	66	f
7	00000111	007	07	BEL	39	00100111	047	27	'	71	01000111	107	47	G	103	01100111	147	67	g
8	00001000	010	08	BS	40	00101000	050	28	(	72	01001000	110	48	H	104	01101000	150	68	h
9	00001001	011	09	HT	41	00101001	051	29	)	73	01001001	111	49	I	105	01101001	151	69	i
10	00001010	012	0A	LF	42	00101010	052	2A	*	74	01001010	112	4A	J	106	01101010	152	6A	j
11	00001011	013	0B	VT	43	00101011	053	2B	+	75	01001011	113	4B	K	107	01101011	153	6B	k
12	00001100	014	0C	FF	44	00101100	054	2C	,	76	01001100	114	4C	L	108	01101100	154	6C	l
13	00001101	015	0D	CR	45	00101101	055	2D	-	77	01001101	115	4D	M	109	01101101	155	6D	m
14	00001110	016	0E	SO	46	00101110	056	2E	.	78	01001110	116	4E	N	110	01101110	156	6E	n
15	00001111	017	0F	SI	47	00101111	057	2F	/	79	01001111	117	4F	O	111	01101111	157	6F	o
16	00010000	020	10	DLE	48	00110000	060	30	0	80	01010000	120	50	P	112	01110000	160	70	p
17	00010001	021	11	DC1	49	00110001	061	31	1	81	01010001	121	51	Q	113	01110001	161	71	q
18	00010010	022	12	DC2	50	00110010	062	32	2	82	01010010	122	52	R	114	01110010	162	72	r
19	00010011	023	13	DC3	51	00110011	063	33	3	83	01010011	123	53	S	115	01110011	163	73	s
20	00010100	024	14	DC4	52	00110100	064	34	4	84	01010100	124	54	T	116	01110100	164	74	t
21	00010101	025	15	NAK	53	00110101	065	35	5	85	01010101	125	55	U	117	01110101	165	75	u
22	00010110	026	16	SYN	54	00110110	066	36	6	86	01010110	126	56	V	118	01110110	166	76	v
23	00010111	027	17	ETB	55	00110111	067	37	7	87	01010111	127	57	W	119	01110111	167	77	w
24	00011000	030	18	CAN	56	00111000	070	38	8	88	01011000	130	58	X	120	01111000	170	78	x
25	00011001	031	19	EM	57	00111001	071	39	9	89	01011001	131	59	Y	121	01111001	171	79	y
26	00011010	032	1A	SUB	58	00111010	072	3A	:	90	01011010	132	5A	Z	122	01111010	172	7A	z
27	00011011	033	1B	ESC	59	00111011	073	3B	;	91	01011011	133	5B	[	123	01111011	173	7B	{
28	00011100	034	1C	FS	60	00111100	074	3C	<	92	01011100	134	5C	\	124	01111100	174	7C	|
29	00011101	035	1D	GS	61	00111101	075	3D	=	93	01011101	135	5D	]	125	01111101	175	7D	}
30	00011110	036	1E	RS	62	00111110	076	3E	>	94	01011110	136	5E	^	126	01111110	176	7E	~
31	00011111	037	1F	US	63	00111111	077	3F	?	95	01011111	137	5F	_	127	01111111	177	7F	DEL

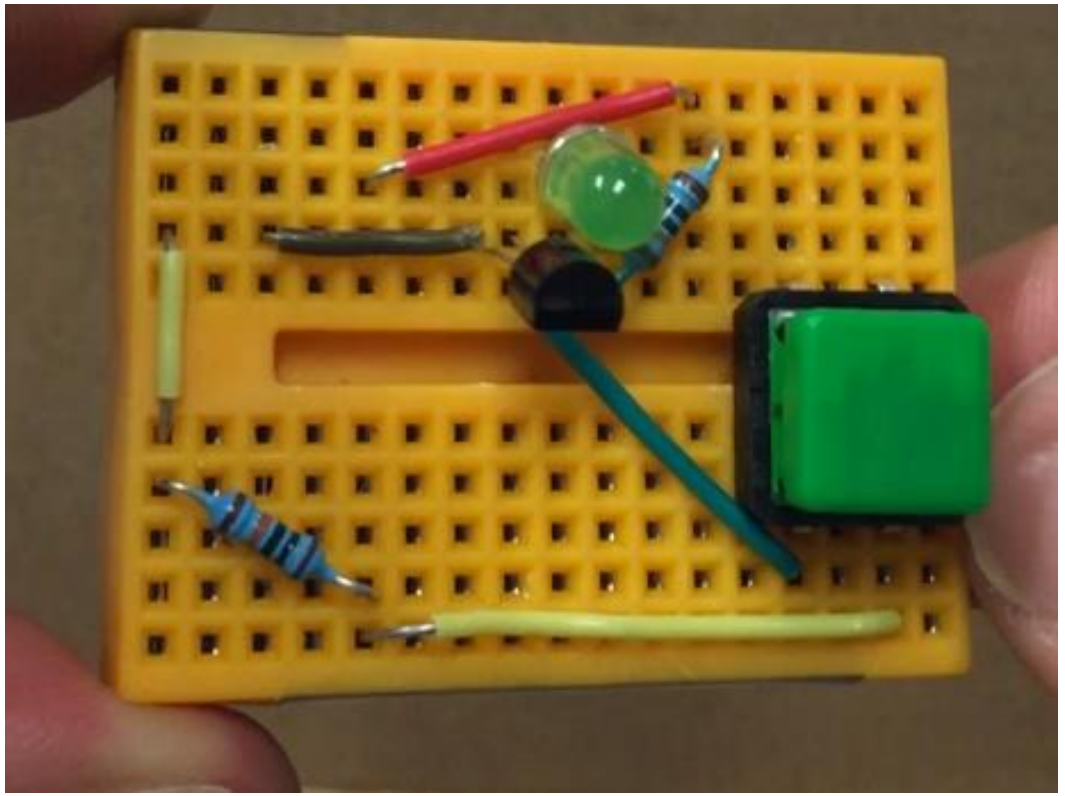
ACII Codes - American Standard Code for Information Interchange

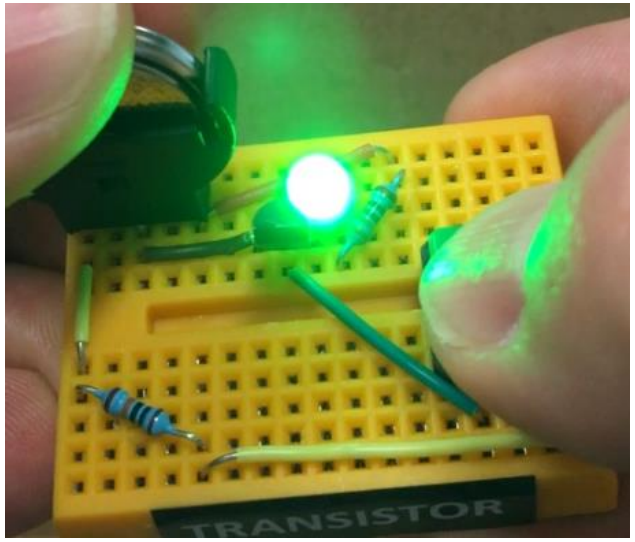
This work is licensed under the Creative Commons Attribution-ShareAlike License. To view a copy of this license, visit http://creativecommons.org/licenses/by-sa/3.0/

ASCII Conversion Chart.doc Copyright © 2008, 2012 Donald Weiman 22 March 2012

Einfache Schaltungen - Aus/Ein mit Transistoren



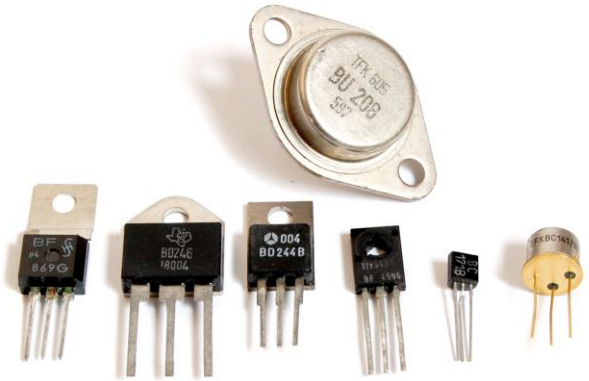


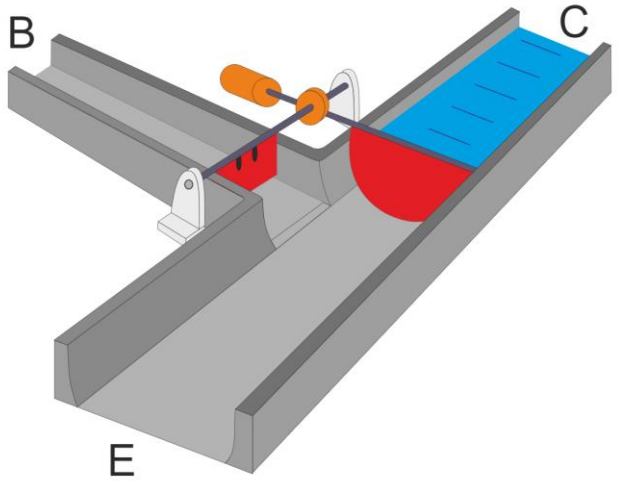


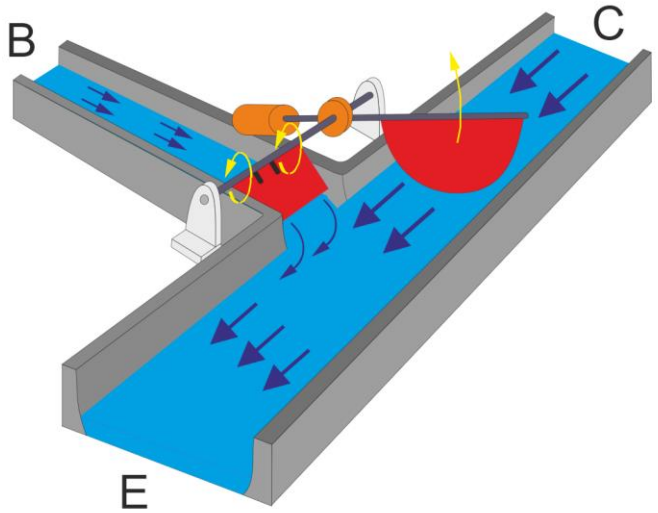
Transistoren

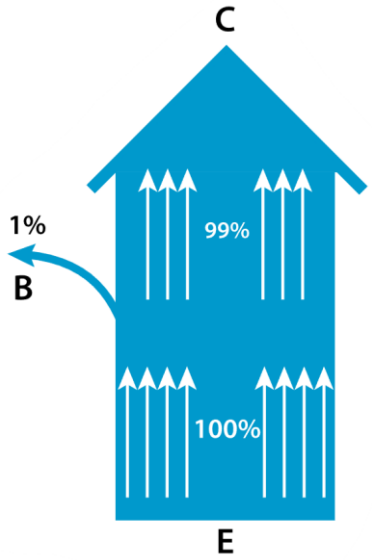
Elementarer Bestandteil im Computer sind Schalter, die mechanisch, elektronisch mit Röhren oder Transitoren umgesetzt wurden. Heute finden wir Milliarden Transistoren ("**Trans**fer Res**istor**") auf kleinster Fläche.

Funktionsweise anhand eines mechanischen Modells:



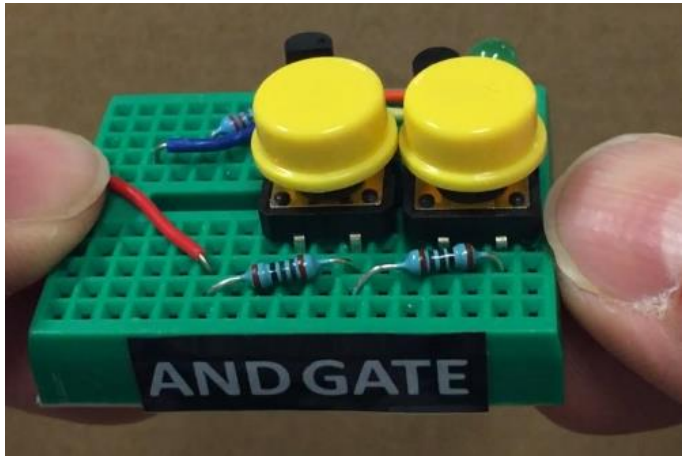


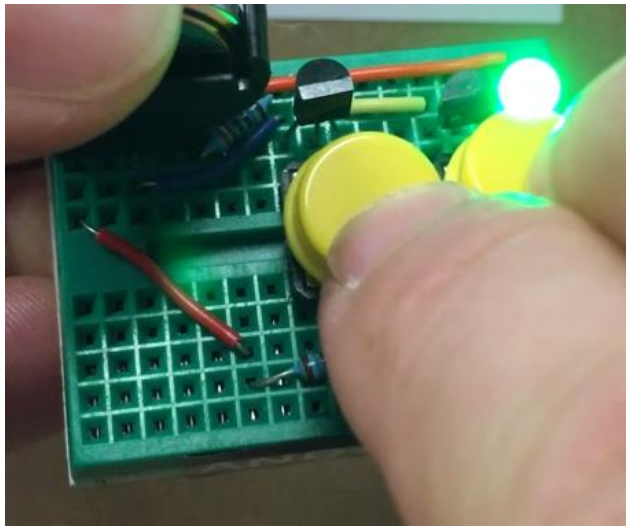




https://www.leifiphysik.de/elektronik/transistor/grundwissen/der-transistor-effekt

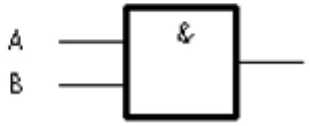
AND Gatter

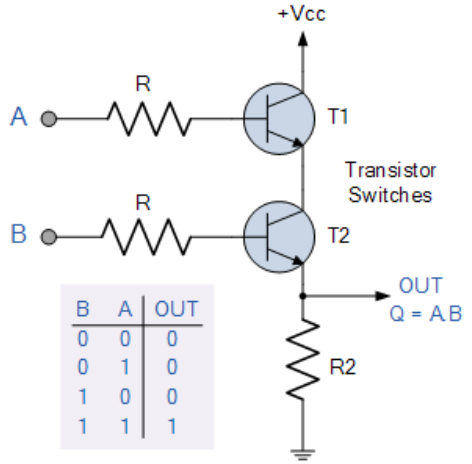




US-Symbol

IEC-Symbol

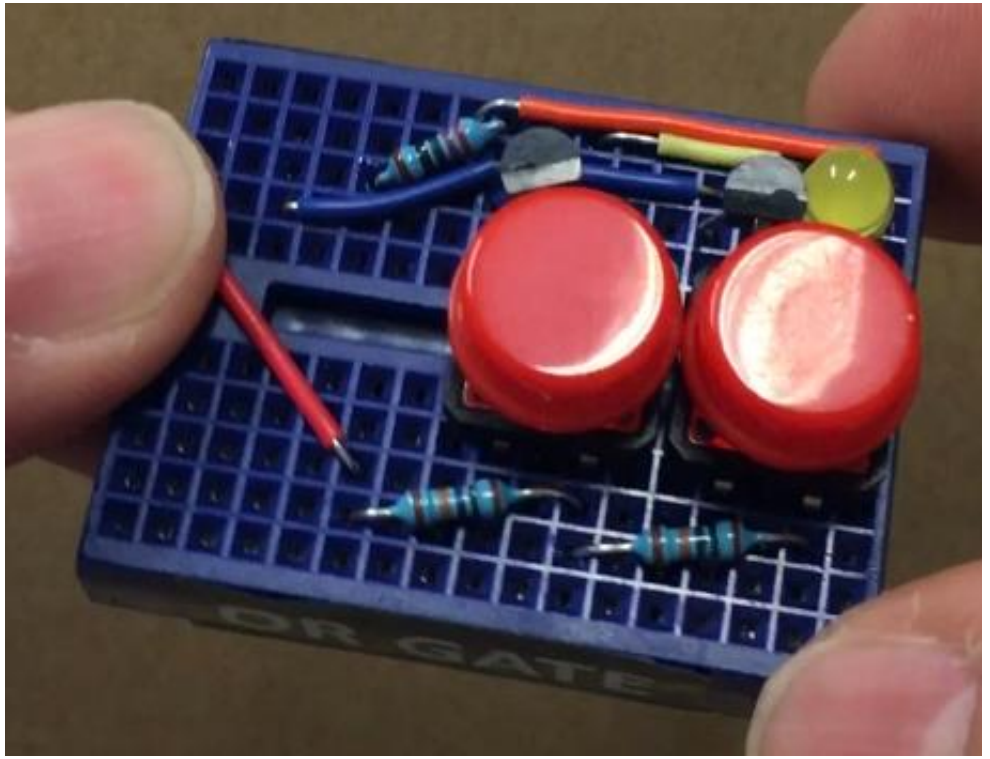


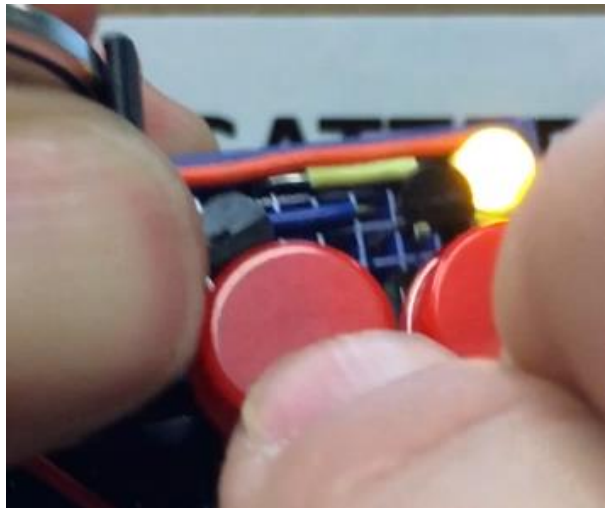


IEC = International Electrotechnical Commission

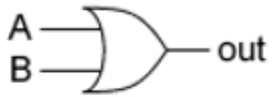
https://www.electronics-tutorials.ws/de/logische/und-gatter.html

OR Gatter

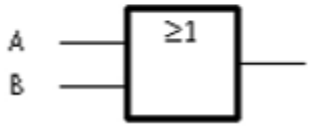




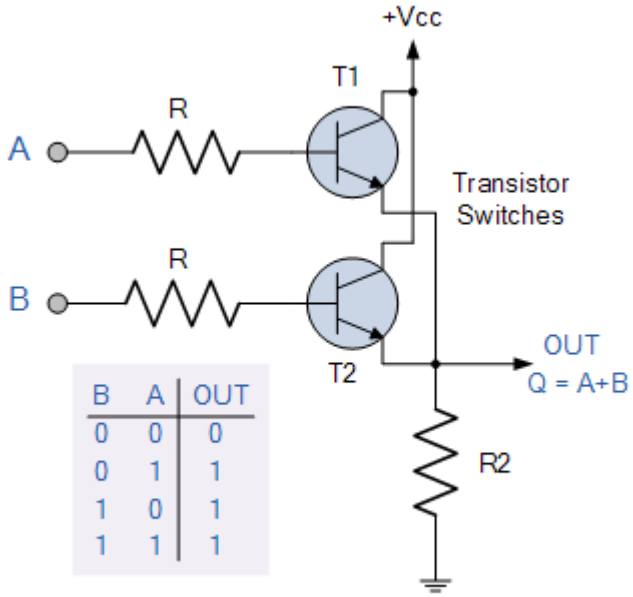
US-Symbol



IEC-Symbol



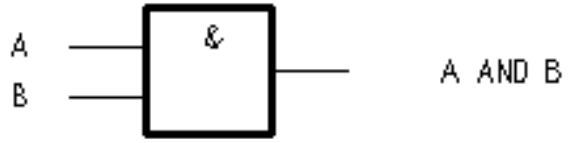
IEC = International Electrotechnical Commission

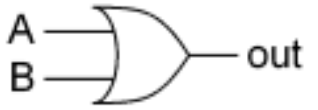


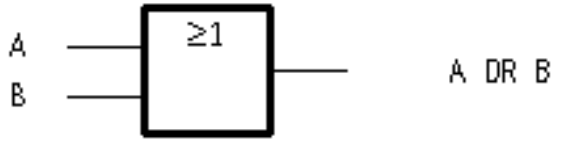
B	A	OUT
0	0	0
0	1	1
1	0	1
1	1	1

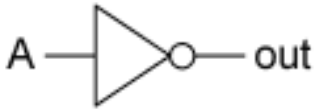
Weitere Gatter

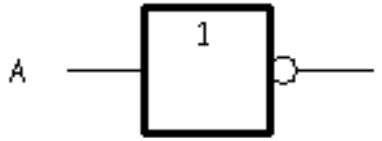
AND-Gatter: US-Symbol 

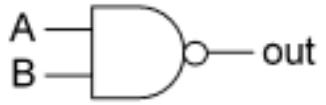
IEC-Symbol 

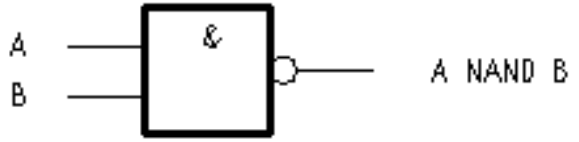
OR-Gatter: US-Symbol 

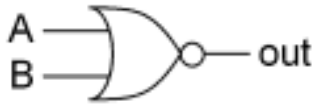
IEC-Symbol 

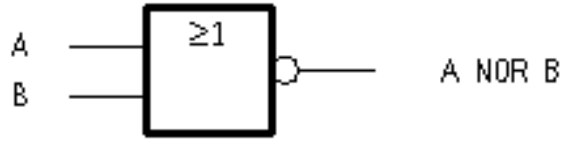
NOT-Gatter: US-Symbol 

IEC-Symbol 

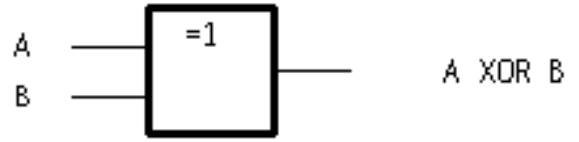
NAND-Gatter: US-Symbol 

IEC-Symbol 

NOR-Gatter: US-Symbol 

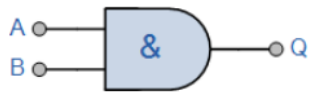
IEC-Symbol 

XOR-Gatter: US-Symbol 

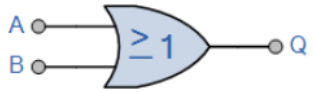
IEC-Symbol 

Gatter - Tabellen

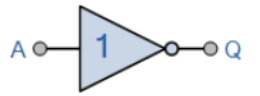
AND

Symbol	Wahrheitstabelle		
 2-Eingang logik UND-Gatter	B	A	Q
	0	0	0
	0	1	0
	1	0	0
	1	1	1
Boolescher Ausdruck **Q = A.B**	Lesen als A **UND** B gibt Q		

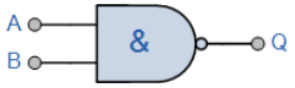
OR

Symbol	Wahrheitstabelle		
	B	A	Q
	0	0	0
	0	1	1
	1	0	1
	1	1	1
Boolescher Ausdruck Q = A + B	Lesen als A **ODER** B gibt Q		

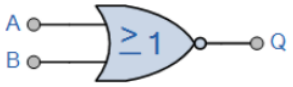
NOT

Symbol	Wahrheitstabelle	
 Wandler oder NICHT-Gatter	A	Q
	0	1
	1	0
Boolescher Ausdruck **Q = NICHT A oder $\overline{A}$**	Als Invers von lesen **A** ergeben Q	

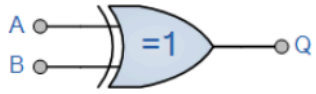
NAND

Symbol	Wahrheitstabelle		
	B	A	Q
	0	0	1
	0	1	1
	1	0	1
	1	1	0
Boolescher Ausdruck Q = $\overline{A \,.\, B}$	Lesen als A **UND** B gibt **NICHT** Q		

NOR

Symbol	Wahrheitstabelle		
	B	A	Q
	0	0	1
	0	1	0
	1	0	0
	1	1	0
Boolescher Ausdruck Q = $\overline{A + B}$	Lesen als A **ODER** B gibt **NICHT** Q		

XOR

Symbol	Wahrheitstabelle		
	B	A	Q
	0	0	0
	0	1	1
	1	0	1
	1	1	0
Boolescher Ausdruck Q = A ⊕ B	Lesen als A **ODER** B aber nicht **BEIDE** gibt Q (ungerade)		

Rechnen mit Bits

Im Dezimalsystem haben wir die Zahlen 0 bis 9 = **10 Zahlen**
Im Binär/Dualsystem nur 0 bis 1 = **2 Zahlen**

Immer wenn wir beim Zählen über unseren Zahlenvorrat Kommen, haben wir einen Übertrag!

Dezimal

```
  3      5      8
+ 4    + 6    +67
---    ---    ---
  7     11     75
```

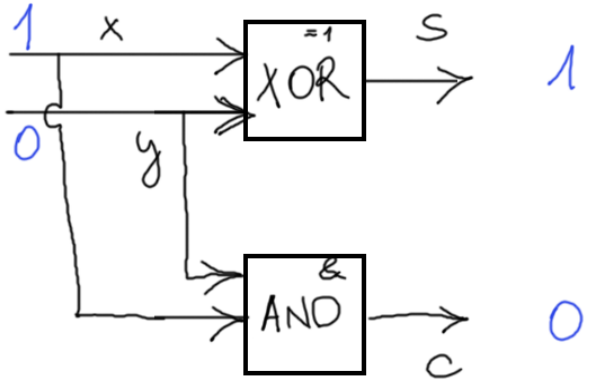
Dual

```
  1      1      1
  0    + 1    + 11
---    ---    ---
+ 1     1      1
        1      1
---    ---    ---
  1     10    100
```

```
  100110
+   1011
  111
--------
  110001
```

HALBADDIERER

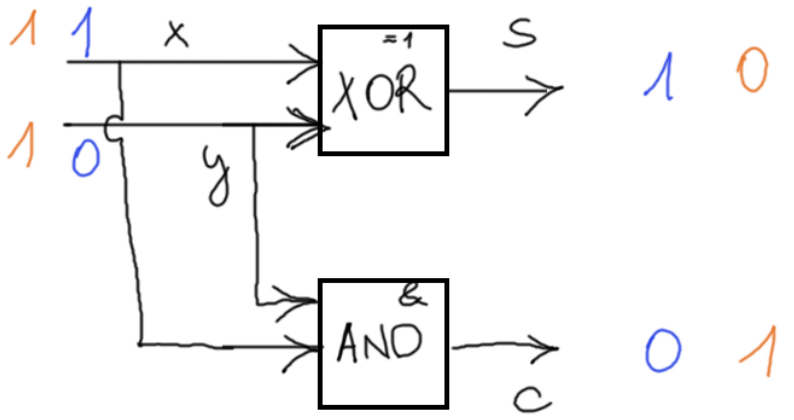
$$\begin{array}{r} 1 \\ 0 \\ \hline +\ 1 \end{array}$$



1.) 1 + 0 → S = 1, C = 0

$$\begin{array}{r} 1 \\ +\ 0 \\ \hline 01 \end{array}$$

$$\begin{array}{r} 1 \\ 1 \\ 1 \\ \hline 10 \end{array}$$



2.) 1 + 1 → S = 0, C = 1 ⇒

$$\begin{array}{r} 1 \\ +\ 1 \\ \hline 10 \end{array}$$

AND:

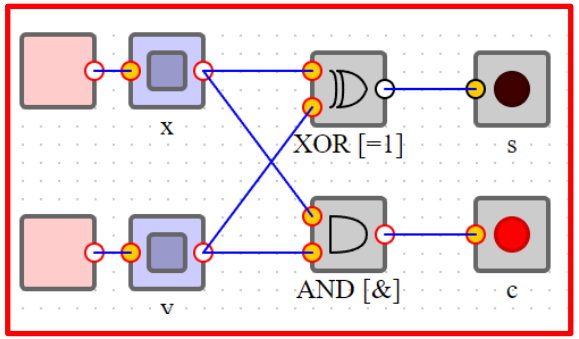
A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

OR:

A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

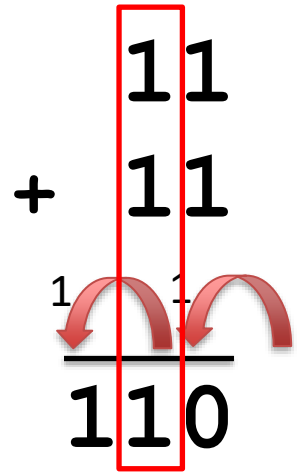
XOR:

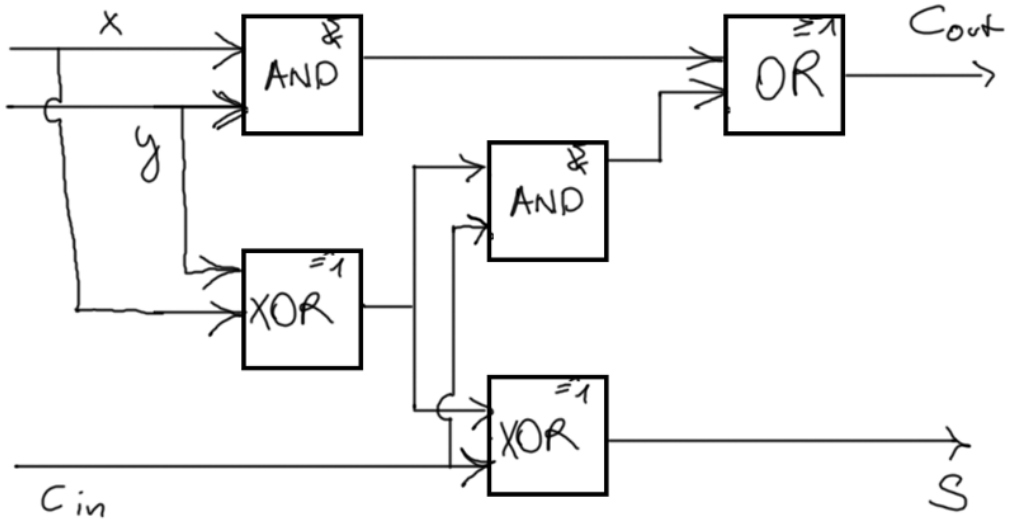
A	B	X
0	0	0
0	1	1
1	0	1
1	1	0

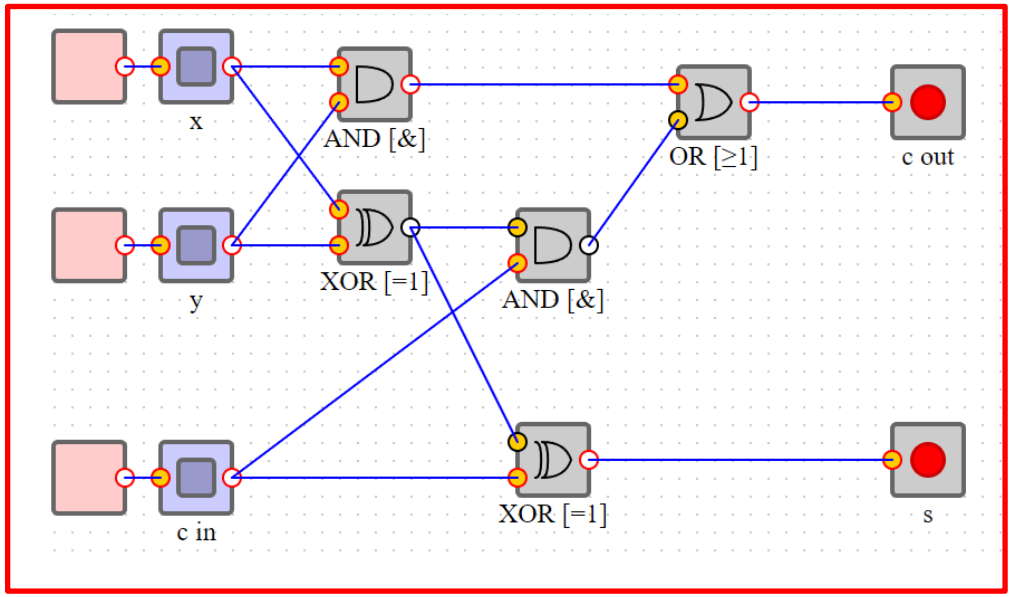


https://elearning.fh-joanneum.at/simcurcuits/half_adder.html

VOLLADDIERER







AND:

A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

OR:

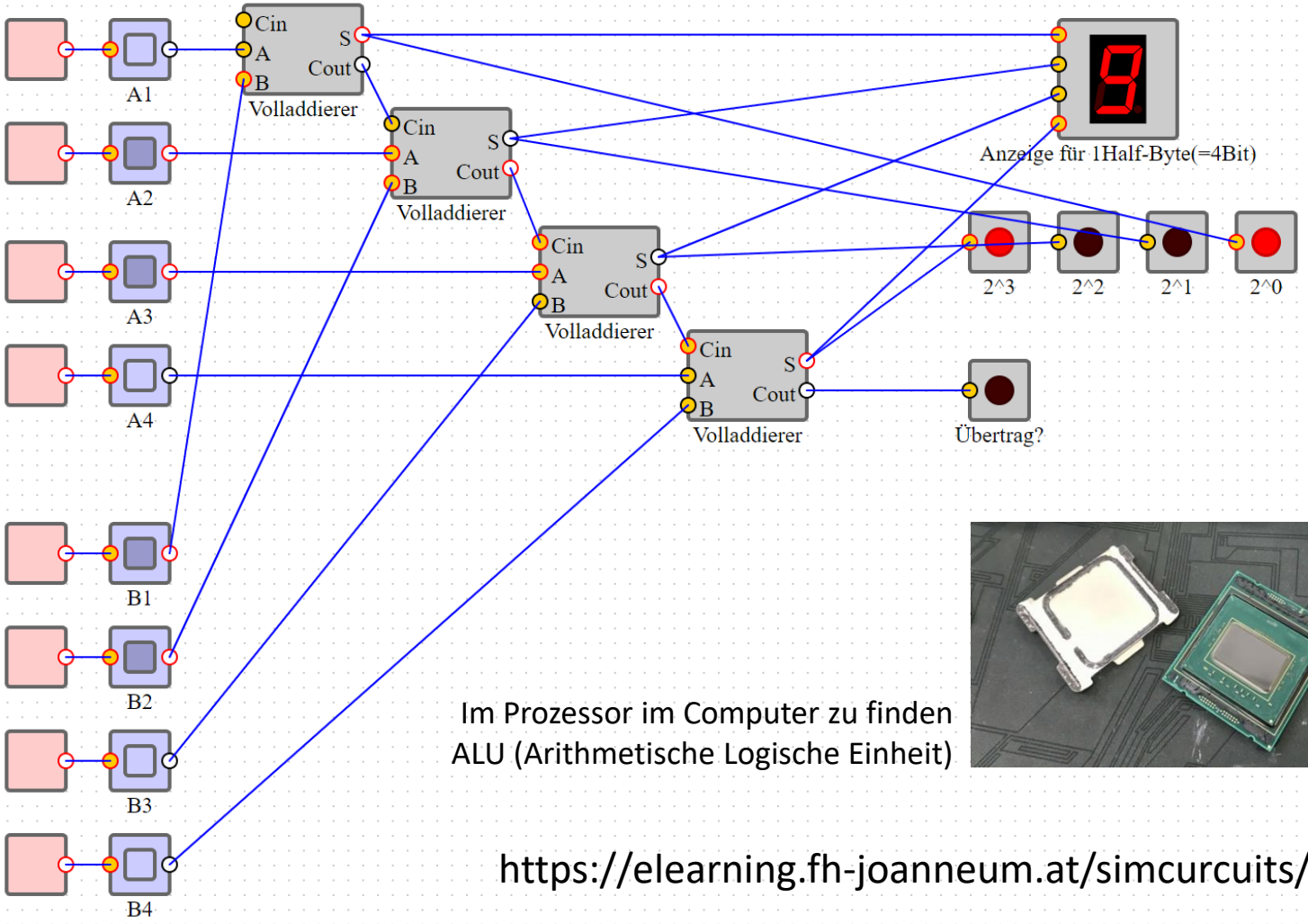
A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

XOR:

A	B	X
0	0	0
0	1	1
1	0	1
1	1	0

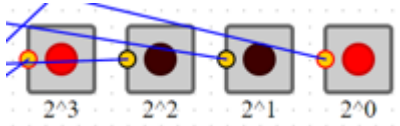
https://elearning.fh-joanneum.at/simcurcuits/full_adder.html

Rechenmaschine aus Volladdierern



Im Prozessor im Computer zu finden
ALU (Arithmetische Logische Einheit)

A4	A3	A2	A1
0	1	1	0
+ B4	B3	B2	B1
0	0	1	1
1	1		
1	0	0	1

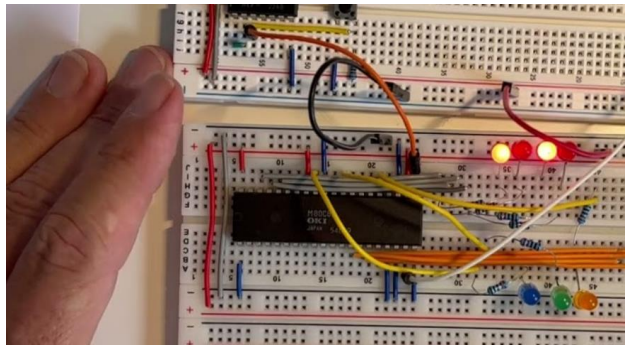


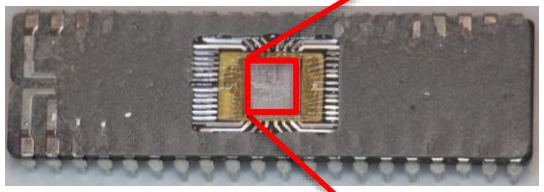
https://elearning.fh-joanneum.at/simcurcuits/4bitmachine.html

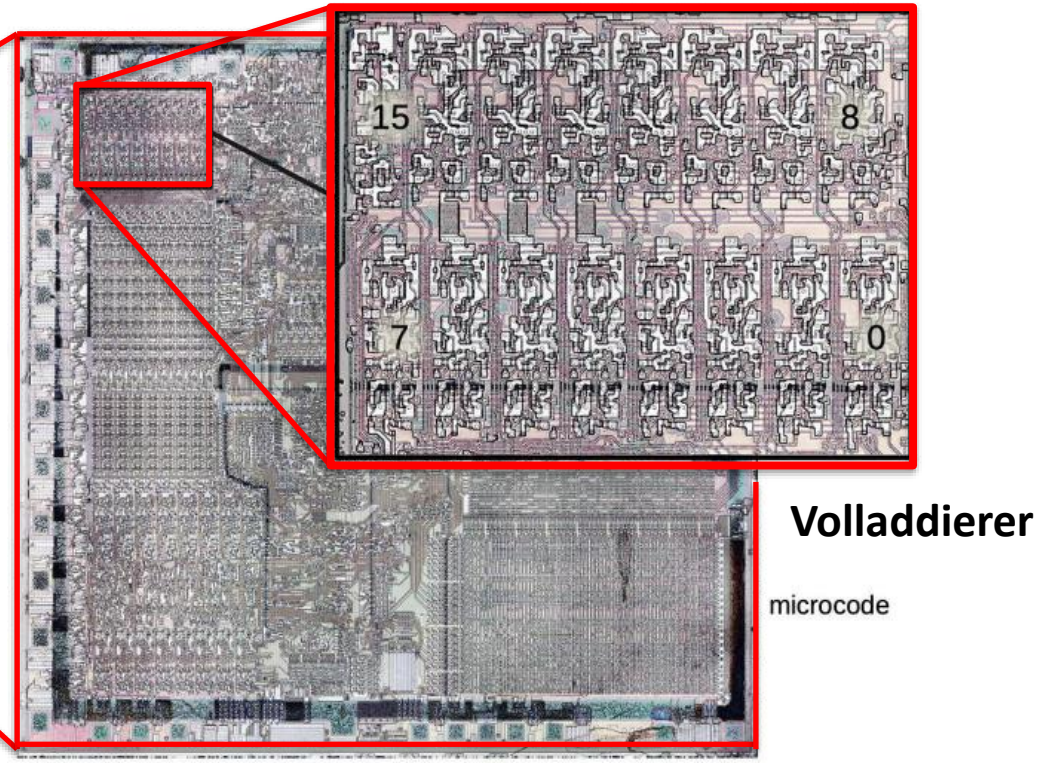
Der Prozessor im Computer

Er steuert und rechnet – das Herzstück eines jeden Computers!

Beispiel: Intel 8086 (1978)

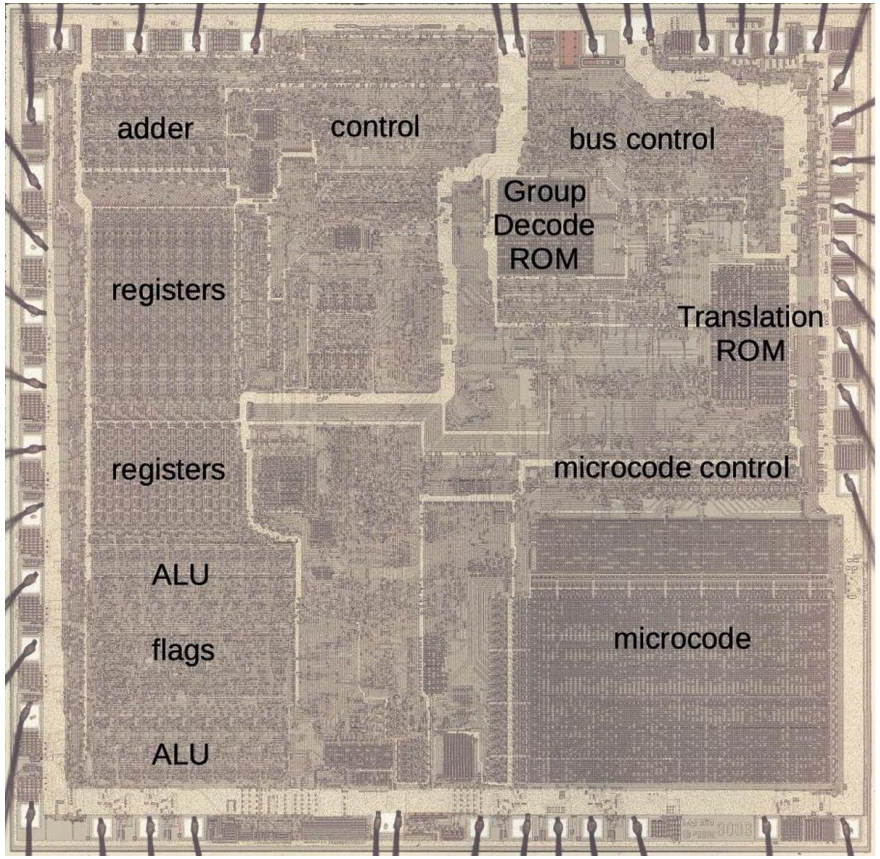


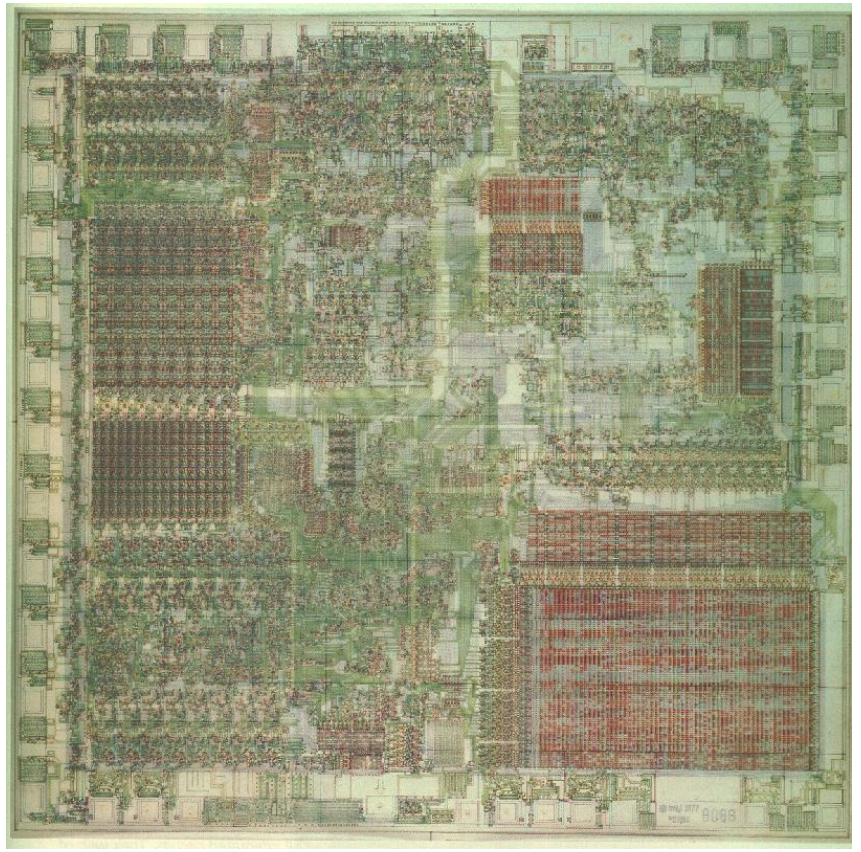


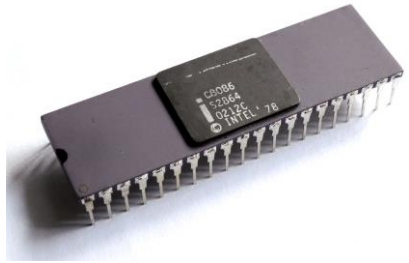


http://www.visual6502.org/sim/varm/armgl.html
https://de.wikipedia.org/wiki/Intel_8086
https://www.righto.com/2020/06/die-shrink-how-intel-scaled-down-8086.html
http://www.righto.com/2015/12/reverse-engineering-arm1-ancestor-of.html

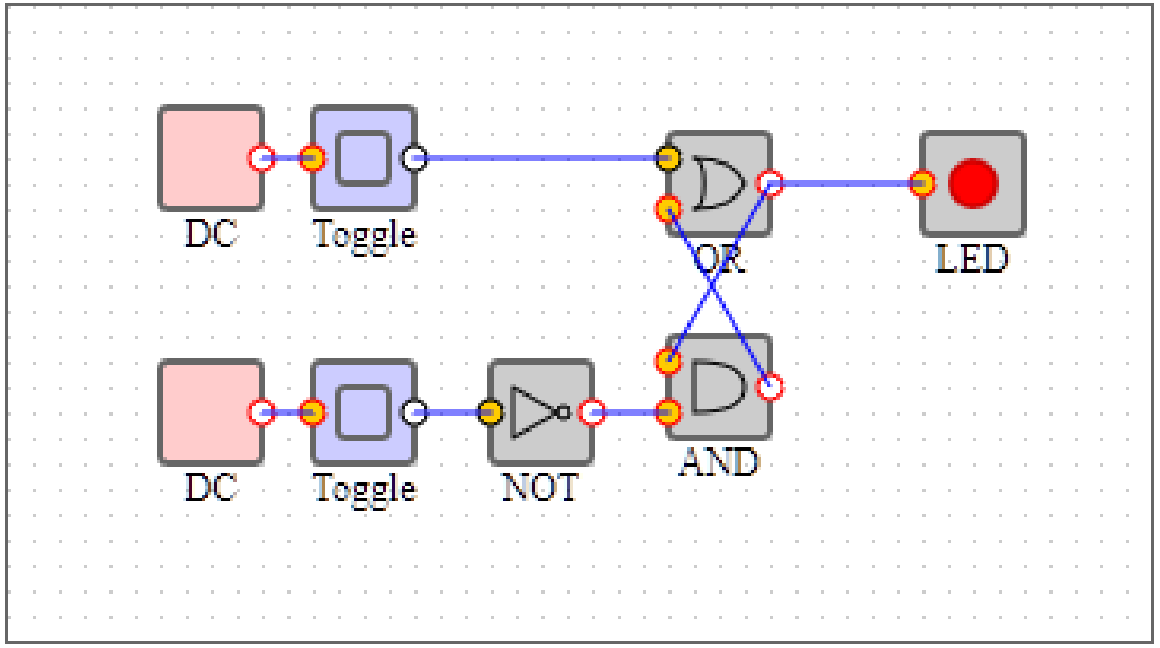
Prozessorbilder (Intel 8086 - 1978)





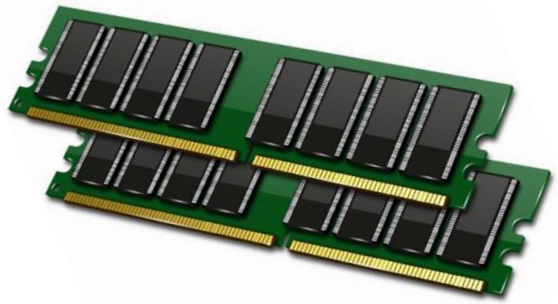


Speichern von Bits - Flip Flop

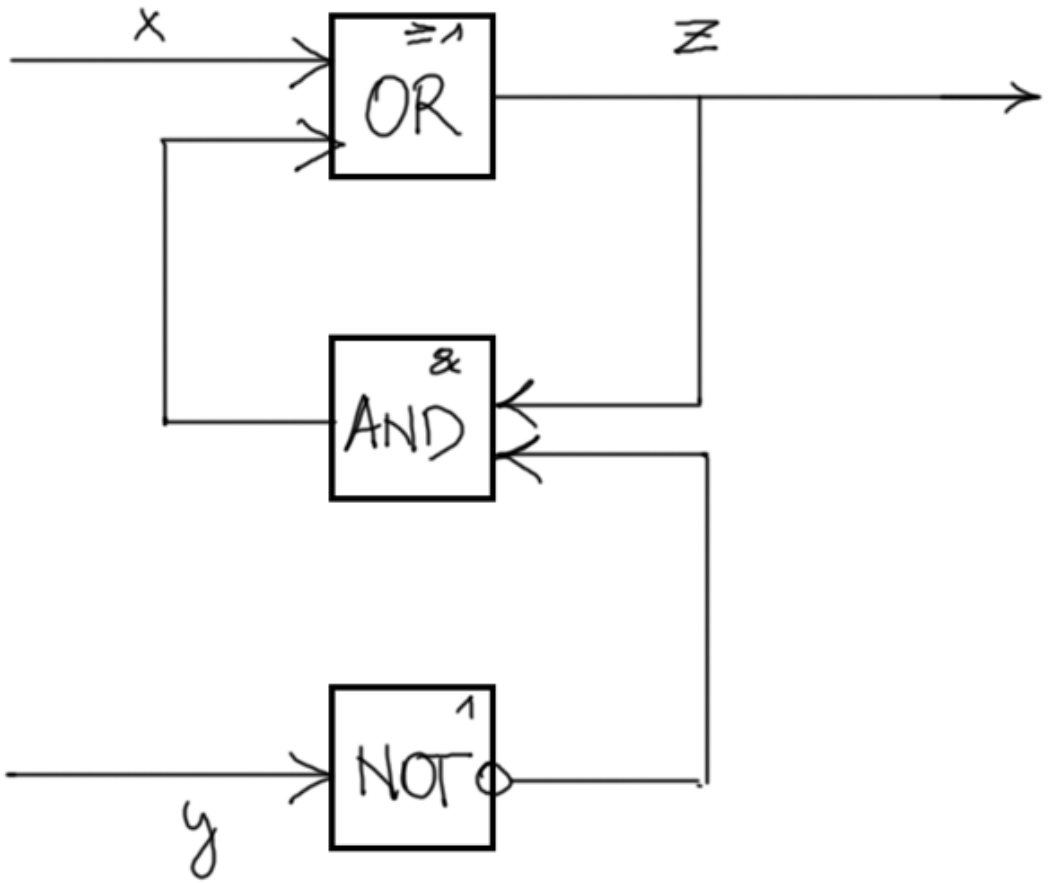


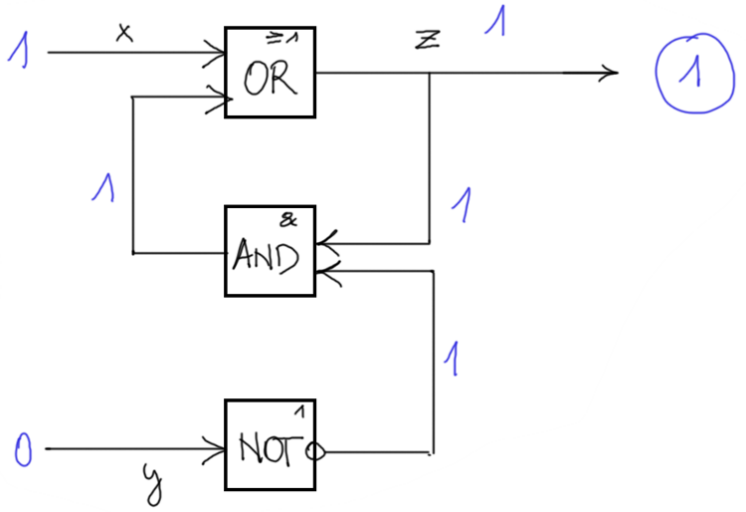
Ein Flip-Flop wirkt wie ein Schalter:
Grundstellung ist mit 0 an x und y
Temporäres Anlegen von 1 an x setzt Output auf 1
Temporäres Anlegen von 1 an y setzt Output auf 0
Ist damit geeignet zur Speicherung einzelner Bits

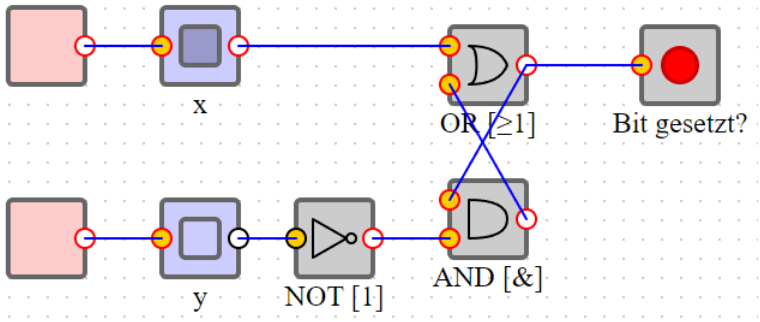
Arbeitsspeicher am Computer
Milliarden von Bits



Flip Flop - Funktion 1

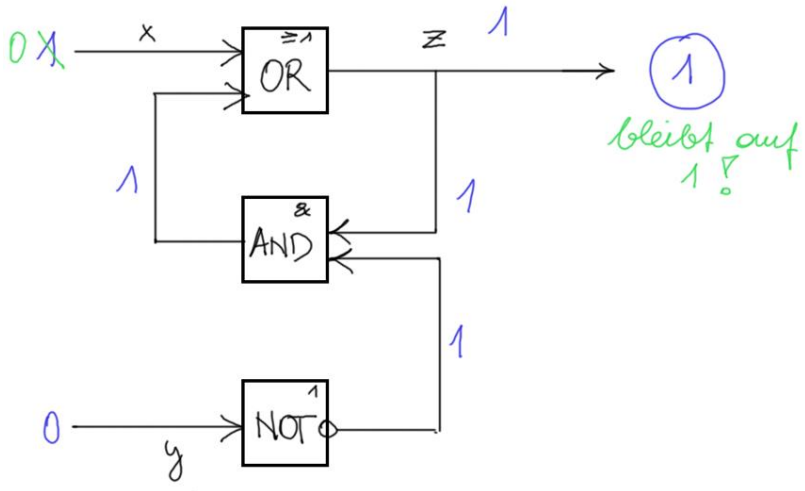


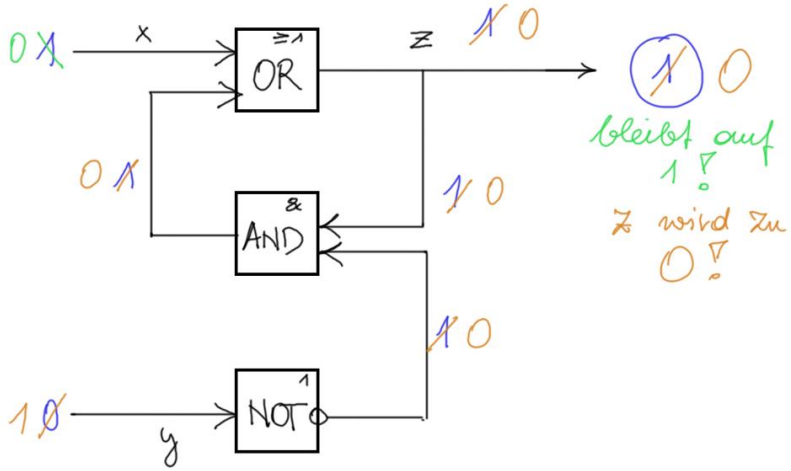


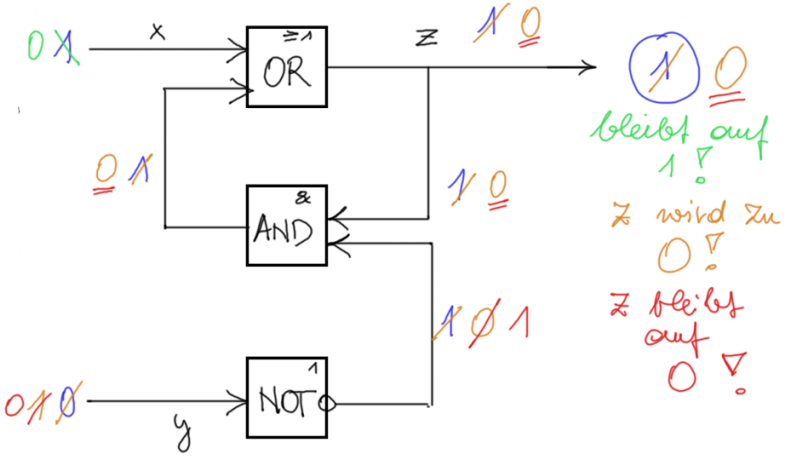


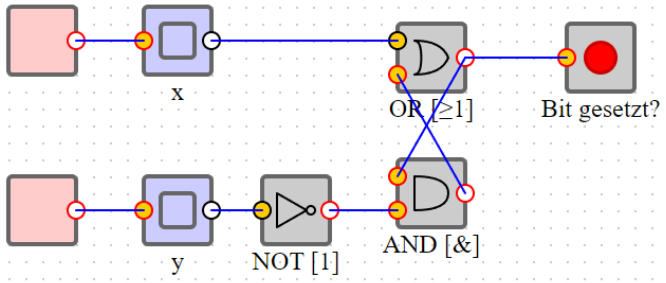
Flip Flop - Funktion 2

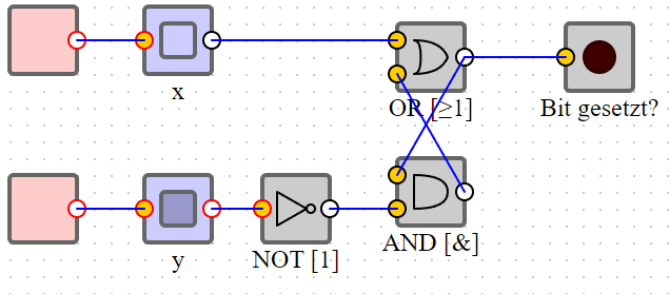
1. x = 1 y = 0
2. x = 0
3. y = 1
4. y = 0

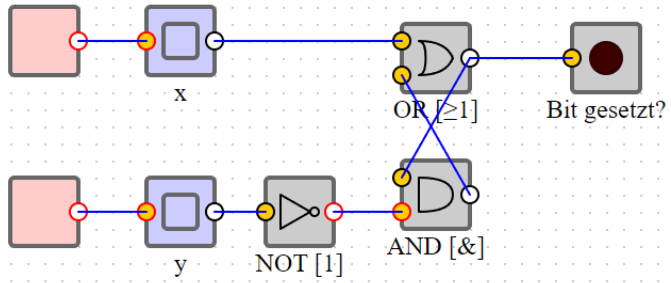




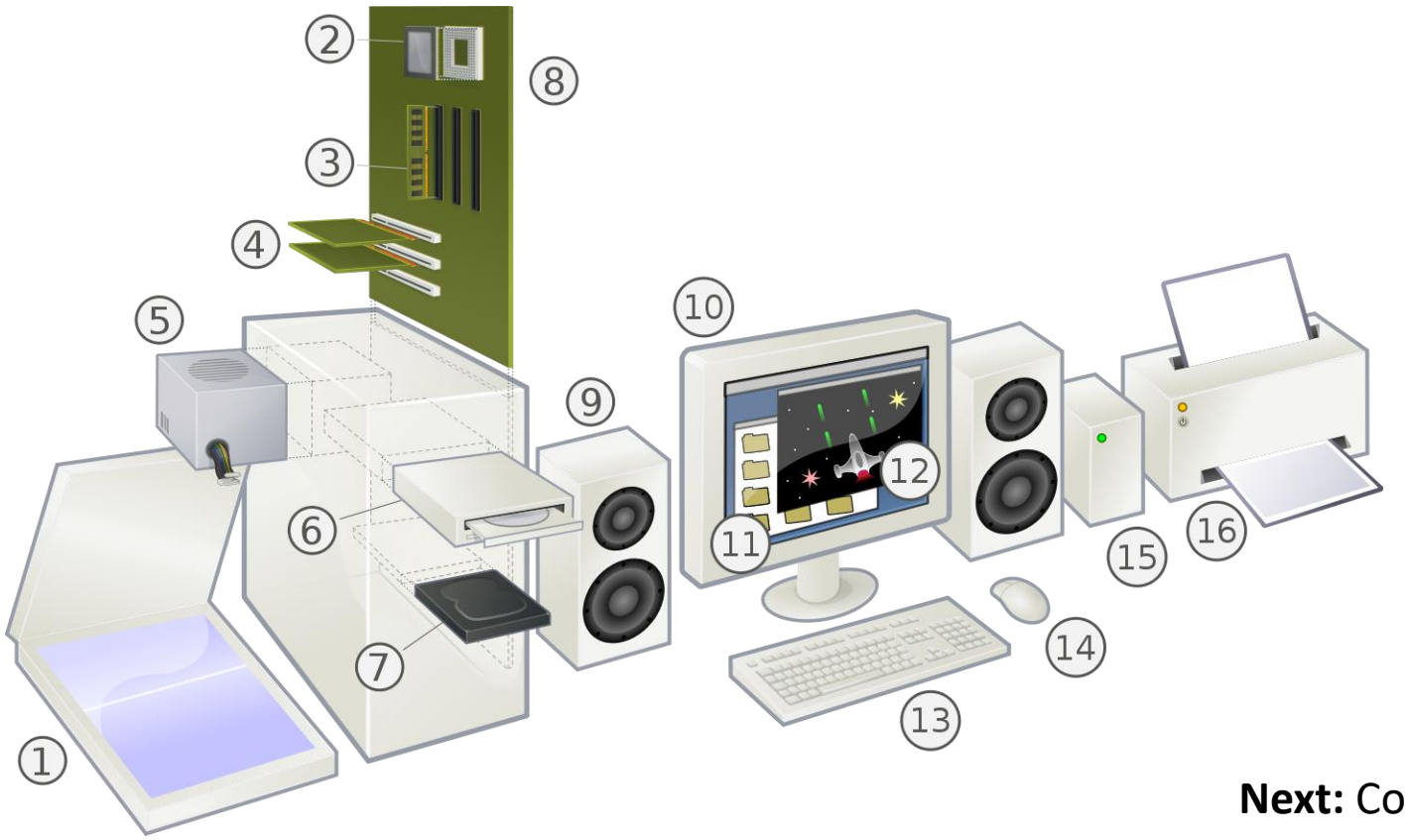








Rechneraufbau



Die wichtigen Teile:

② **Zentraleinheit**
(CPU, Central Processing Unit)

③ **Arbeitsspeicher**
(RAM, Random Access Memory)

④ Erweiterungskarten
z.B. Grafikkarten

⑧ Hauptplatine
(Mainboard oder Motherboard)

⑤ Netzteil

Next: Computer und Server auseinandernehmen

FH | JOANNEUM
University of Applied Sciences

Cloud Computing Lab

FFG
Forschung wirkt.