

1. Grundlagen

- Binär: Es existieren nur zwei definierte Zustände (0, 1), (L, H), (0V, 5V),...
- Redundanz: Anzahl notwendige Bitstellen, um Zustände zu kodieren: $Z = 2n$ (n =Anzahl Bits)
- Kontinuierlich \leftrightarrow Diskret
- Jedes logische Element besitzt ein Toleranzfeld zwischen 0 (0V-0.8V) und 1 (4.5V-5.5V) für CMOS

2. Gatter

AND	OR	NOT	NAND	NOR	XNOR	XOR																																																																																
$Y = A \cdot B$ $Y = A \wedge B$	$Y = A + B$ $Y = A \vee B$	$Y = \bar{A}$	$Y = \bar{A} \cdot \bar{B}$ $Y = \bar{A} \wedge \bar{B}$	$Y = \bar{A} + \bar{B}$ $Y = \bar{A} \vee \bar{B}$	$Y = \bar{A} \oplus \bar{B}$ $Y = A \cdot B + \bar{A} \cdot \bar{B}$ $Y = ABC + \bar{A}\bar{B}\bar{C}$	$Y = A \oplus B$ $Y = A \cdot \bar{B} + \bar{A} \cdot B$ $Y = (A + B)(\bar{A} + \bar{B})$																																																																																
<table border="1"> <tr><th>A</th><th>B</th><th>Y</th></tr> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </table>	A	B	Y	0	0	0	0	1	0	1	0	0	1	1	1	<table border="1"> <tr><th>A</th><th>B</th><th>Y</th></tr> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </table>	A	B	Y	0	0	0	0	1	1	1	0	1	1	1	1	<table border="1"> <tr><th>A</th><th>Y</th></tr> <tr><td>0</td><td>1</td></tr> <tr><td>1</td><td>0</td></tr> </table>	A	Y	0	1	1	0	<table border="1"> <tr><th>A</th><th>B</th><th>Y</th></tr> <tr><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </table>	A	B	Y	0	0	1	0	1	1	1	0	1	1	1	0	<table border="1"> <tr><th>A</th><th>B</th><th>Y</th></tr> <tr><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </table>	A	B	Y	0	0	1	0	1	0	1	0	0	1	1	0	<table border="1"> <tr><th>A</th><th>B</th><th>Y</th></tr> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </table>	A	B	Y	0	0	0	0	1	1	1	0	1	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	Y																																																																																					
0	1																																																																																					
1	0																																																																																					
A	B	Y																																																																																				
0	0	1																																																																																				
0	1	1																																																																																				
1	0	1																																																																																				
1	1	0																																																																																				
A	B	Y																																																																																				
0	0	1																																																																																				
0	1	0																																																																																				
1	0	0																																																																																				
1	1	0																																																																																				
A	B	Y																																																																																				
0	0	0																																																																																				
0	1	1																																																																																				
1	0	1																																																																																				
1	1	0																																																																																				
					Y = 1, wenn Eingänge gleich sind	Y = 1, wenn beide Eingänge ungleich sind $XOR = \overline{XNOR}$																																																																																

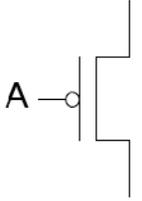
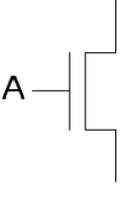
- Mit AND, OR und NOT lassen sich alle möglichen log. Verknüpfungen realisieren
- Dasselbe gilt auch für NAND oder NOR
- *Analyse*: gegebenes Schaltnetz untersuchen
- *Synthese*: Schaltnetz gemäss Anforderung konstruieren

Schaltungen ausschliesslich mit NAND- oder NOR-Gattern:

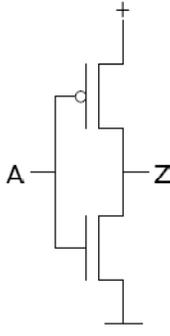
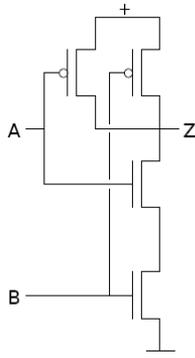
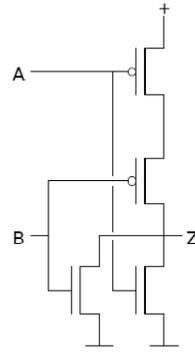
	NAND	NOR
Inverter		
UND		
ODER		

3. Logikschaltungen

- MOS-Schaltungen sind wegen ihres symmetrischen Aufbaus *unipolar*: Drain und Source sind vertauschbar
- Keine Stromaufnahme im durchgeschalteten Zustand
- Keine statische Leitung zum Ansteuern nötig

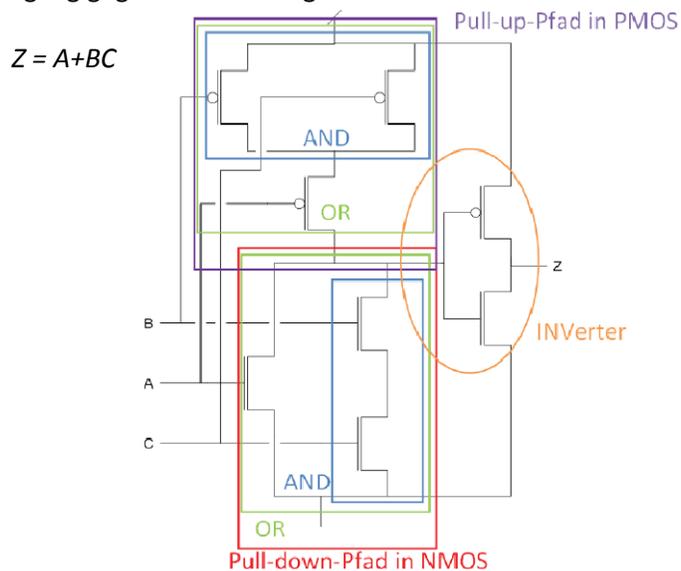
PMOS	NMOS
 <ul style="list-style-type: none"> • Schaltet durch für A low • Nur gegen +5V verwenden 	 <ul style="list-style-type: none"> • Schaltet durch für A high • Nur gegen Masse verwenden

• CMOS-Grundgatter:

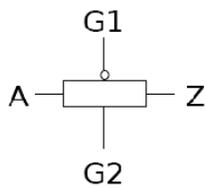
Inverter	NAND	NOR
		

• Konstruktion von CMOS-Gattern:

1. Funktion als Boolesche Algebra aufschreiben
2. pull-DOWN Pfad aus NMOS erstellen (AND=Serie, OR=Parallel)
3. pull-UP Pfad aus PMOS erstellen: komplementär zum pull-down Pfad
4. Der negierte Ausgang verbindet den pull-up und den pull-down Pfad
5. Ausgang gegebenenfalls negieren



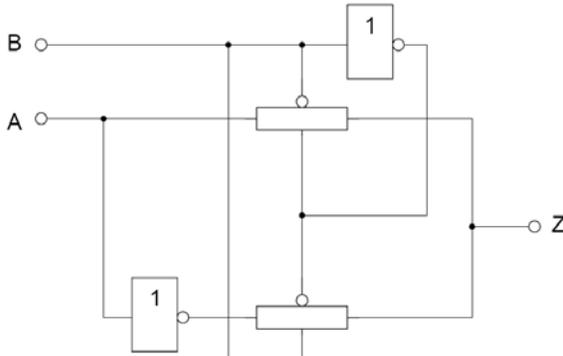
- Transmission-Gate als Beispiel eines Tristate-Buffers (kontrollierte Schaltung)



$G1$	$G2$	Z
0	0	floating (undefiniert)
0	1	A (niederohmig)
1	0	Gate sperrt (hochohmig)
1	1	Nicht erlaubt, Kurzschluss

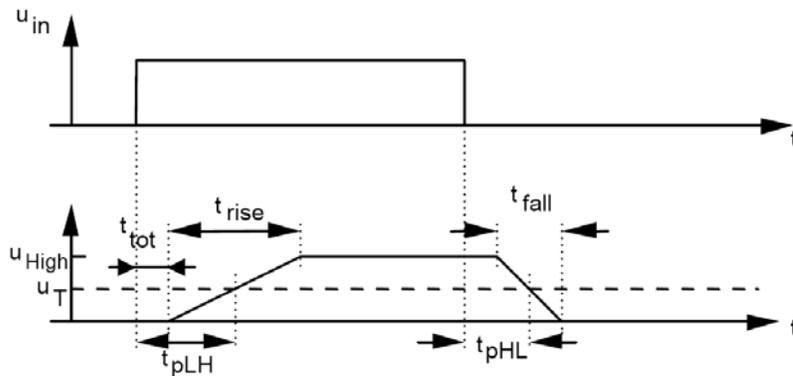
→ $G1$ immer $\overline{G2}$

Beispiel für EXOR: $Z = A \oplus B = A \cdot \overline{B} + \overline{A} \cdot B$



Für B low wird A durchgeschaltet (oberes T-Gate).
Für B high wird \overline{A} durchgeschaltet (unteres T-Gate).

- Dynamisches Verhalten von CMOS (hier: Inverter)



- t_{pHL} : Verzögerung Übergang H → L bei 50% des Pegelbusses
- t_{pLH} : Verzögerung Übergang L → H bei 50% des Pegelbusses
- $t_{R/TLH}$: Rise-Zeit (i.d.R. zwischen 10% und 90% des Pegelbusses)
- $t_{F/THL}$: Fall-Zeit (i.d.R. zwischen 90% und 10% des Pegelbusses)
- U_T : Threshold zwischen H und L

- Wie lange vor akt. Taktflanke müssen Eingänge stabil sein?
→ Längsten Pfad vom Eingang bis FF: $t_{Gatter} + t_{setupFF}$
- Wann nach akt. Taktflanke ist der Ausgang stabil?
→ Längsten Pfad von FF bis Ausgang: $t_{Gatter} + t_{PDFF}$
- Kürzester Taktzyklus?
→ Längsten Pfad von FF-Ausgang bis FF-Eingang: $t_{PDFF} + t_{Gatter} + t_{SetupFF} (+t_{hold})$

4. Schaltalgebra:*Allgemein:*

- Klammer vor UND vor ODER
- UND = Serieschaltung
- ODER = Parallelschaltung

De Morgansche Regeln

1. $\overline{A+B} = \overline{A} \cdot \overline{B}$
2. $\overline{A \cdot B} = \overline{A} + \overline{B}$
3. $A \cdot B = \overline{\overline{A \cdot B}} = \overline{\overline{A} + \overline{B}}$ (NOR)
4. $A + B = \overline{\overline{A+B}} = \overline{\overline{A} \cdot \overline{B}}$

- ➔ Jedes der Universalgatter NOR oder NAND ist ausreichend
- ➔ Gilt auch für mehrere Variablen

Weitere Regeln:

$$\overline{0} = 1$$

$$\overline{1} = 0$$

$$\overline{\overline{A}} = A$$

$$A \cdot B \cdot C = C \cdot B \cdot A$$

Kommutativität

$$A + B + C = C + B + A$$

$$(A \cdot B) \cdot C = C \cdot (B \cdot A)$$

Assoziativität

$$(A + B) + C = C + (B + A)$$

$$A \cdot (B + C) = A \cdot B + A \cdot C$$

Distributivität

$$A + (B \cdot C) = A + B \cdot A + C$$

$$A \cdot A = A$$

Idempotenz

$$A + A = A$$

$$A \cdot \overline{A} = 0$$

Komplemente

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

$$A \cdot 1 = A$$

Neutrale

$$A + 0 = A$$

$$A \cdot 0 = 0$$

Dominanz

$$A + 1 = 1$$

$$A \cdot (A + B) = A, \text{ da } (A + 0) \cdot (A + B) = A + (0 \cdot B) = A$$

Absorption

$$A + (A \cdot B) = A, \text{ da } (A \cdot 1) + (A \cdot B) = A \cdot (1 + B) = A$$

Beispiele:

$$- (A \cdot \overline{B \cdot C}) + A + B \cdot C = A \cdot (\overline{B \cdot C} + 1) + B \cdot C = A \cdot 1 + B \cdot C = A + B \cdot C$$

$$- \text{XOR: } A \oplus B = \overline{A}B + A\overline{B} = \overline{A}B + \overline{A\overline{B}} = \overline{\overline{A} + \overline{B}} + \overline{A} + \overline{B}$$

$$- \text{XNOR: } \overline{A \oplus B} = \overline{A}B + AB = \overline{\overline{A} + \overline{B}} + \overline{\overline{A} + \overline{B}} = \overline{\overline{A} + \overline{B}}$$

$$- (A \cdot D) + (B \cdot C \cdot D) + (A \cdot \overline{D} \cdot \overline{C}) + (\overline{A} \cdot B \cdot C) + (A \cdot C \cdot \overline{D}) = A \cdot (D + \overline{D} \cdot \overline{C} + C \cdot \overline{D}) +$$

$$(B \cdot C)(D + \overline{A}) = A \cdot (D + \overline{D} \cdot (C + \overline{C})) + B \cdot C \cdot (D + \overline{A}) = A + B \cdot C(D + \overline{A}) = A \cdot$$

$$(B \cdot C + \overline{B \cdot C}) + B \cdot C \cdot D + B \cdot C \cdot \overline{A} = B \cdot C(A + D + \overline{A}) + A \cdot \overline{B \cdot C} + A \cdot B \cdot C = B \cdot C + A \cdot$$

$$(B \cdot C + \overline{B \cdot C}) = A + BC$$

5. Schaltungssynthese

- Minterme (Vollkonjunktion): UND-Verknüpfungen ($10 = A \cdot \bar{B}$)
- Maxterme (Volldisjunktion): ODER-Verknüpfungen ($10 = \bar{A} + B$)
- **Kanonisch disjunktive Normalformen** (ODER-Normalform): disjunktive Verknüpfung aller Minterme: $Z = ABC\bar{C} + ABC + \bar{A}BC$: nicht vereinfacht
- **Kanonisch konjunktive Normalformen** (UND-Normalform): konjunktive Verknüpfung aller Maxterme: $Z = (A + \bar{B} + C)(A + B + \bar{C})$: nicht vereinfacht
- Jede Schaltfunktion ist darstellbar in der kanonisch disjunktiven Normalform und in der kanonisch konjunktiven Normalform → Dualität
- Vereinfachungen: Schaltalgebra, KV-Diagramm, Methode von Quine-McCluskey

Ausfüllen einer Wahrheitstabelle (immer *vollständig ausfüllen* mit evtl. Don't-Cares):

A	B	C	D	Z
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	1
0	1	1	0	0
0	1	1	1	0
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	1
1	1	0	1	1
1	1	1	0	0
1	1	1	1	0

CD \ AB	00	01	11	10
00	1	0	0	1
01	1	1	0	0
11	1	1	0	0
10	1	1	0	1

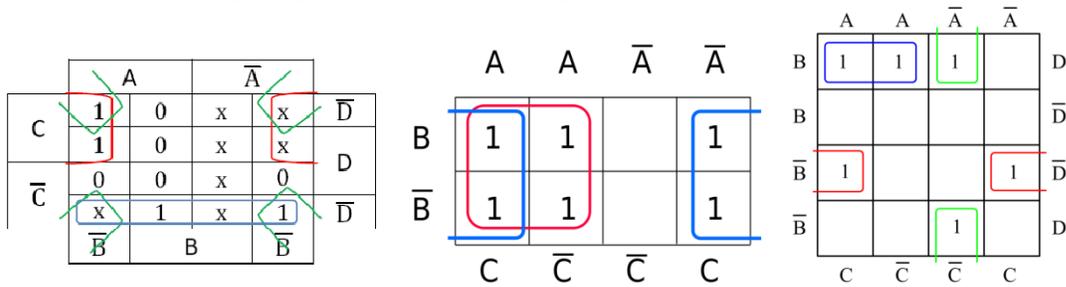
Einfüllreihenfolge:

CD \ AB	00	01	11	10
00	1	2	4	3
01	5	6	8	7
11	13	14	16	15
10	9	10	12	11

KV-Diagramm ↔ Logikfunktion:

Disjunktive Normalform (Minterme)	Konjunktive Normalform (Maxterme)																																																		
<table border="1"> <thead> <tr> <th>CD \ AB</th> <th>00</th> <th>01</th> <th>11</th> <th>10</th> </tr> </thead> <tbody> <tr> <th>00</th> <td>1</td> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <th>01</th> <td>1</td> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <th>11</th> <td>1</td> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <th>10</th> <td>1</td> <td>1</td> <td>0</td> <td>1</td> </tr> </tbody> </table> <p>$Z = A\bar{C} + B\bar{C} + \bar{B}\bar{D}$</p> <ul style="list-style-type: none"> - Blöcke von 1 bilden - Möglichst wenige, möglichst grosse 	CD \ AB	00	01	11	10	00	1	0	0	1	01	1	1	0	0	11	1	1	0	0	10	1	1	0	1	<table border="1"> <thead> <tr> <th>CD \ AB</th> <th>00</th> <th>01</th> <th>11</th> <th>10</th> </tr> </thead> <tbody> <tr> <th>00</th> <td>1</td> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <th>01</th> <td>1</td> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <th>11</th> <td>1</td> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <th>10</th> <td>1</td> <td>1</td> <td>0</td> <td>1</td> </tr> </tbody> </table> <p>$Z = (A + B + \bar{D})(\bar{C} + \bar{D})(\bar{B} + \bar{C})$</p> <ul style="list-style-type: none"> - Blöcke von 0 bilden - Möglichst wenige, möglichst grosse - Variablen hier im invertierten Zustand verwenden! 	CD \ AB	00	01	11	10	00	1	0	0	1	01	1	1	0	0	11	1	1	0	0	10	1	1	0	1
CD \ AB	00	01	11	10																																															
00	1	0	0	1																																															
01	1	1	0	0																																															
11	1	1	0	0																																															
10	1	1	0	1																																															
CD \ AB	00	01	11	10																																															
00	1	0	0	1																																															
01	1	1	0	0																																															
11	1	1	0	0																																															
10	1	1	0	1																																															

- **Don't Care:** Dies sind Zustände, die nicht auftreten können oder nicht definiert sind. Solche Zustände werden meistens mit 'X' gekennzeichnet und dürfen sowohl 0 als auch 1 annehmen. Solche Zustände sollte man zugunsten der Minimierung von KV-Diagrammen einsetzen.
- Zusammenfassungsmöglichkeiten von KV-Diagrammen:

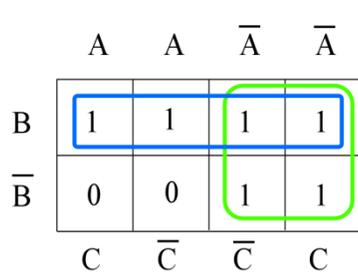


- Päckchen mit 2^n Elementen
- Über Ecken und Grenzen hinweg erlaubt

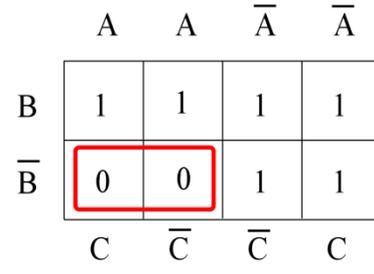
Bsp: Lampe mit Schalter

C	B	A	Lampe
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

als Minterme: $L = \bar{A} \vee B$



als Maxterme: $L = \bar{A} \vee B$



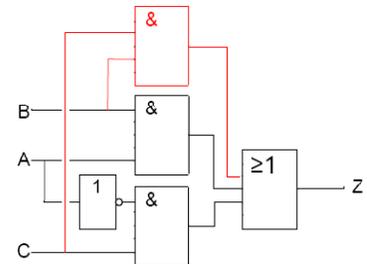
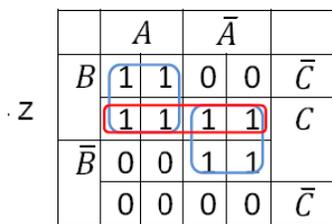
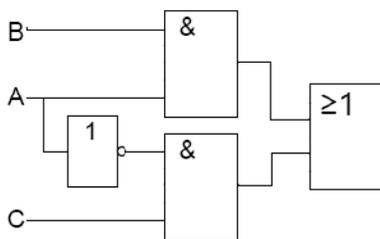
→ Min- und Maxterme sind äquivalent

- **Hazards:** Sie entstehen, wenn zwei Zustandsvariablen gleichzeitig von einem logischen Zustand in den anderen wechseln. Im KV-Diagramm sind sie als benachbarte Päckchen erkennbar. Als Ausweg muss man eine redundante Verknüpfung hinzufügen: benachbarte Päckchen überbrücken

Bsp1 (disjunktive Normalform):

$Z = AB + A\bar{C}$ (mit Hazards) →

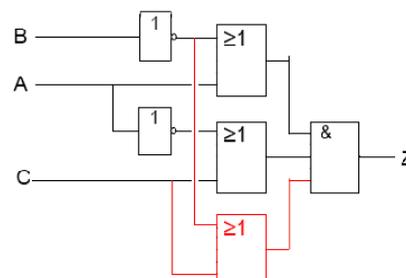
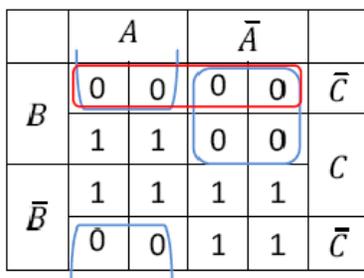
Hazard-freie Lösung: $Z = AB + A\bar{C} + BC$



Bsp2 (konjunktive Normalform):

$Z = (A + \bar{B})(\bar{A} + C)$ (mit Hazards) →

Hazard-freie Lösung: $Z = (A + \bar{B})(\bar{A} + C)(\bar{B} + C)$



6. Programmierbare Bausteine

Techniken für PLD (programmable logic devices):

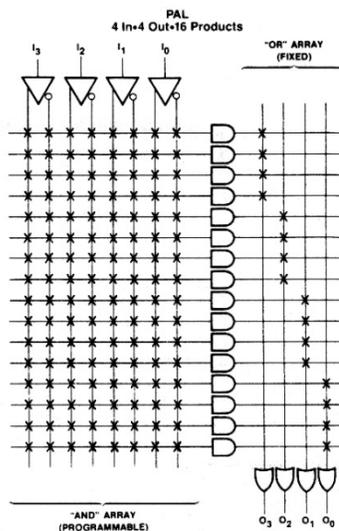
1. SRAM: in einer SRAM-Zelle können 1-Bit-Informationen gespeichert werden (vereinfachtes SR-Latch); überschreibbar, aber Verlust bei Spannungsunterbruch
2. Antifuse-Technik: durch Spannungsimpuls von 10V werden Leiterbahnkreuzungen geschmolzen. Platzsparend, kleiner Widerstand, nicht reversibel
3. "Floating Gate": Ladungen auf einem Zusatzgate (floating gate) schalten einen MOS-Transistor permanent ein oder aus. Bsp: EEPROM, durch elektrischen Impuls löschtbar

PAL (programmable array logic):

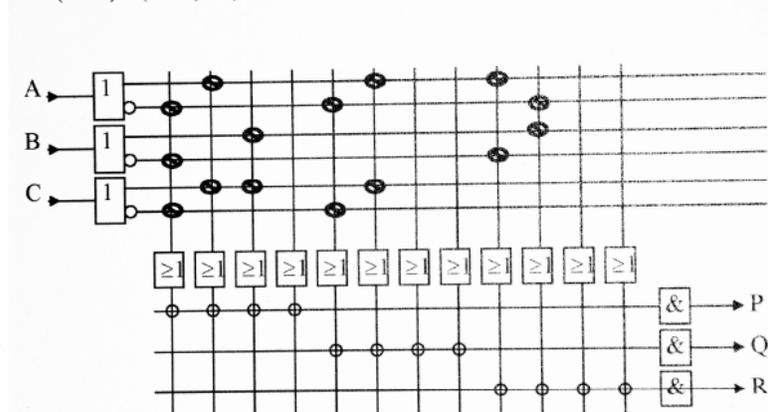
- PALs bestehen aus programmierbaren UND-Arrays und fest verdrahteten ODER-Verknüpfungen (Matrix)
- Bei PLAs sind auch noch die ODER-Verknüpfungen programmierbar → verdrängt von PAL
- Schaltungstechnisch werden die UND-Verknüpfungen durch „wired-AND“-Technik realisiert

Bsp1: Eingangsverstärker mit einem negierten und nicht-negierten Ausgang

Bsp2 (aus einer alten Basisprüfung):

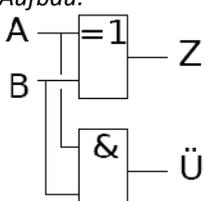
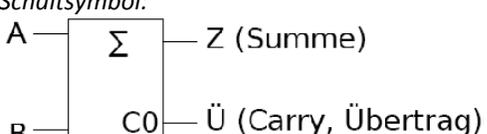
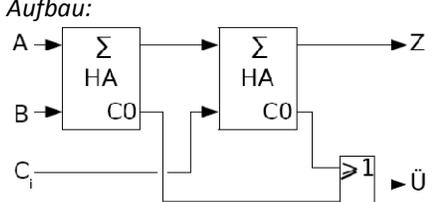


$$R = (A \cdot B) + (\bar{A} \cdot \bar{C}) = (\bar{A} + \bar{B} + \bar{C}) \cdot (B + C) \cdot (A + C) \text{ (nicht verlangt)}$$



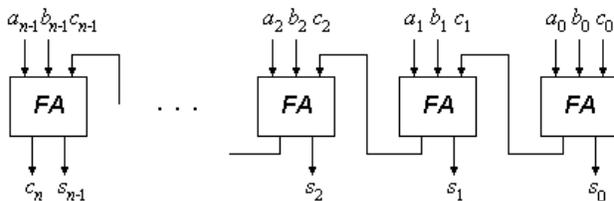
- **FPGA** (Feldprogrammierbare Gate-Arrays): bestehen aus 8*8 bis 22*22 Logikblöcke und die Verbindungen zwischen den Logikblöcken sowie die Konfiguration der Logikblöcke selber sind programmierbar

7. Rechenschaltungen

Halbaddierer				Volladdierer																																																														
<table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <tr><th>A</th><th>B</th><th>Z</th><th>Ü</th></tr> <tr><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td></tr> </table> <p style="margin-top: 10px;"> $Z = A \oplus B$ $\ddot{U} = A \cdot B$ </p> <p style="margin-top: 10px;"> Aufbau:  </p> <p style="margin-top: 10px;"> Schaltsymbol:  </p>	A	B	Z	Ü	0	0	0	0	0	1	1	0	1	0	1	0	1	1	1	1	<table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <tr><th>A</th><th>B</th><th>C_i</th><th>Z</th><th>Ü</th></tr> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr> </table> <p style="margin-top: 10px;"> $Z = A \oplus B \oplus C_i = \bar{A} \cdot \bar{B} \cdot C_i + \bar{A} \cdot B \cdot \bar{C}_i + A \cdot \bar{B} \cdot \bar{C}_i + A \cdot B \cdot C_i$ $\ddot{U} = A \cdot B + A \cdot C_i + B \cdot C_i$ </p> <p style="margin-top: 10px;"> Aufbau:  </p> <p style="margin-top: 10px;"> Schaltsymbol:  </p>	A	B	C _i	Z	Ü	0	0	0	0	0	0	0	1	1	0	0	1	0	1	0	0	1	1	0	1	1	0	0	1	0	1	0	1	0	1	1	1	0	0	1	1	1	1	1	1
A	B	Z	Ü																																																															
0	0	0	0																																																															
0	1	1	0																																																															
1	0	1	0																																																															
1	1	1	1																																																															
A	B	C _i	Z	Ü																																																														
0	0	0	0	0																																																														
0	0	1	1	0																																																														
0	1	0	1	0																																																														
0	1	1	0	1																																																														
1	0	0	1	0																																																														
1	0	1	0	1																																																														
1	1	0	0	1																																																														
1	1	1	1	1																																																														

Mehrbit-Addierer:

- Serieller Addierer: wie von Hand mit Schieberegister, langsam, wenig Gatter (1 VA, 1 FF)
- Paralleler Addierer:
 - Aus der Wahrheitstabelle in Normalform: mind. 3 Gatterlaufzeiten, schnell, wird rasch aufwendig zum Herstellen
 - Ripple-Carry Addierer: für jede Stelle, ausser dem niedrigwertigsten, wird ein Volladdierer gebraucht (für 4 Bit: 3 VA & 1 HA); Addierzeit wächst linear mit Stellenanzahl
 - Carry-Look-Ahead Addierer (fast carry): Überträge werden parallel zur Summenbildung berechnet, mind. 2 Gatterlaufzeiten; Zwitter zwischen den beiden obengenannten



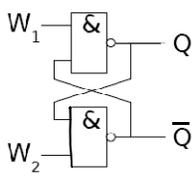
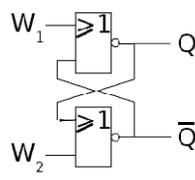
Bsp: Ripple-Carry Addierer: hier nur Volladdierer

Multiplizierer: Booth-Multiplizierer nicht vergessen!!!

- i. Bitweise Multiplikation der Partialprodukte
- ii. Aufsummieren der verschobenen Partialprodukte
→ Shift & Add

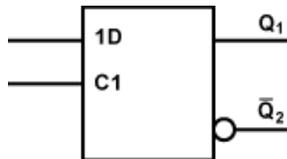
8. Sequentielle Schaltungen: Latches, Flip-Flops

- Anders als bei kombinatorischen Schaltungen hängen die Ausgangswerte nun auch von vorangegangenen Werten ab: Rückkoppelung (feedback loops)

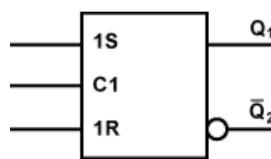
<p>NAND-Latch</p>  <p><i>Wahrheitstabelle</i></p> <table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <thead> <tr> <th>$W_1(R)$</th> <th>$W_2(S)$</th> <th>Q_{n+1}</th> <th>\overline{Q}_{n+1}</th> <th>Funktion</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>- (1)</td> <td>- (1)</td> <td>Verboten</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>0</td> <td>Setzen</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>1</td> <td>Rücksetzen</td> </tr> <tr> <td>1</td> <td>1</td> <td>Q_n</td> <td>\overline{Q}_n</td> <td>Speichern</td> </tr> </tbody> </table> <p>$W_1 = W_2 = 0$ nicht zulässig</p> <p><i>Charakteristische Gleichung:</i> $Q_{n+1} = W_2 \cdot Q_n + \overline{W}_1$</p>	$W_1(R)$	$W_2(S)$	Q_{n+1}	\overline{Q}_{n+1}	Funktion	0	0	- (1)	- (1)	Verboten	0	1	1	0	Setzen	1	0	0	1	Rücksetzen	1	1	Q_n	\overline{Q}_n	Speichern	<p>NOR-Latch</p>  <p><i>Wahrheitstabelle</i></p> <table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <thead> <tr> <th>$W_1(R)$</th> <th>$W_2(S)$</th> <th>Q_{n+1}</th> <th>\overline{Q}_{n+1}</th> <th>Funktion</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>Q_n</td> <td>\overline{Q}_n</td> <td>Speichern</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>0</td> <td>Setzen</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>1</td> <td>Rücksetzen</td> </tr> <tr> <td>1</td> <td>1</td> <td>- (0)</td> <td>- (0)</td> <td>Verboten</td> </tr> </tbody> </table> <p>$W_1 = W_2 = 1$ nicht zulässig</p> <p><i>Charakteristische Gleichung:</i> $Q_{n+1} = \overline{W}_1 \cdot Q_n + W_2$</p>	$W_1(R)$	$W_2(S)$	Q_{n+1}	\overline{Q}_{n+1}	Funktion	0	0	Q_n	\overline{Q}_n	Speichern	0	1	1	0	Setzen	1	0	0	1	Rücksetzen	1	1	- (0)	- (0)	Verboten
$W_1(R)$	$W_2(S)$	Q_{n+1}	\overline{Q}_{n+1}	Funktion																																															
0	0	- (1)	- (1)	Verboten																																															
0	1	1	0	Setzen																																															
1	0	0	1	Rücksetzen																																															
1	1	Q_n	\overline{Q}_n	Speichern																																															
$W_1(R)$	$W_2(S)$	Q_{n+1}	\overline{Q}_{n+1}	Funktion																																															
0	0	Q_n	\overline{Q}_n	Speichern																																															
0	1	1	0	Setzen																																															
1	0	0	1	Rücksetzen																																															
1	1	- (0)	- (0)	Verboten																																															

(Takt-)Zustandsgesteuerte Latches:

D-Latch:



SR-Latch

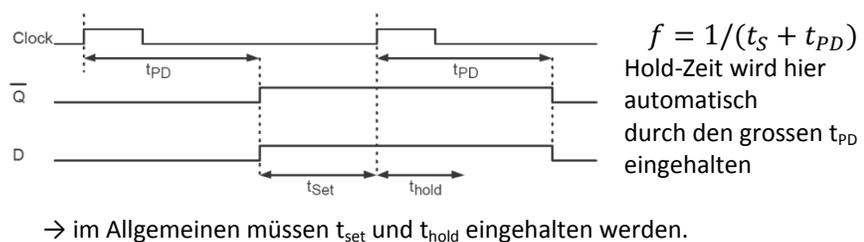
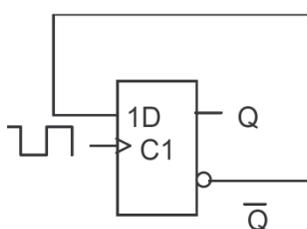


- Latches sind während der ganzen aktiven Taktphase transparent → empfindlich gegenüber Störungen
- Ein rückgekoppeltes D-Latch ist ein swinging latch, während ein rückgekoppeltes D-FF ein Takthalbierer ist

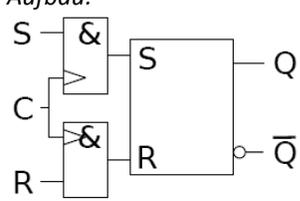
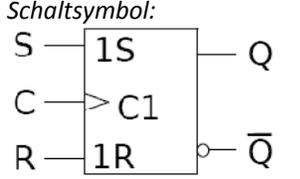
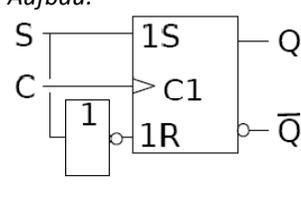
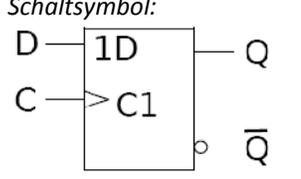
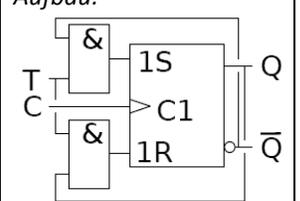
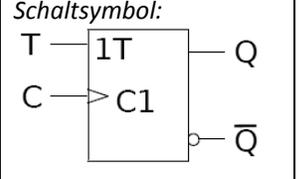
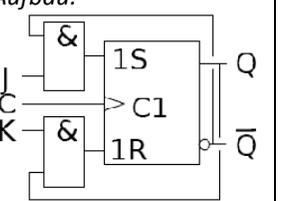
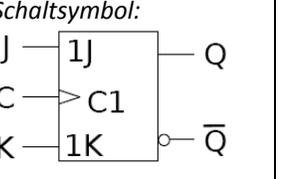
Dynamik von FF, Latch:

- t_p : Propagation Delay: Zeit vom Eingang bis Erscheinen am Ausgang
- t_s : Setup-Zeit: Wie lange muss das Datensignal vor der aktiven Taktflanke anliegen?
- t_h : Hold-Zeit: Wie lange muss das Datensignal nach der aktiven Taktflanke anliegen?

Bsp (Übg 5):

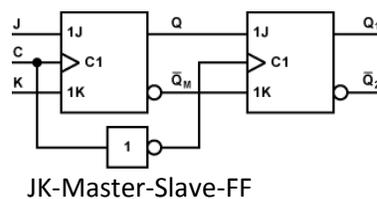
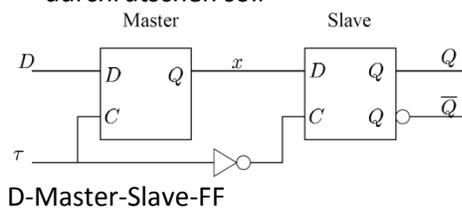


- **Flip-Flops:** flankengesteuerte Kippschaltungen, der Eingangszustand wird nur zum Zeitpunkt des Taktwechsels (Flanke) wirksam. Dazwischen bleibt der Zustand gespeichert

<p>SR-FF (Set-Reset)</p> <p><i>Aufbau:</i></p>  <p><i>Schaltsymbol:</i></p>  <p>S: set R: reset S=R=1 unzulässig</p> $Q_{n+1} = S + \bar{R} \cdot Q_n$ <p><i>Wahrheitstabelle:</i></p> <table border="1" style="width:100%; text-align: center;"> <tr><th>S</th><th>R</th><th>Q_{n+1}</th></tr> <tr><td>0</td><td>0</td><td>Q_n</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>irregulär</td></tr> </table> <p><i>Ausführliche W.Tabelle:</i></p> <table border="1" style="width:100%; text-align: center;"> <tr><th>S</th><th>R</th><th>Q_n</th><th>Q_{n+1}</th></tr> <tr><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>X</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>X</td></tr> </table>	S	R	Q_{n+1}	0	0	Q_n	0	1	0	1	0	1	1	1	irregulär	S	R	Q_n	Q_{n+1}	0	0	0	0	0	0	1	1	0	1	0	0	0	1	1	0	1	0	0	1	1	0	1	1	1	1	0	X	1	1	1	X	<p>D-FF (Delay)</p> <p><i>Aufbau:</i></p>  <p><i>Schaltsymbol:</i></p>  <p>Zustand von D wird bei Flanke übernommen</p> $Q_{n+1} = D_n$ <p><i>Wahrheitstabelle:</i></p> <table border="1" style="width:100%; text-align: center;"> <tr><th>D</th><th>Q_{n+1}</th></tr> <tr><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td></tr> </table> <p><i>Ausführliche W.Tabelle:</i></p> <table border="1" style="width:100%; text-align: center;"> <tr><th>D</th><th>Q_n</th><th>Q_{n+1}</th></tr> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </table>	D	Q_{n+1}	0	0	1	1	D	Q_n	Q_{n+1}	0	0	0	0	1	0	1	0	1	1	1	1	<p>T-FF (Toggle)</p> <p><i>Aufbau:</i></p>  <p><i>Schaltsymbol:</i></p>  <p>Für T=1 kippt der Zustand von Q</p> $Q_{n+1} = \bar{Q}_n \cdot T + Q_n \cdot \bar{T}$ $= Q_n \oplus T$ <p><i>Wahrheitstabelle:</i></p> <table border="1" style="width:100%; text-align: center;"> <tr><th>T</th><th>Q_{n+1}</th></tr> <tr><td>0</td><td>Q_n</td></tr> <tr><td>1</td><td>\bar{Q}_n</td></tr> </table> <p><i>Ausführliche W.Tabelle:</i></p> <table border="1" style="width:100%; text-align: center;"> <tr><th>T</th><th>Q_n</th><th>Q_{n+1}</th></tr> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </table>	T	Q_{n+1}	0	Q_n	1	\bar{Q}_n	T	Q_n	Q_{n+1}	0	0	0	0	1	1	1	0	1	1	1	0	<p>JK-FF (Jump-Kill)</p> <p><i>Aufbau:</i></p>  <p><i>Schaltsymbol:</i></p>  <p>J: set K: reset J=K: wie T-FF</p> $Q_{n+1} = J \cdot \bar{Q}_n + \bar{K} \cdot Q_n$ <p><i>Wahrheitstabelle:</i></p> <table border="1" style="width:100%; text-align: center;"> <tr><th>J</th><th>K</th><th>Q_{n+1}</th></tr> <tr><td>0</td><td>0</td><td>Q_n</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>\bar{Q}_n</td></tr> </table> <p><i>Ausführliche W.Tabelle:</i></p> <table border="1" style="width:100%; text-align: center;"> <tr><th>J</th><th>K</th><th>Q_n</th><th>Q_{n+1}</th></tr> <tr><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>0</td></tr> </table>	J	K	Q_{n+1}	0	0	Q_n	0	1	0	1	0	1	1	1	\bar{Q}_n	J	K	Q_n	Q_{n+1}	0	0	0	0	0	0	1	1	0	1	0	0	0	1	1	0	1	0	0	1	1	0	1	1	1	1	0	1	1	1	1	0
S	R	Q_{n+1}																																																																																																																																																	
0	0	Q_n																																																																																																																																																	
0	1	0																																																																																																																																																	
1	0	1																																																																																																																																																	
1	1	irregulär																																																																																																																																																	
S	R	Q_n	Q_{n+1}																																																																																																																																																
0	0	0	0																																																																																																																																																
0	0	1	1																																																																																																																																																
0	1	0	0																																																																																																																																																
0	1	1	0																																																																																																																																																
1	0	0	1																																																																																																																																																
1	0	1	1																																																																																																																																																
1	1	0	X																																																																																																																																																
1	1	1	X																																																																																																																																																
D	Q_{n+1}																																																																																																																																																		
0	0																																																																																																																																																		
1	1																																																																																																																																																		
D	Q_n	Q_{n+1}																																																																																																																																																	
0	0	0																																																																																																																																																	
0	1	0																																																																																																																																																	
1	0	1																																																																																																																																																	
1	1	1																																																																																																																																																	
T	Q_{n+1}																																																																																																																																																		
0	Q_n																																																																																																																																																		
1	\bar{Q}_n																																																																																																																																																		
T	Q_n	Q_{n+1}																																																																																																																																																	
0	0	0																																																																																																																																																	
0	1	1																																																																																																																																																	
1	0	1																																																																																																																																																	
1	1	0																																																																																																																																																	
J	K	Q_{n+1}																																																																																																																																																	
0	0	Q_n																																																																																																																																																	
0	1	0																																																																																																																																																	
1	0	1																																																																																																																																																	
1	1	\bar{Q}_n																																																																																																																																																	
J	K	Q_n	Q_{n+1}																																																																																																																																																
0	0	0	0																																																																																																																																																
0	0	1	1																																																																																																																																																
0	1	0	0																																																																																																																																																
0	1	1	0																																																																																																																																																
1	0	0	1																																																																																																																																																
1	0	1	1																																																																																																																																																
1	1	0	1																																																																																																																																																
1	1	1	0																																																																																																																																																

Master-Slave:

- Information am Eingang soll nicht direkt nach einer Taktflanke gespeichert werden, sondern erst mit der Rückflanke am Ausgang erscheinen → gleiches Muster, Slave um 1 Takt hinten drin
- Dies ist notwendig, wenn in einer Serienschaltung mit synchronem Takt die Information nicht durchrutschen soll



9. Zahlen, Codes:

- Festkommazahl (ohne Exponent) ↔ Fließkommazahl (mit Exponent)

Dualzahlen:

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}
128	64	32	16	8	4	2	1	0.5	0.25	0.125	0.0625

Umrechnung Festkommazahlen → Dezimalzahl:

$$D = \sum_{i=-\infty}^{\infty} b_i \cdot r^i \quad (b_i \text{ Koeffizienten und } r \text{ Basis})$$

oder mittels "Horner-Schema"

- $3564_{(8)} = 3 \cdot 8^3 + 5 \cdot 8^2 + 6 \cdot 8^1 + 4 \cdot 8^0 = 1908_{(10)}$
- $11.101_{(2)} = 1 \cdot 2^1 + 1 \cdot 2^0 + 1 \cdot 2^{-1} + 1 \cdot 2^{-3} = 3.625_{(10)}$
- $A3D_{(16)} = 10 \cdot 16^2 + 3 \cdot 16^1 + 13 \cdot 16^0 = 2621_{(10)}$

Päckchenweises Umrechnen Dual ↔ Oktal ↔ Hexadezimal:

i. $E \quad 6 \quad 0 \quad 5_{(16)} = 58'885_{(10)}$

$$1 \mid 110 \mid 011 \mid 0 \mid 00 \mid 00 \mid 0 \mid 101_{(2)}$$

$$1 \quad 6 \quad 3 \quad 0 \quad 0 \quad 5_{(8)}$$

ii. $0. \quad 2 \quad C_{(16)} = 0.171875_{(10)}$

$$0. \mid 001 \mid 0 \mid 11 \mid 00_{(2)}$$

$$0. \quad 1 \quad 3 \quad 0_{(8)}$$

- Bei Dualzahlen mit Stellen hinter Komma sind die Gruppen vom Komma aus nach links und rechts zu bilden
- Nicht jede rationale Dezimalzahl lässt sich exakt in einem anderen Zahlensystem darstellbar → abbrechen

Dual 1 Bit	Oktal 3 Dualstellen	Hexadezimal 4 Dualstellen (Tetrade)	Dezimal
0000	00	0	0
0001	01	1	1
0010	02	2	2
0011	03	3	3
0100	04	4	4
0101	05	5	5
0110	06	6	6
0111	07	7	7
1000	10	8	8
1001	11	9	9
1010	12	A	10
1011	13	B	11
1100	14	C	12
1101	15	D	13
1110	16	E	14
1111	17	F	15

Dezimalsystem → Dual / Oktal / Hexadezimal:

i. Ganze Zahlen (Division mit Zahlenbasis):	ii. $0 \leq x \leq 1$ (Multiplikation mit Zahlenbasis)
$50 : 2 = 25 \text{ (R: 0) (LSB)}$ $25 : 2 = 12 \text{ (R: 1)}$ $12 : 2 = 06 \text{ (R: 0)}$ $06 : 2 = 03 \text{ (R: 0)}$ $03 : 2 = 01 \text{ (R: 1)}$ $01 : 2 = 00 \text{ (R: 1) (MSB)}$ $\rightarrow 50_{(10)} = 110010_{(2)}$	$0.171875 \cdot 16 = 2.75 \text{ (2) (MSB)}$ $0.750000 \cdot 16 = 12 \text{ (C) (LSB)}$ $\rightarrow 0.171875_{(10)} = 0.2C_{(16)}$

Fließkommazahlen:

$$x = M^i \cdot B^E \quad (M: \text{Mantisse, } B: \text{Basis, } E: \text{Exponent})$$

31	30	22	0
VZ	Exponent	Mantisse	

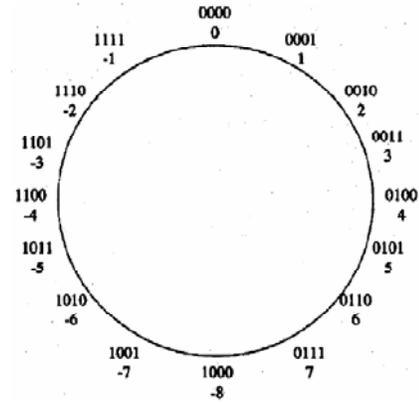
- VZ: „0“ gleich positiv, „1“ gleich negativ
 Exponent: 8 Bit, Basis 2, beginnend mit -127
 Mantisse: 23 Bit, 1.xxx, wobei nur xxx gespeichert, evtl. mit ,0' auffüllen
- $+53.2265625_{(10)} = +110101.001101_{(2)} = (1).10101001101_{(2)} \cdot 2^5$
- VZ: 0 (,+)
 Exponent: 10000100 (+127+5)
 Mantisse: 1010100110100000000000 (23 Bit)

2er Komplement:

Dualzahl allgemein:

1 Bit	(n-1) Bit
VZ	Betrag

2er-Komplement	VZ	Betrag
positiv	0	MSB, ..., LSB = $ X_{(2)} $
negativ	1	i. $ X_{(2)} $ bitweise invertieren ii. „1 ₍₂₎ “ dazuzählen



Wertebereich: $-2^{(n-1)} \leq D \leq 2^{(n-1)} - 1$ (n: Anzahl Bits)

2er Komplement → Dezimalzahl:

$D_{10} = -b_n \cdot 2^n + \sum_{i=0}^{n-1} b_i \cdot 2^i$ Bsp: $1011_{(2)} = -2^3 + 2^1 + 2^0 = -5$

Hinweise:

- i. Der Zahlenwert ist durch die vorgegebene Stellenzahl bestimmt → führende Nullen nicht weglassen
- ii. Das 2er Komplement eines 2er Komplements ergibt wieder die ursprüngliche Zahl

Dezimalzahl → 2er Komplement:

1. Darstellen der vorzeichenlosen Zahl als Dualzahl
2. Bitweises Invertieren
3. Zur Inversion eine „1₍₂₎“ an der Stelle des LSB dazuzaddieren

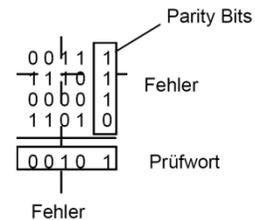
Bsp: $-5.4375_{(10)} \rightarrow \text{????} . \text{????}_{(2)}$ (4 Stellen vor und nach Komma)

1. $5_{(10)} = 0101_{(2)}$; $0.4375_{(10)} = 0.0111_{(2)} \rightarrow 5.4375_{(10)} = 0101.0111_{(2)}$
2. Invertiert: $1010.1000_{(2)}$
3. 1₍₂₎ addieren: $1010.1001_{(2)}$

- Subtraktion mit 2er Komplement (A-B) = Addition von A mit 2er Komplement von B (Übertrag geg. ignorieren)

Fehlerkorrektur:

- Odd parity: Kode wird zur Ungeradzahligkeit ergänzt
- Even parity: Kode wird zur Geradzahligkeit ergänzt



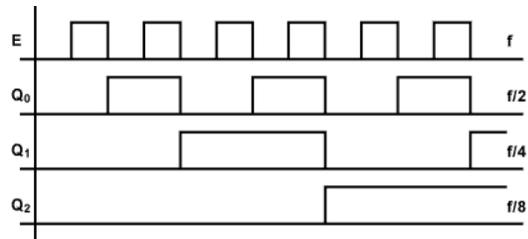
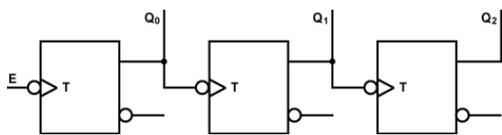
Übersicht Tetraden-Codes (4 Binärstellen)

Binär	BCD 8-4-2-1	Excess-3 Symm.	Aiken Symm.	4-2-2-1 A/D- Wandler	Gray einschrittig	O'Brien einschrittig
0000	0		0	0	0	
0001	1		1	1	1	
0010	2		2	2	3	0
0011	3	0	3	3	2	
0100	4	1	4		7	4
0101	5	2			6	3
0110	6	3		4	4	1
0111	7	4		5	5	2
1000	8	5			15	
1001	9	6			14	
1010		7			12	9
1011		8	5		13	
1100		9	6	6	8	5
1101			7	7	9	6
1110			8	8	11	8
1111			9	9	10	7

10. Zähler und Frequenzteiler

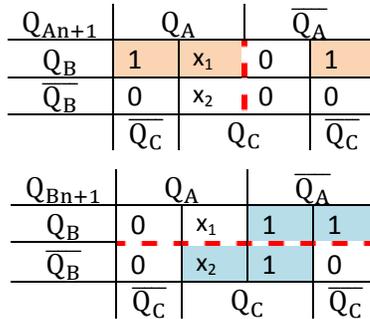
Asynchronzähler	Synchronzähler
i. Grundbaustein ist T-FF ii. Reihenschaltung mit T-FF iii. Frequenzteiler: $f_t = \frac{f_E}{2^n}$, für n T-FF iv. Wegen t_p ist das Abfangen der Signalen von entscheidender Bedeutung v. Maximale Taktfrequenz ist limitiert durch $f_{max} = \frac{1}{\sum_{i=1}^n t_{pi}}$, d.h. langsam für grosse Bit-Dualzähler (höchstens 2.8 MHz für 12-Bit-Dualzähler mit $t_p = 30ns$)	Eigenschaften: <ul style="list-style-type: none"> - Der Takt liegt an allen FF gleichzeitig an - Ein Verknüpfungsnetz zum selektiven Schalten der einzelnen FF notwendig - JK/D/SR-FF ersetzen T-FF, da vielseitiger Synthese: <ul style="list-style-type: none"> i. Ablaufabelle erstellen, t_{n+1} & t_n bestimmen ii. Jede Binärstelle mittels KV-Diagramm vereinfachen (aufpassen bei JK-FF) iii. Koeff.Vergleich mit char. Gleichung der FF iv. Folgezustände aus 'don't care' überprüfen

Jeder Dualzähler teilt die Frequenz in Stufen von Zweierpotenzen: hier 3-Bit



Bsp: zyklischer 3 Bit Pseudo-Gray-Code-Generator mit JK-FF:

Nr	t_n			t_{n+1}		
	Q_{An}	Q_{Bn}	Q_{Cn}	Q_{An+1}	Q_{Bn+1}	Q_{Cn+1}
1	0	0	0	0	0	1
2	0	0	1	0	1	1
3	0	1	1	0	1	0
4	0	1	0	1	1	0
5	1	1	0	1	0	0
6	1	0	0	0	0	0
7	1	1	1	x_1	x_1	x_1
8	1	0	1	x_2	x_2	x_2



Achtung bei JK-FF: Grenzen beachten!

Char.Gleichung für JK-FF (farbig):

$$Q_{i(n+1)} = (J_i \cdot \overline{Q_i} + \overline{K_i} \cdot Q_i)_n \rightarrow J_i \text{ \& } K_i \text{ bestimmen}$$

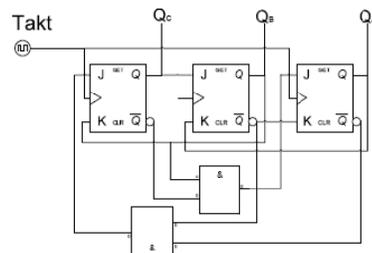
$$Q_{An+1} = Q_B \cdot \overline{Q_C} \cdot \overline{Q_A} + Q_B \cdot Q_A \rightarrow J_A = Q_B \cdot \overline{Q_C}, K_A = \overline{Q_B}$$

$$Q_{Bn+1} = Q_C \cdot \overline{Q_B} + \overline{Q_A} \cdot Q_B \rightarrow J_B = Q_C, K_B = Q_A$$

$$Q_{Cn+1} = \overline{Q_A} \cdot \overline{Q_B} \cdot \overline{Q_C} + \overline{Q_B} \cdot Q_C \rightarrow J_C = \overline{Q_A} \cdot \overline{Q_B}, K_C = Q_B$$

Kontrolle

- Zustand nach 111: $J_A = 0, K_A = 0$ (speichern), $J_B = 1, K_B = 1(T)$, $J_C = 0, K_C = 1(K)$ ist folglich 100
- Zustand nach 101: $J_A = 0, K_A = 1(K)$, $J_B = 1, K_B = 1(T)$, $J_C = 0, K_C = 0$ (speichern) ist folglich 011



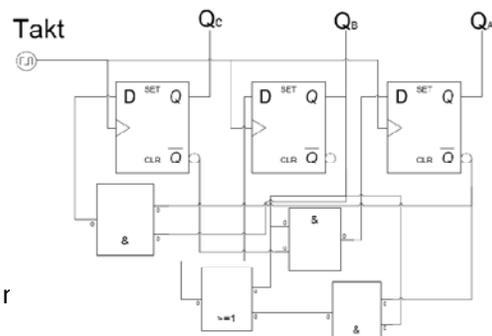
Char.Gleichung für D-FF:

$$Q_{i(n+1)} = D_n \rightarrow \text{keine Beschränkungen mehr}$$

$$Q_{An+1} = Q_B \cdot \overline{Q_C}$$

$$Q_{Bn+1} = \overline{Q_C} + \overline{Q_A} \cdot Q_B$$

$$Q_{Cn+1} = \overline{Q_A} \cdot \overline{Q_B}$$



11. Automaten

- Reagiert auf Eingang
- Generiert Ausgang
- Vom Eingangssignal und momentanem Zustand eindeutig abhängig.

- $X = (x_1, x_2, \dots, x_e)$: Eingabealphabet mit e Eingangszuständen x_i
- $Y = (y_1, y_2, \dots, y_b)$: Ausgabealphabet mit b Ausgangszuständen y_i
- $Z = (z_1, z_2, \dots, z_m)$: Zustandsmenge mit m Zuständen z_i (in FF gespeichert)
- $g: (x_i, z_j) \rightarrow z_k$: Zustandsübergangsfunktion, Übergangsfunktion
- $f: (x_i, z_j) \rightarrow y_r$: Ausgangsfunktion

Mealy	Moore	Medwedjew
<p>- synchron, meist D-FF mit positiver Flanke - Master-Slave FF auch möglich - #FF = $\lfloor \log_2 m \rfloor + 1$</p> <p>Zustandsgraph: </p>	<p>- Ausgang Y_n nur von den inneren Speicherzuständen abhängig</p> <p>Zustandsgraph: </p> <p>∃! möglicher Ausgang pro Zustand!</p>	<p>- Ausgang ist mit den Speicherinhalten identisch, keine Ausgangsfunktion</p> <p>Zustandsgraph: </p>

Beschreibungsmöglichkeiten von Automaten:

- Folgezustandstabelle: zu jeder Kombination der internen Zustände und des Eingangsvektors werden die Folgezustände und Ausgangswerte aufgelistet (nur für endliche Automaten FSM möglich)
- Zustandsgraph: äquivalent zur Folgezustandstabelle, bestehen aus Knoten und Kanten

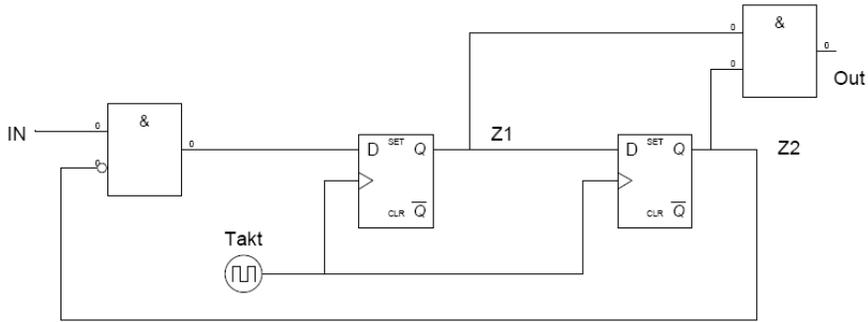
Analyse von Schaltwerken:

- Automatentyp bestimmen
- Schaltfunktionen $f(x_i, z_j)$ und $g(x_i, z_j)$ bestimmen
- Folgezustandstabelle aufstellen
- Zustandsgraph (-diagramm) bestimmen

Synthese von Schaltwerken:

- Automatentyp bestimmen
- Zustandsmenge, Anzahl Zustandsvariablen und Speicherglieder (Mealy \leq Moore) bestimmen
- Definition der Ein- und Ausgangsvariablen, Zustandskodierung wählen
- Darstellung im Zustandsdiagramm
- Aufstellen der Zustandsfolgetabelle
- Bestimmen und Minimieren der Übergangs- und Ausgangsfunktion (KV-Diagramm)
- Prüfung auf unbenutzte Zustände (Hängebleiben)
- Schaltplan konstruieren

Bsp Analyse:



i. Moore-Automat

ii. $X = (IN), Y = (OUT), Z = (z_1, z_2),$

$f(z_1, z_2) = z_1 \cdot z_2 = OUT, g(IN, z_1, z_2):$

$Z_{1n+1} = \overline{Z_{2n}} \cdot IN, Z_{2n+1} = Z_{1n}$

iii. Folgezustandstabelle:

No	IN	Z1 _n	Z2 _n	Code	Z1 _{n+1}	Z2 _{n+1}	Code	OUT _n
1	0	0	0	A	0	0	A	0
2	0	0	1	B	0	0	A	0
3	0	1	0	C	0	1	B	0
4	0	1	1	D	0	1	B	1
5	1	0	0	A	1	0	C	0
6	1	0	1	B	0	0	A	0
7	1	1	0	C	1	1	D	0
8	1	1	1	D	0	1	B	1

iv. Zustandsdiagramm:

