

## Benutzte Quellen

- Vorlesungen von Dr.-Ing. Vogelmann, Universität – Karlsruhe
- Vorlesungen von Dr.-Ing. Klos, Universität – Karlsruhe
- Vorlesungen von Dr.-Ing. Crokol, Universität – Karlsruhe
- Halbleiter Schaltungstechnik, Tietze u., Schenk Th.
- <http://www.elektronik-kompodium.de/sites>
- [http://www.eelab.usyd.edu.au/digital\\_tutorial/part2/hpage.html](http://www.eelab.usyd.edu.au/digital_tutorial/part2/hpage.html)

## Benutzte Bezeichnungen

- Aufschrift der Thema, die zum ersten Mal erscheint
- Aufschrift der Thema, die schon bekannt ist

## Logik

- **Logische Grundfunktionen und Rechenregeln**
- Realisierung von Gattern
- Schaltnetze (Kombinatorische Logik)
- Schaltwerke (Sequentielle Logik)

## Logik

Lernziel:

- Antworten auf folgende und ähnliche Fragen geben zu können:
  - Wie kann man Zahlen darstellen?
  - Welche sind die logischen Grundfunktionen?
  - Was für Flip-flops gibt es und wie funktionieren sie?
  - Was für Flip-flops benutzt man bei Zähler und Register?
  - Wozu benutzt man Komparatoren?

## Logischer (Boolscher) Zustand - Bit

Logischer Zustand					Spannungswerte
1	ja	Ein	High	wahr	z.B. 2,5...5 V
0	nein	Aus	Low	falsch	z.B. 0...0,4 V

## Bit-Kombinationen

- Mit n Bits kann man  $2^n$  Zustände beschreiben
- $Z = 2^n$

n	Z
8	256
16	65.536
24	16,7 Mio
32	4.294 Mio

Bit 1	Bit 2	Zustand
0	0	0
0	1	1
1	0	2
1	1	3

Beispiel mit  $n=2$

## Darstellung einer Bit-Kombination

- Dualzahl (binäre Darstellung)

$$Z_n = 2^n \cdot b_n + 2^{n-1} \cdot b_{n-1} + \dots + 2^3 \cdot b_3 + 2^2 \cdot b_2 + 2^1 \cdot b_1 + 2^0 \cdot b_0$$

$$Z_n = \dots + 16 \cdot b_4 + 8 \cdot b_3 + 4 \cdot b_2 + 2 \cdot b_1 + 1 \cdot b_0 = \sum_{i=0}^n 2^i \cdot b_i$$

$$Z_n = b_n b_{n-1} b_{n-2} \dots b_1 b_0 \text{ mit } b_i \in \{0,1\}$$

- Hexadezimalzahl (Zahlendarstellung zur Basis 16)

$$Z_n = h_n \cdot 16^n + \dots + h_1 \cdot 16^1 + h_0 \cdot 16^0 = \sum_{i=0}^n 16^i \cdot h_i$$

$$Z_n = h_n h_{n-1} h_{n-2} \dots h_1 h_0 \text{ mit } h_i \in \{0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F\}$$

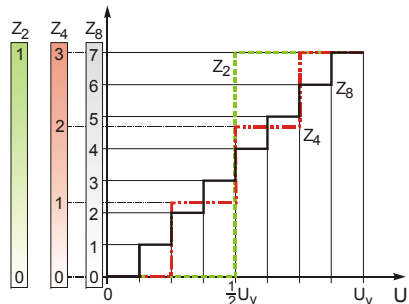
- Beispiel:  $125_{(10)} = 0b01111101 = 0x7D = 7DH$

## Zählen im Dual- und Hexadezimalsystem

Dezimal	Binär	Hex
0	00000	0
1	00001	1
2	00010	2
3	00011	3
4	00100	4
5	00101	5
6	00110	6
7	00111	7
8	01000	8
9	01001	9

Dezimal	Binär	Hex
10	01010	A
11	01011	B
12	01100	C
13	01101	D
14	01110	E
15	01111	F
16	10000	10
17	10001	11
18	10010	12
19	10011	13

## Darstellung einer analogen Spannung



## Codierung

Binärzahlen ermöglichen zunächst nur die Darstellung natürlicher Zahlen: 0,1,2... . Die Darstellung ganzer oder gebrochener Zahlen erfordert eine Codierung, die den Aufgaben entsprechend ausgewählt oder angepaßt wird.

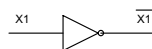
BCD-Code (Binär codiertes Dezimalsystem)

	Tausender	Hunderter	Zehner	Einer
Dezimal-Zahl	3	9	6	1
BCD-Zahl (binär)	0011	1001	0110	0001

Beim BCD-Code werden die Ziffern einer Dezimalzahl als Dualzahl codiert

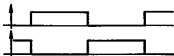
## Negation - "HE" (NOT)

$$f1(x1) = \bar{x1}$$



Schalt-symbol

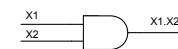
Nº	x1	f1(x1)
0	0	1
1	1	0



Zeitfunktion

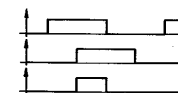
## UND-Funktion - "I" (AND)

$$f2(x1, x2) = x1 \cdot x2$$



Schalt-symbol

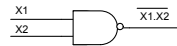
Nº	x1	x2	f2(x1,x2)
0	0	0	0
1	0	1	0
2	1	0	0
3	1	1	1



Zeitfunktion

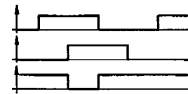
## NICHT-UND-Funktion (NAND)

$$f_3(x_1, x_2) = \overline{x_1 \cdot x_2}$$



Schaltsymbol

N°	x1	x2	f3(x1,x2)
0	0	0	1
1	0	1	1
2	1	0	1
3	1	1	0



Zeitfunktion

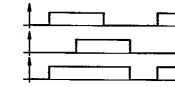
## ODER-Funktion– “ИЛИ” (OR)

$$f_4(x_1, x_2) = x_1 + x_2$$



Schaltsymbol

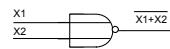
N°	x1	x2	f4(x1,x2)
0	0	0	0
1	0	1	1
2	1	0	1
3	1	1	1



Zeitfunktion

## NICHT-ODER-Funktion (NOR)

$$f_5(x_1, x_2) = \overline{x_1 + x_2}$$



Schaltsymbol

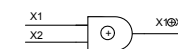
N°	x1	x2	f5(x1,x2)
0	0	0	1
1	0	1	0
2	1	0	0
3	1	1	0



Zeitfunktion

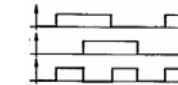
## Antivalenz (XOR)

$$f_6(x_1, x_2) = x_1 \oplus x_2$$



Schaltsymbol

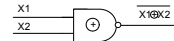
N°	x1	x2	f6(x1,x2)
0	0	0	0
1	0	1	1
2	1	0	1
3	1	1	0



Zeitfunktion

## Äquivalenz (XNOR)

$$f7(x1, x2) = \overline{x1 \oplus x2}$$



Schaltsymbol

Nº	x1	x2	f7(x1,x2)
0	0	0	1
1	0	1	0
2	1	0	0
3	1	1	1



Zeitfunktion

## Rechenregeln

### Konjunktion:

$$y = x_1 \wedge x_2 = x_1 \cdot x_2 = x_1 x_2$$

### Disjunktion:

$$y = x_1 \vee x_2 = x_1 + x_2$$

### Negation:

$$y = \overline{x} = /x$$

### Kommutatives Gesetz:

$$x_1 x_2 = x_2 x_1$$

$$x_1 + x_2 = x_2 + x_1$$

### Assoziatives Gesetz:

$$x_1 (x_2 x_3) = (x_1 x_2) x_3$$

$$x_1 + (x_2 + x_3) = (x_1 + x_2) + x_3$$

### Distributives Gesetz:

$$x_1 (x_2 + x_3) = x_1 x_2 + x_1 x_3$$

$$x_1 + x_2 x_3 = (x_1 + x_2)(x_1 + x_3)$$

### Absorptionsgesetz:

$$x_1 (x_1 + x_2) = x_1$$

$$x_1 + x_1 x_2 = x_1$$

### Tautologie:

$$xx = x$$

$$x + x = x$$

### Gesetz für die Negation:

$$x\overline{x} = 0$$

$$x + \overline{x} = 1$$

### Doppelte Negation:

$$\overline{(\overline{x})} = x$$

### De Morgans Gesetz:

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

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

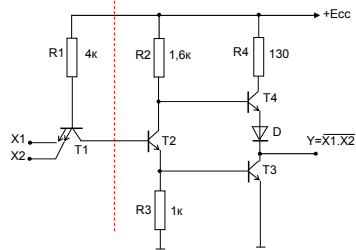
## Logik

- Logische Grundfunktionen und Rechenregeln
- Realisierung von Gattern
- Schaltnetze (Kombinatorische Logik)
- Schaltwerke (Sequentielle Logik)

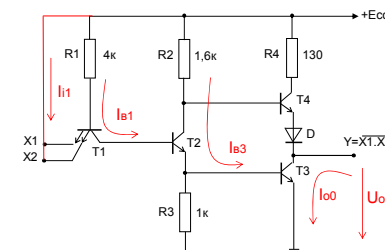
## Realisierung

	Gasättigte Logik				Ungasättigte Logik
	CMOS Complementary MOS	HCMOS High Speed-CMOS	TTL Transistor-Transistor Logic	STTL Schottky-TTL	ECL Emmitter-Coupled-Logic
$U_{cc}$	(5...15) V	5 V	5 V	5 V	5 V
Ein	11...14,8 V	4,9 V	3,6 V	3,5 V	-0,9 V
Aus	0,2...4 V	0,1 V	0,5 V	0,5 V	-1,7 V
Schaltgeschwindigkeit	35 ns	8 ns	10 ns	4 ns	1 ns
Leistungsaufnahme	10 nW	25 nW	10 mW	20 mW	25 mW

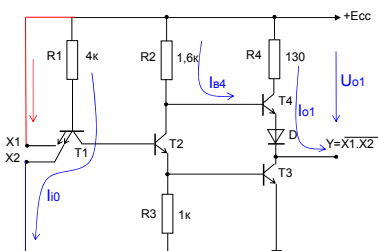
# Realisierung TTL



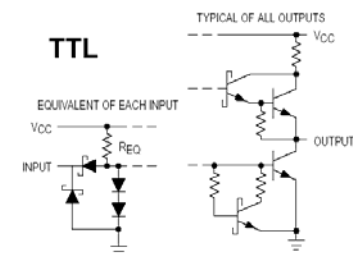
# Realisierung TTL



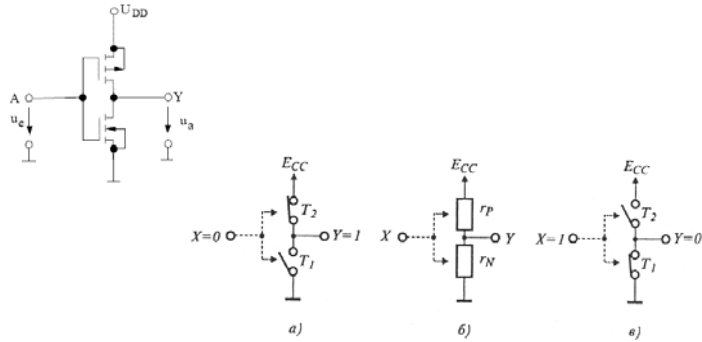
# Realisierung TTL



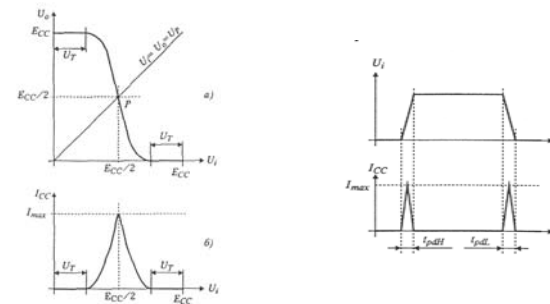
# Realisierung TTL



# Realisierung CMOS

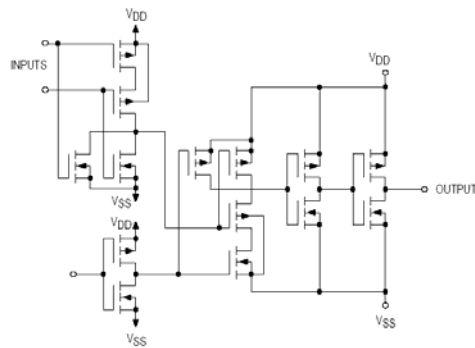


# Realisierung CMOS

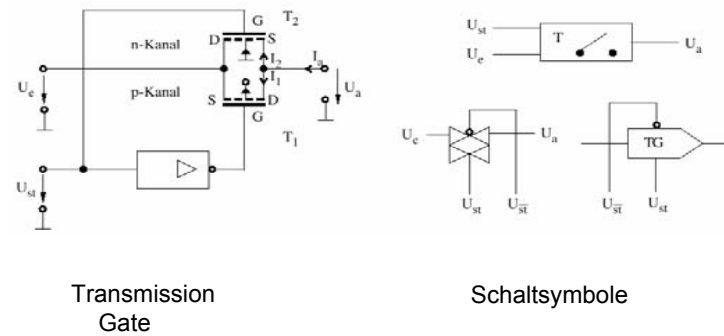


# Realisierung CMOS

CMOS



# Realisierung CMOS



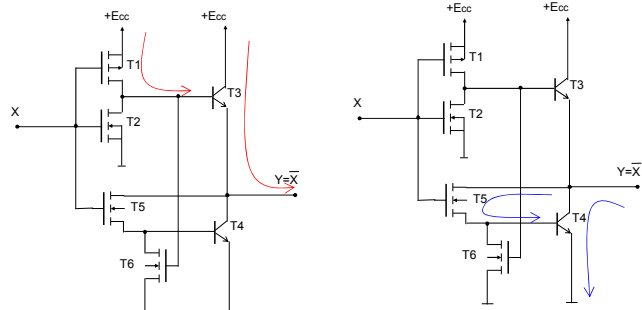
Transmission Gate

Schaltsymbole

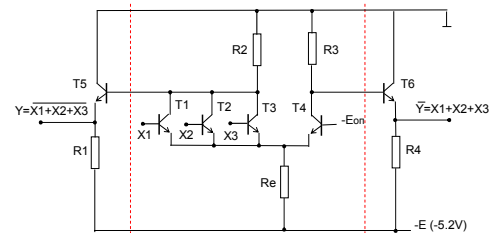
# Realisierung BiCMOS

X = 0, T1,T3,T6 leitend

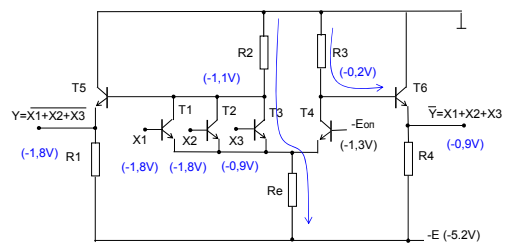
X = 1, T2,T5,T4 leitend



# Realisierung ECL

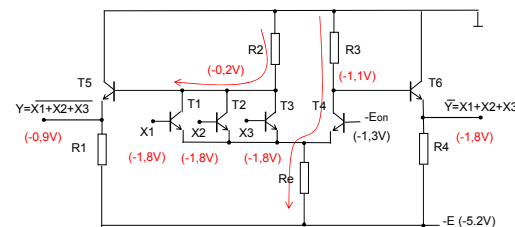


# Realisierung ECL



$X1, X2 = 0; X3 = 1 \rightarrow Y = 0; \bar{Y} = 1$

# Realisierung ECL

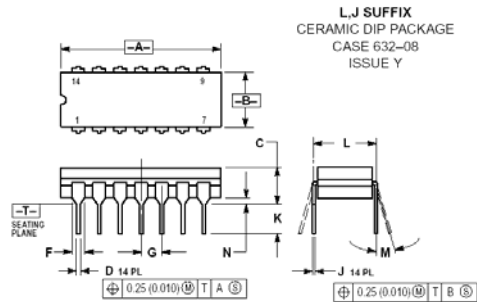


$X1, X2, X3 = 0; \rightarrow Y = 1; \bar{Y} = 0$



## Realisierung

### 14-Pin Packages



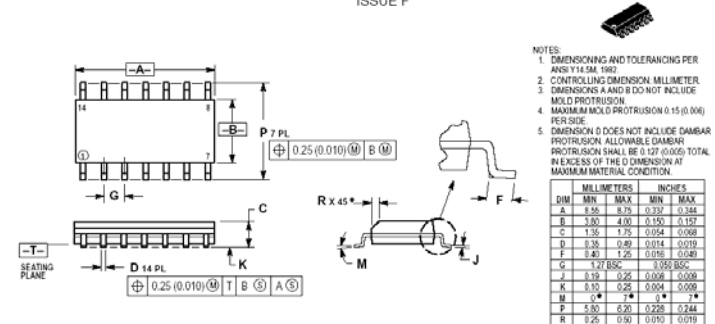
NOTES:

1. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1992.
2. CONTROLLING DIMENSION: INCH.
3. DIMENSION L TO CENTER OF LEAD WHEN FORMED PARALLEL.
4. DIMENSION F MAY NARROW TO 0.76 (0.030) WHERE THE LEAD ENTERS THE CERAMIC BODY.

DIM	INCHES		MILLIMETERS	
	MIN	MAX	MIN	MAX
A	0.750	0.785	19.05	19.94
B	0.245	0.280	6.23	7.11
C	0.150	0.200	3.94	5.08
D	0.015	0.050	0.39	0.90
F	0.050	0.080	1.27	1.95
G	0.100 BSC		2.54 BSC	
J	0.008	0.015	0.21	0.38
K	0.025	0.100	0.64	2.54
L	0.200 BSC		5.08 BSC	
M	0.015	0.015	0.38	0.38
N	0.020	0.040	0.51	1.01

## Realisierung

### D SUFFIX PLASTIC SOIC PACKAGE CASE 751A-03 ISSUE F



NOTES:

1. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1992.
2. CONTROLLING DIMENSION: MILLIMETER.
3. DIMENSIONS A AND B DO NOT INCLUDE MOLD PROTRUSION.
4. MAXIMUM MOLD PROTRUSION 0.15 (0.006) PER SIDE.
5. DIMENSION D DOES NOT INCLUDE DAMBAR PROTRUSION. ALLOWABLE DAMBAR PROTRUSION SHALL BE 0.127 (0.005) TOTAL IN EXCESS OF THE D DIMENSION AT MAXIMUM MATERIAL CONDITION.

DIM	MILLIMETERS		INCHES	
	MIN	MAX	MIN	MAX
A	8.65	8.75	0.337	0.344
B	1.90	4.00	0.075	0.157
C	1.30	1.75	0.051	0.069
D	0.35	0.49	0.014	0.019
F	0.81	1.25	0.032	0.049
G	0.27 BSC		0.010 BSC	
J	0.19	0.25	0.008	0.009
K	0.10	0.25	0.004	0.009
M	0.15	0.15	0.006	0.006
P	5.30	6.20	0.208	0.244
R	0.25	0.50	0.010	0.019

## Logik

- ❑ Logische Grundfunktionen und Rechenregeln
- ❑ Realisierung von Gattern
- ❑ **Schaltnetze (Kombinatorische Logik)**
- ❑ Schaltwerke (Sequentielle Logik)

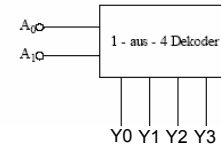
## Schaltnetze (Kombinatorische Logik)

- ❑ Ein Schaltnetz enthält keine Speicher.
- ❑ Die Ausgangsvariablen sind eindeutig von den Eingangsvariablen abhängig.
- ❑ Schaltnetze werden durch Wahrheitstafeln (Tabellen) oder boolesche Gleichungen beschrieben.
- ❑ Realisiert werden sie mit Gattern, ROMs oder programmierbaren logischen Bausteinen (PLDs).
- Beispiele: Kodierschaltungen, Multiplexer, Komparatoren, Addierer.

# Schaltnetze (Kombinatorische Logik)

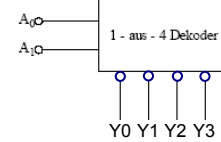


# 1-aus-n-Dekoder



N°	A1	A0	Y0	Y1	Y2	Y3
0	0	0	1	0	0	0
1	0	1	0	1	0	0
2	1	0	0	0	1	0
3	1	1	0	0	0	1

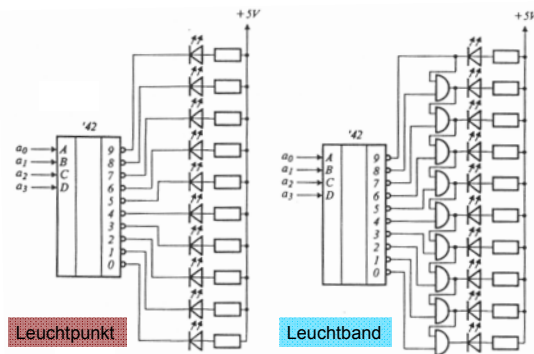
Steuerung von LCD



N°	A1	A0	Y0	Y1	Y2	Y3
0	0	0	0	1	1	1
1	0	1	1	0	1	1
2	1	0	1	1	0	1
3	1	1	1	1	1	0

Steuerung von LED

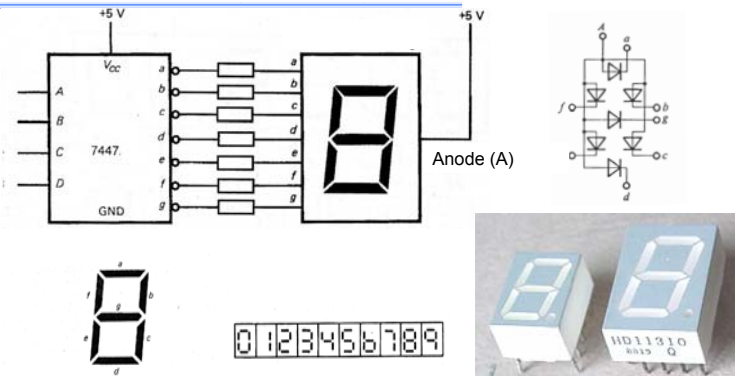
# BCD-Dekoder (Anzeige)



Leuchtpunkt

Leuchtband

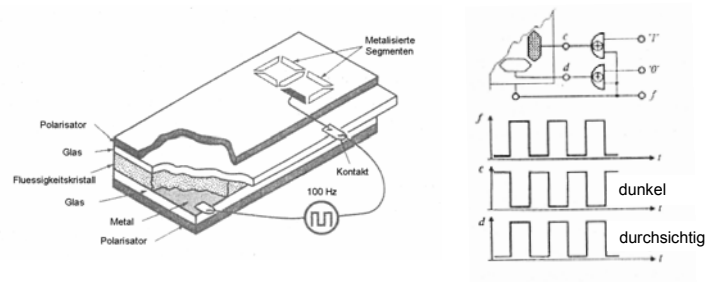
# Sieben-Segment-Dekoder (Anzeige)



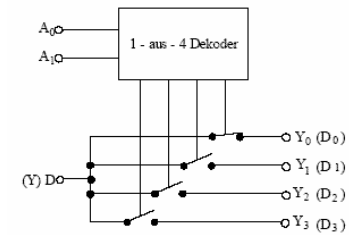
0	1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---	---



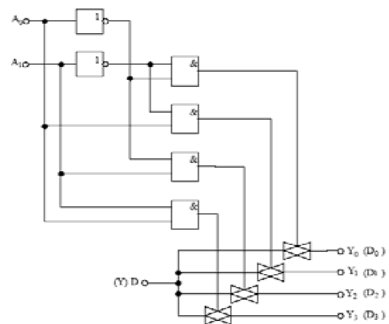
## Flüssigkeitskristalle (LCD)



## Multiplexer/Demultiplexer



## Multiplexer/Demultiplexer



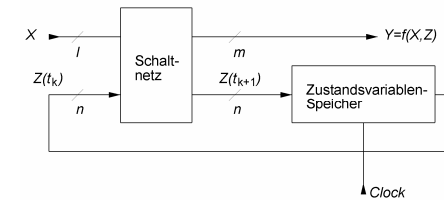
## Logik

- Logische Grundfunktionen und Rechenregeln
- Realisierung von Gattern
- Schaltnetze (Kombinatorische Logik)
- Schaltwerke (Sequentielle Logik)

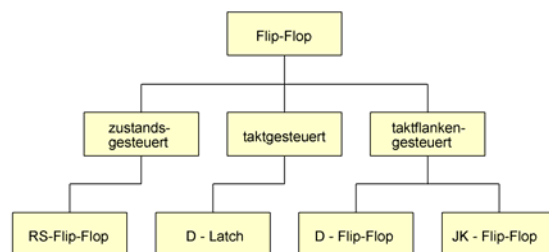
## Schaltwerke (Sequentielle Logik)

- Ein Schaltwerk enthält Variablenpeicher.
- Die Ausgangsvariablen  $Y$  hängen von den Eingangsvariablen  $X$  und dem Zustand des Systems  $Z$  ab.
- Der Zustand wird bitweise in Flip-Flops gespeichert.
- Schaltwerke werden durch Zustandstabellen oder Flußdiagramme beschrieben.
- Realisiert werden sie mit Flip-Flops oder programmierbaren logischen Bausteinen (PLDs).
- Beispiele: Zähler, Zeitgeber, Datenspeicher.

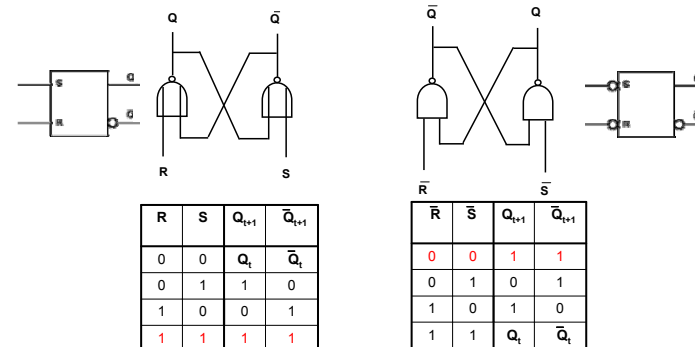
## Schaltwerke (Sequentielle Logik)



## Flip-Flops

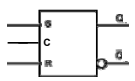
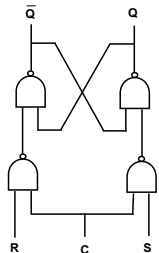


## Transparente Flip-Flops



Logisch unzulässig

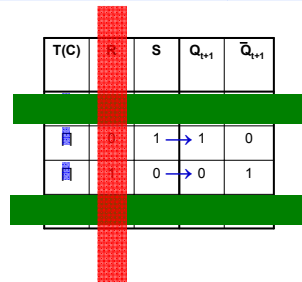
# Taktzustandgesteuertes RS-Flip-Flop



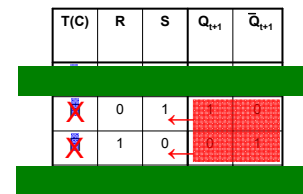
T(C)	R	S	Q <sub>t+1</sub>	Q̄ <sub>t+1</sub>
0	0	0	Q <sub>t</sub>	Q̄ <sub>t</sub>
0	1	1	1	0
1	0	0	0	1
1	1	1	1	1

Logisch unzulässig

# Weiterentwicklung

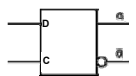
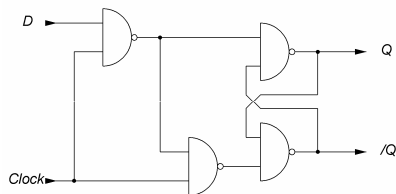


D-Flip flop

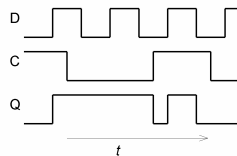


T-Flip flop

# Taktzustandgesteuertes D-Flip-Flop

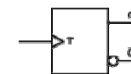
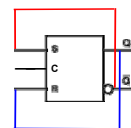


D	Clock	Q
0	0	keine Änderung
0	1	0
1	0	keine Änderung
1	1	1



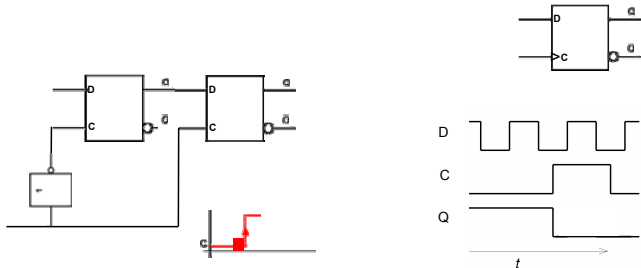
Transparentes D-Flip-Flop (D-Latch)

# Flankengesteuerte T-Flip-Flops



# Einflankengesteuerte Flip-Flops

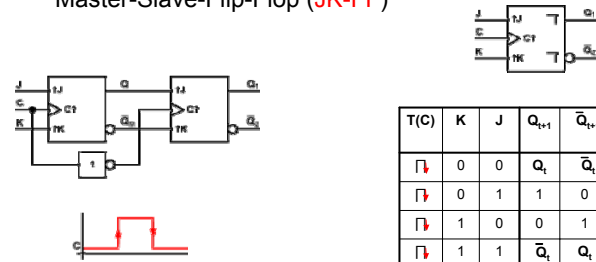
D-Flip-Flop (zweistufige Struktur)



Bauelemente für Register

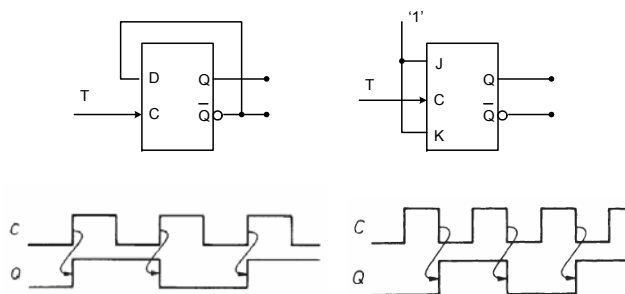
# Zweiflankengesteuerte Flip-Flops

Master-Slave-Flip-Flop (JK-FF)

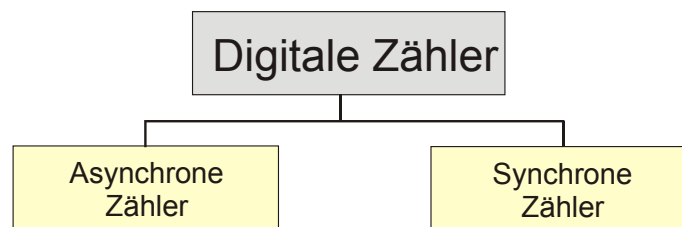


Bauelemente für Zähler

# T-Flip-Flops

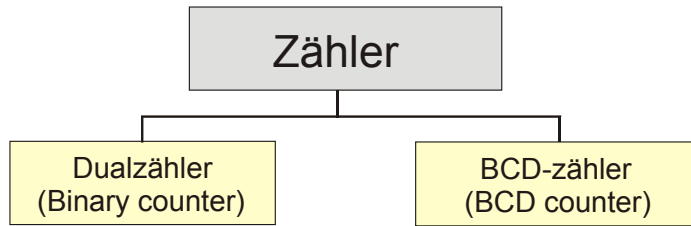


# Zähler

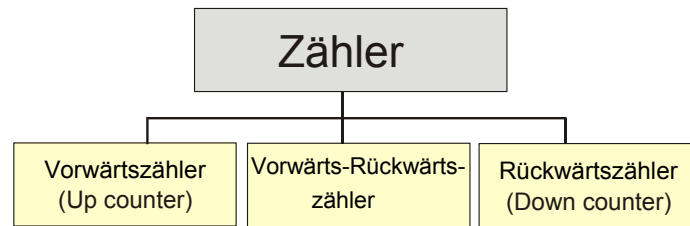


9\_FF.cdr ØETI Vo

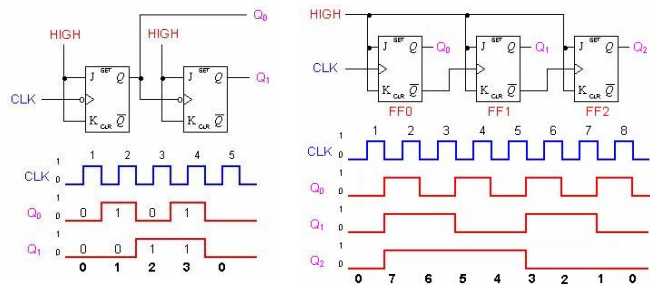
# Zähler



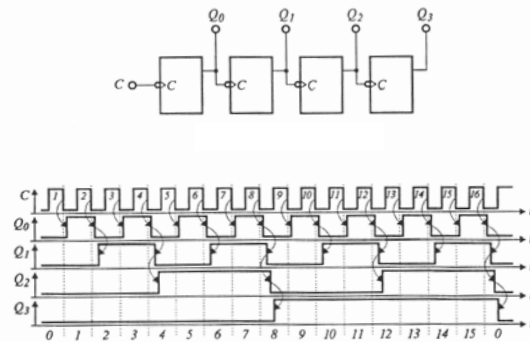
# Zähler



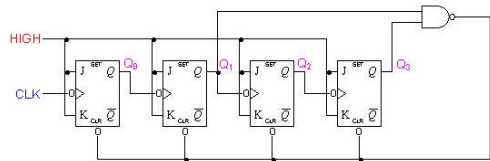
# Asynchrone Zähler (Binary)



# Asynchrone Zähler (Binary)

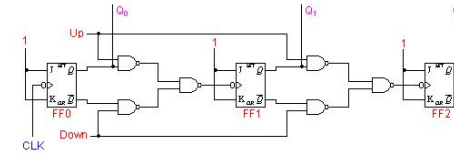


# Asynchrone Zähler (BCD)



Clock Pulse	Q3	Q2	Q1	Q0
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1

# Asynchrone Zähler

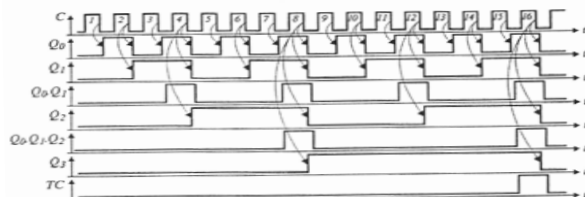
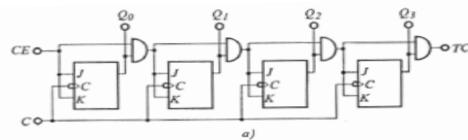


Vorwärts-Rückwärts-Zähler

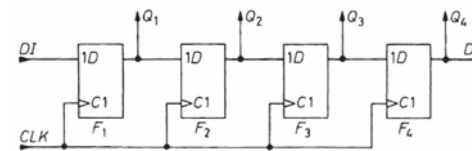
Up = 1, Down = 0 Vorwärts

Up = 0, Down = 1 Rückwärts

# Synchrone Zähler



# Register



CLK	Q1	Q2	Q3	Q4
1	D1	-	-	-
2	D2	D1	-	-
3	D3	D2	D1	-
4	D4	D3	D2	D1
5	D5	D4	D3	D2
6	D6	D5	D4	D3
7	D7	D6	D5	D4



# Register

