A

16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
65.536	32.768	16.384	8.192	4.096	2.048	1.024	512	256	128	64	32	16	8	4	2	1

A	B	C	D	E	F
10	11	12	13	14	15

- nur Buchstaben oder _ am Anfang
- insgesamt nur Buchstaben, _ oder Ziffern
- Groß- und Kleinschreibung sind unterschiedliche Variablen
- generell kein $
- aufpassen bei 0 (Okt., max. 7) oder 0x (Hex., max. F)
- Variable muss Wert haben bevor man mit ihr rechnen kann
- Variable kann nicht zweimal deklariert werden
- .3 bedeutet 0.3 und ist i.O.
- ´a´ hinten in printf ist i.O.
- *unsigned* und trotzdem -1 ist i.O.
- a = a++ ist falsch
- Operator gleich in der Deklaration ist i.O. (zB int a = 5 > 7)

Mikroprozessor
- Befehlscodiereinheit
- Arbeitsregister
- Befehlszählerregister
- Statusregister
- ALU/Addierwerk

Zentraleinheit
- Bussystem
- Ein-/Ausgabe
- Arbeitsspeicher
- Prozessor

Gray

1 Bit	2 Bit	3 Bit	4 Bit
0	0 0	0 0 0	0 0 0 0
1	0 1	0 0 1	0 0 0 1
	1 1	0 1 1	0 0 1 1
	1 0	0 1 0	0 0 1 0
		1 1 0	0 1 1 0
		1 1 1	0 1 1 1
		1 0 1	0 1 0 1
		1 0 0	0 1 0 0
			1 1 0 0
			1 1 0 1
			1 1 1 1
			1 1 1 0
			1 0 1 0
			1 0 1 1
			1 0 0 1
			1 0 0 0

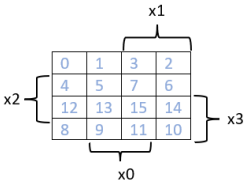
Prozessor-Register
L1-Cache
L2-Cache
Arbeitsspeicher
Festplatte

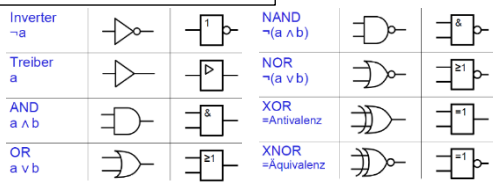
B

Analog → real
Digital → abstrakt
A nach D: Komparator od. Abtaster u. Quantisierer
D nach A: D/A-Wandler u. Filter

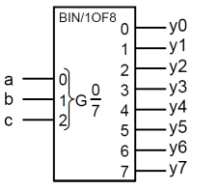
Boolsche Funktionen: 2^{2^x} (x= Anzahl Var.)
Min-Term: kleinster Wert wird zurückgegeben (AND)
Max-Term: größter Wert wird zurückgegeben (OR)

t_r und t_f: 10% bis 90%
t_{PHL} und t_{PLH}: 50% bis 50%
t_{Setup}: Zeit vor Taktflanke mit stabilem Zustand
t_{Hold}: Zeit nach Taktflanke mit stabilem Zustand
IMMER: $t_P > t_{Hold}$! (D-FF, aber JK und T bestehen aus D)
Fan-Out: $\min\left\{\frac{I_{OLmax}}{I_{IL}}\ (abgerundet), \frac{I_{OHmax}}{I_{IH}}\ (abgerundet)\right\}$
Noise-Margin: $S_H = V_{OHmin} - V_{IHmin}\ \ S_L = V_{ILmax} - V_{OLmax}$



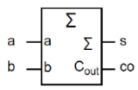


Decoder

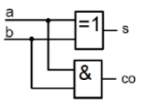


c	b	a	y0	y1	y2	y3	y4	y5	y6	y7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

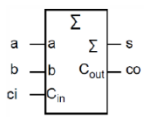
Halbaddierer



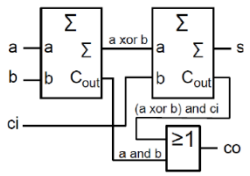
	a	b	co	s
0	0	0	0	0
1	0	1	0	1
2	1	0	0	1
3	1	1	1	0



VA

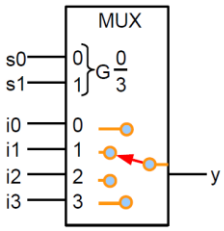


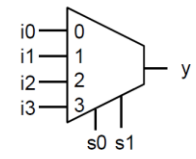
	ci	a	b	co	s
0	0	0	0	0	0
1	0	0	1	0	1
2	0	1	0	0	1
3	0	1	1	1	0
4	1	0	0	0	1
5	1	0	1	1	0
6	1	1	0	1	0
7	1	1	1	1	1



NAND a*b			NOR a+b		
a	b	Q	a	b	Q
0	0	1	0	0	1
0	1	1	0	1	0
1	0	1	1	0	0
1	1	0	1	1	0

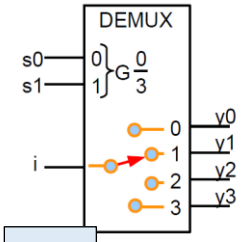
Multiplexer

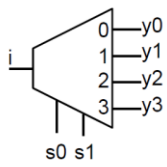




Anzahl Selekt-Leitungen (s): log2(n) aufgerundet

Demultiplexer

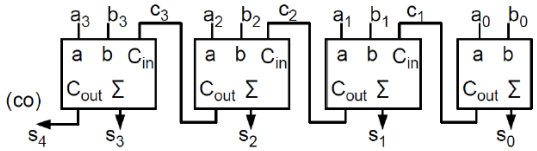




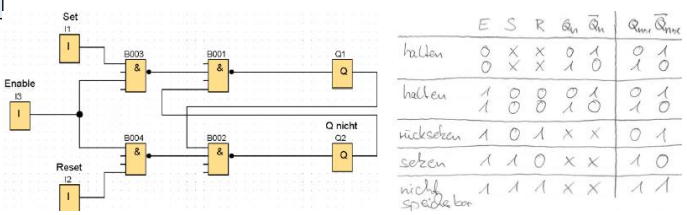
Anzahl Selekt-Leitungen (s): log2(n) aufgerundet

n-Bit-Addierer

(ripple-carry-adder wg. langer Durchlaufzeit; besser: carry look-ahead adder)



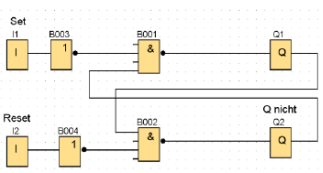
RS-L mit EN



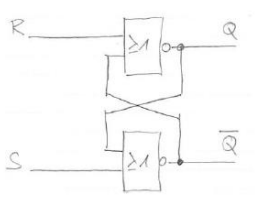
E	S	R	Q_n	$\overline{Q_n}$	Q_{n+1}	$\overline{Q_{n+1}}$
halten	0	0	0	1	0	1
	0	0	1	0	1	0
halten	1	0	0	1	0	1
	1	0	1	0	1	0
rücksetzen	1	0	1	x	0	1
setzen	1	1	0	x	1	0
nicht speicherbar	1	1	1	x	1	1

RS-L

mit NAND



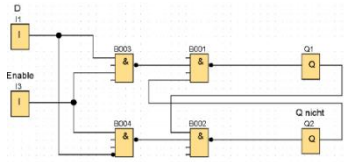
mit NOR



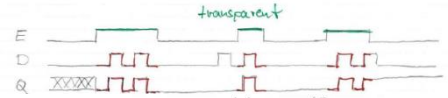
S	R	Q_n	$\overline{Q_n}$	Q_{n+1}	$\overline{Q_{n+1}}$
halten / speichern	0	0	0	1	0	1
	0	0	1	0	1	0
rücksetzen	0	1	x	x	0	1
setzen	1	0	x	x	1	0
nicht speicherbar	1	1	x	x	1	1

S	R	Q_n	$\overline{Q_n}$	Q_{n+1}	$\overline{Q_{n+1}}$
Halten / Speichern	0	0	0	1	0	1
	0	0	1	0	1	0
rücksetzen	0	1	x	x	0	1
setzen	1	0	x	x	1	0
„verboten" nicht speicherbar	1	1	x	x	0	0

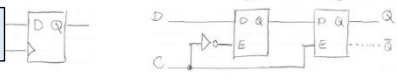
D-L

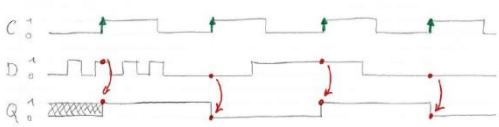


E	D	Q_n	Q_{n+1}
0	x	0	0
0	x	1	1
1	0	x	0
1	1	x	1

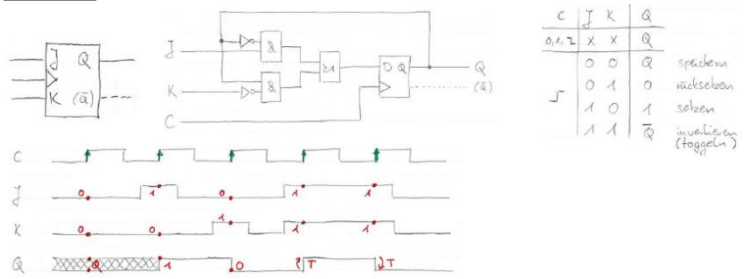


D-FF

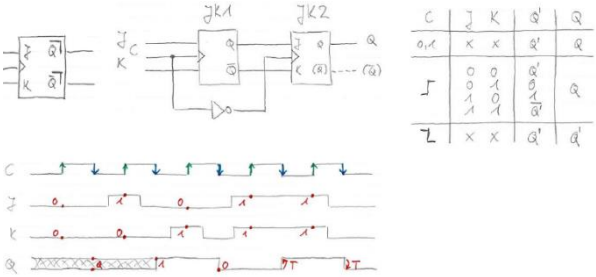




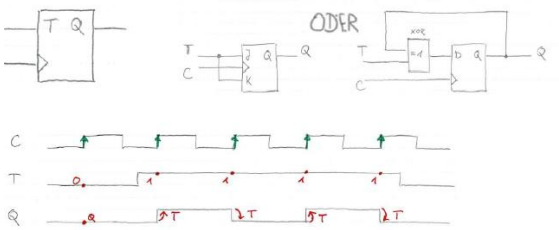
JK



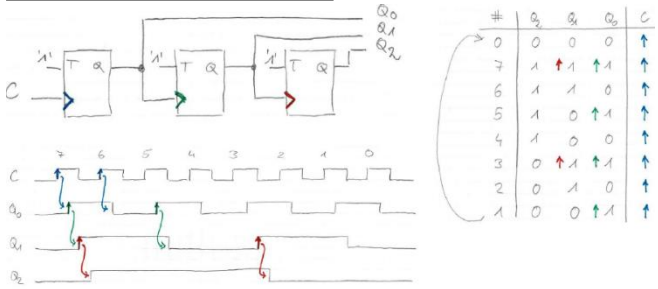
JK-MS



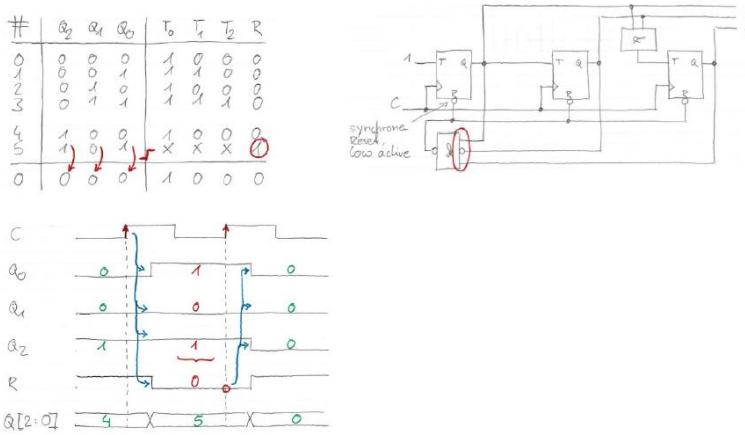
T



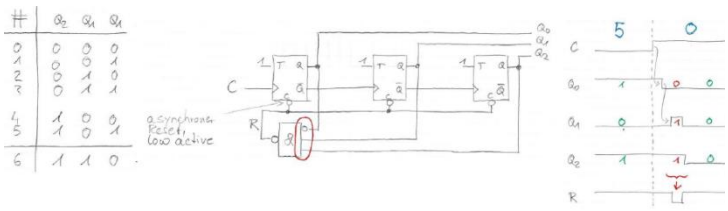
Asynchronz. rückwärts



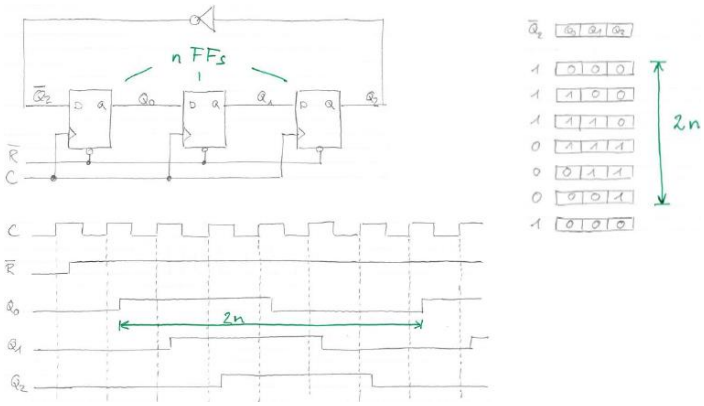
Synchronz. vorwärts + Reset



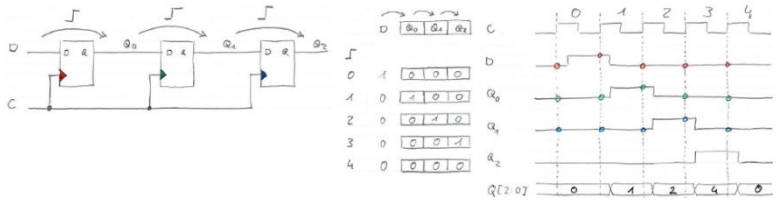
Asynchronz. vorwärts + Reset



Johnson-Zähler



Schieberegister



Linear rückgekoppeltes Schieberegister: 0-0-0 kann nicht verlassen werden u. max. Sequenzlänge 2^n-1

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

$f_{\overline{x_1}}(x_1 = 0, x_2, x_3) \;=\; (0 + x_2) \cdot (1 + x_3)$

$f_{x_1}(x_1 = 1, x_2, x_3) \;=\; (1 + x_2) \cdot (0 + x_3)$

$f = \overline{x_1} \cdot f_{\overline{x_1}} + x_1 \cdot f_{x_1}$

$= \overline{x_1} \cdot [(0 + x_2) \cdot (1 + x_3)] + x_1 \cdot [(1 + x_2) \cdot (0 + x_3)]$

$= \overline{x_1} \cdot x_2 + \overline{x_1} \cdot x_2 \cdot x_3 + x_1 \cdot x_3 + x_1 \cdot x_2 \cdot x_3$

$= \overline{x_1} \cdot x_2 + x_1 \cdot x_3$