

PVK HS2020

Josephine Loehle
jloehle@student.ethz.ch



Organisatorisches

- Über mich: Josephine, 3. Semester Elektrotechnik
- Webseite: <https://www.n.ethz.ch/student/jloehle/>
 - Folien, Zusammenfassung, Notizen
- Email: jloehle@student.ethz.ch

Ablauf

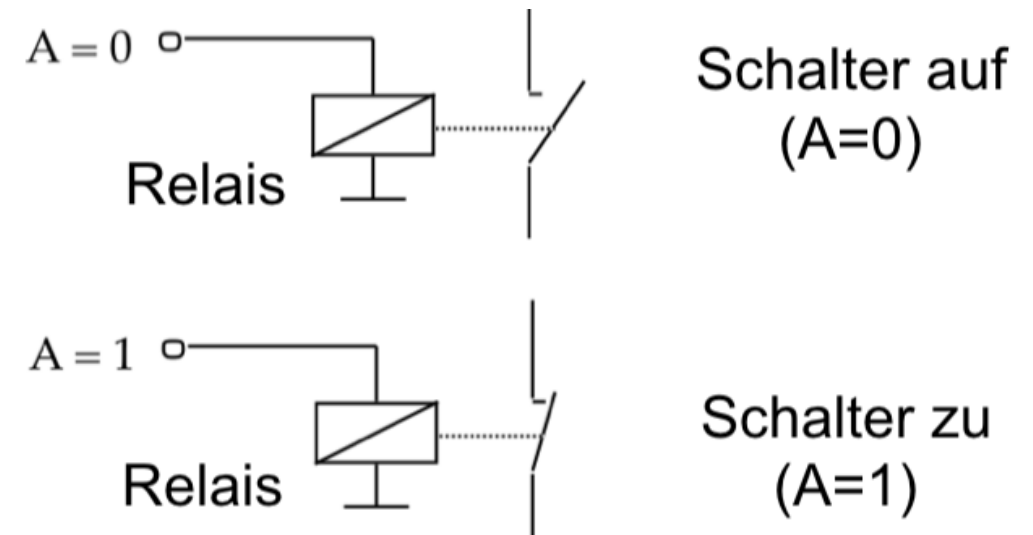
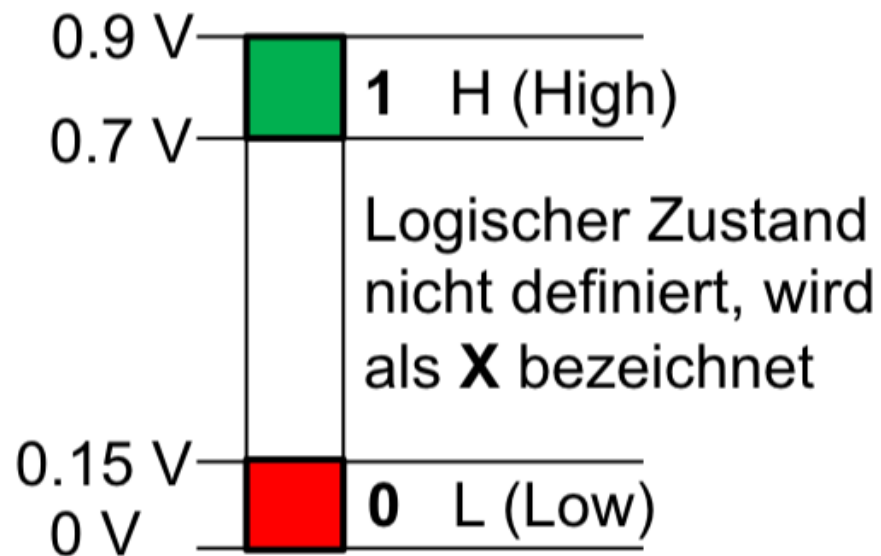
- 9:00-10:30: Theorie 1
- Kurze Pause
- 10:40-12:00: Theorie 2
- Mittagspause
- 13:00-16:00: Prüfung lösen + Zeit für Fragen

Logische Verknüpfungen

Zweiwertige Logik – Bits

Bits

- 2 binäre Zustände: 0 & 1
- n Bits können 2^n Zustände beschreiben
- MSB ... LSB



Wahrheitstabellen

- N Eingänge
- M Ausgänge

2^N Zeilen

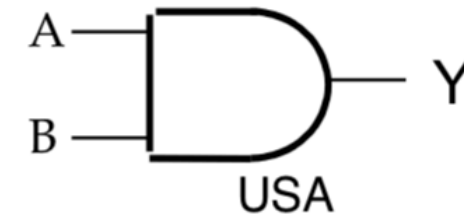
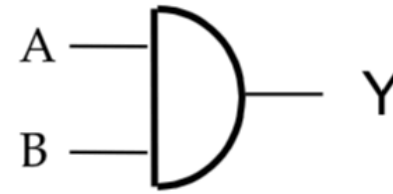
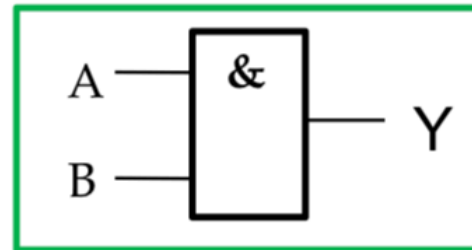
A	B	C	Y
0	0	0	$f(0,0,0) \in \{0,1\}$
0	0	1	$f(0,0,1) \in \{0,1\}$
0	1	0	$f(0,1,0) \in \{0,1\}$
0	1	1	$f(0,1,1) \in \{0,1\}$
1	0	0	$f(1,0,0) \in \{0,1\}$
1	0	1	$f(1,0,1) \in \{0,1\}$
1	1	0	$f(1,1,0) \in \{0,1\}$
1	1	1	$f(1,1,1) \in \{0,1\}$

M + N Spalten

UND-Verknüpfung

- $Y = 1$, wenn $A = 1$ & $B = 1$

Hier verwendet



A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

$$Y = A \wedge B$$

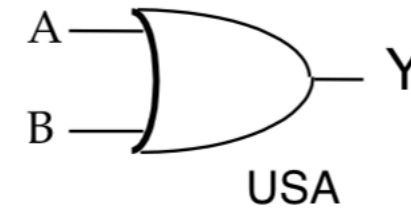
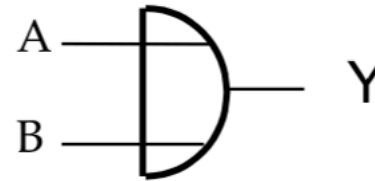
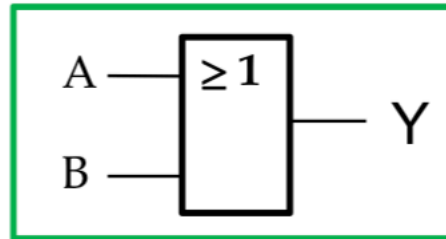
$$Y = A \cdot B$$

Hier verwendet

ODER-Verknüpfung

- $Y = 1$, wenn $A = 1$ &/oder $B = 1$

Hier verwendet



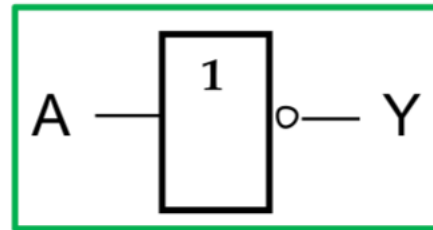
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

$$Y = A \vee B$$

$$Y = A + B$$

INVERTER

- $Y = 1$, wenn $A = 0$



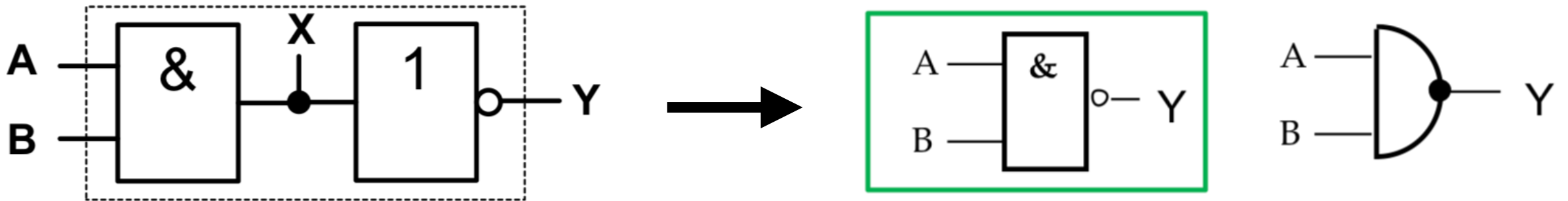
A	Y
0	1
1	0

Hier verwendet

$$Y = \bar{A}$$

NAND-Gatter

- $Y = 1$, wenn $A = 0$ &/oder $B = 0$



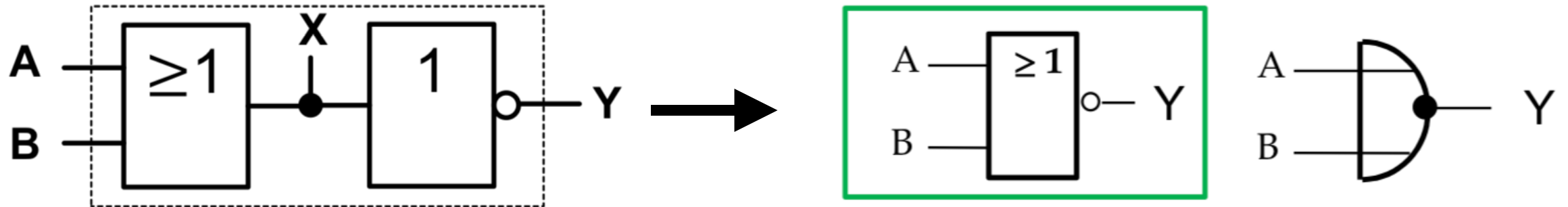
A	B	X	Y
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

$$Y = \overline{A \wedge B} \qquad Y = \overline{A \cdot B}$$

Hier verwendet

NOR-Gatter

- $Y = 1$, wenn $A = 0$ & $B = 0$



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

$$Y = \overline{A \vee B}$$

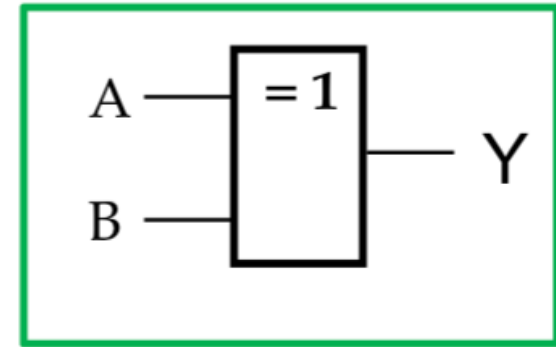
$$Y = \overline{A + B}$$

XOR-Gatter (= Exclusive Or)

- $Y = 1$, wenn $A = 0$ oder $B = 0$

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

Hier verwendet



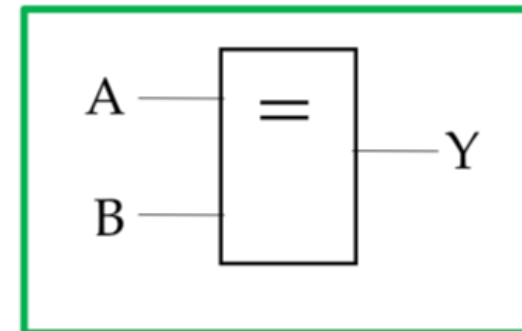
Hier verwendet

$$Y = A \oplus B$$

XNOR-Gatter

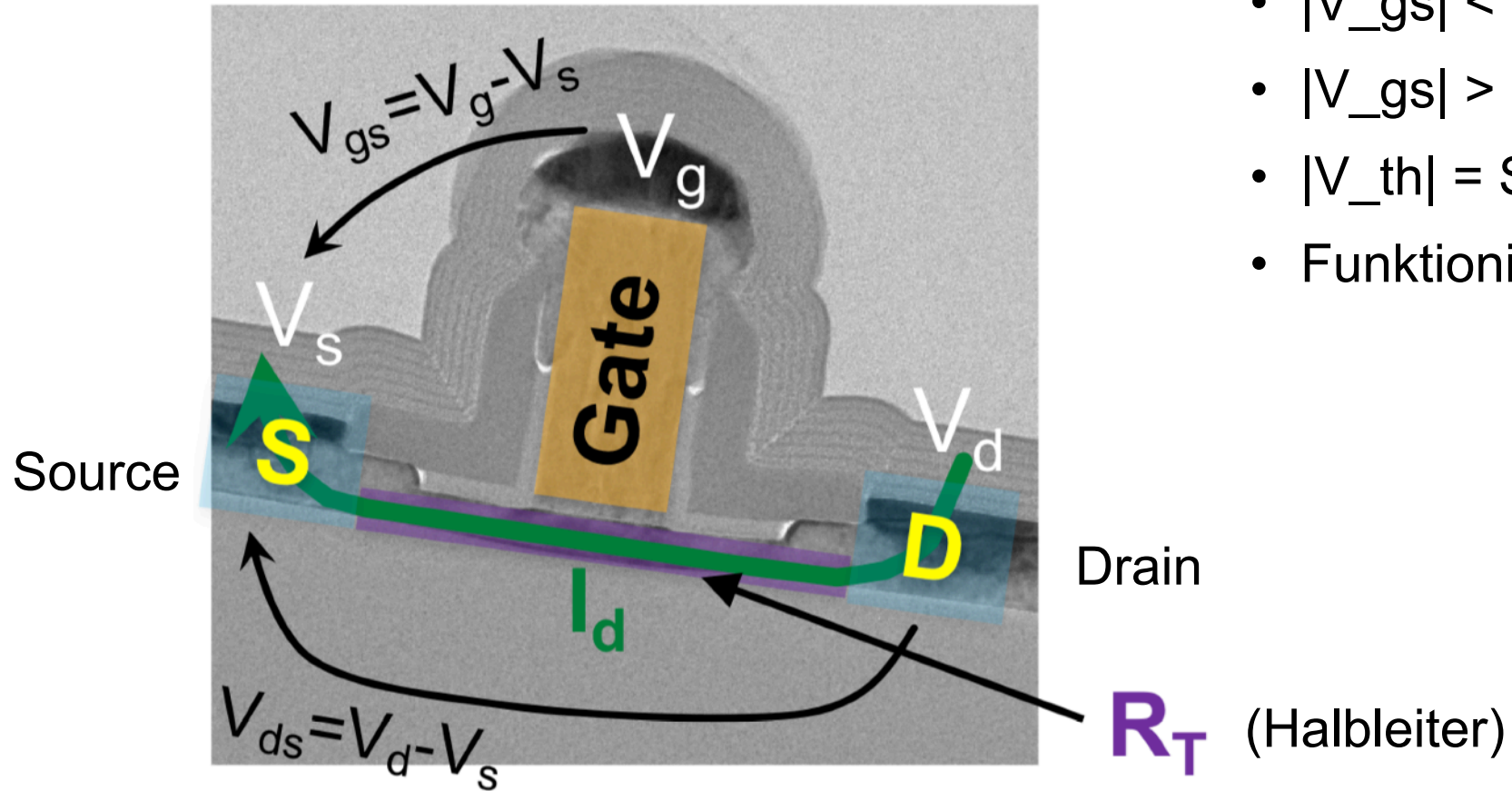
- Invertiertes XOR-Gatter

$$Y = \overline{A \oplus B}$$



CMOS Schaltungen

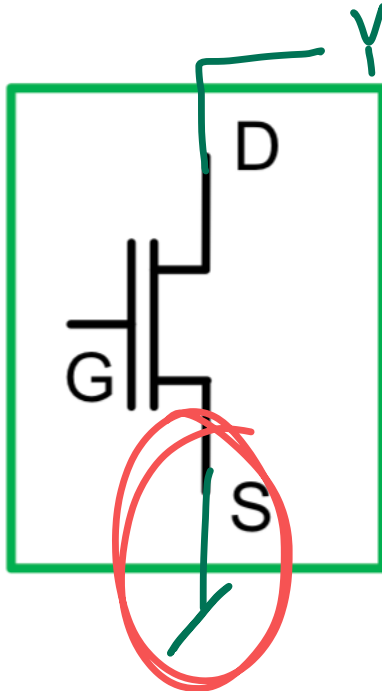
MOS-Transistoren



- $|V_{gs}| < |V_{th}|$ kein Strom fließt
- $|V_{gs}| > |V_{th}|$ es fließt Strom
- $|V_{th}|$ = Schwellenspannung (ca. 0.8V)
- Funktioniert durch Halbleiter

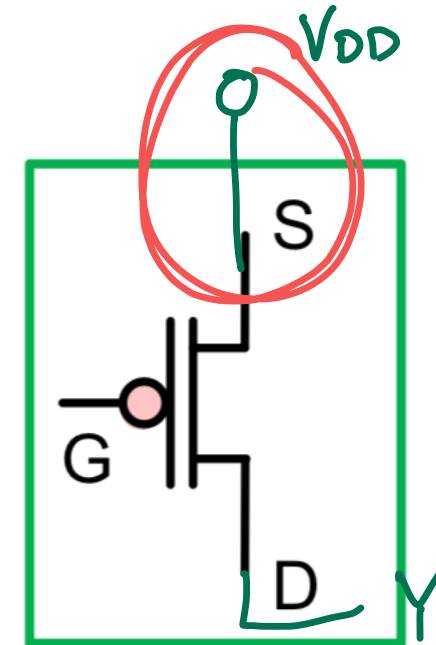
NMOS

- Elektronen fließen von S zu D
- P-Typ Halbleiter
- Für Pull-down Schaltung verwendet



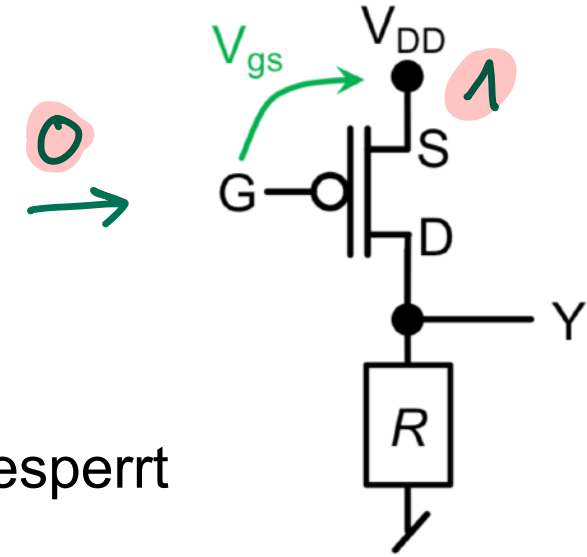
PMOS

- Löcher fließen von S zu D
- N-Typ Halbleiter
- Für Pull-up Schaltung verwendet



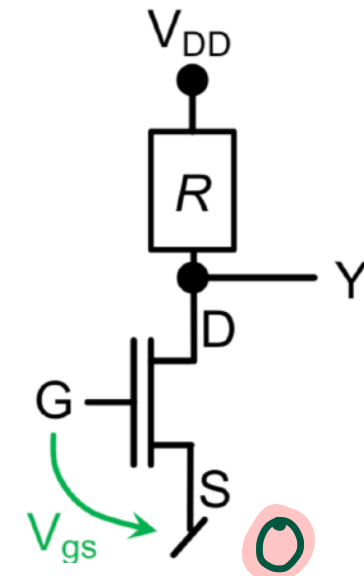
Pull-up Schaltung

- PMOS-Transistoren
- Leitet, wenn $G \neq S$ ($G = 0$)
- Leitet eine 1 weiter (an V_{DD} gebunden)
- NN (Unbestimmter Zustand) wenn Transistor davor gesperrt



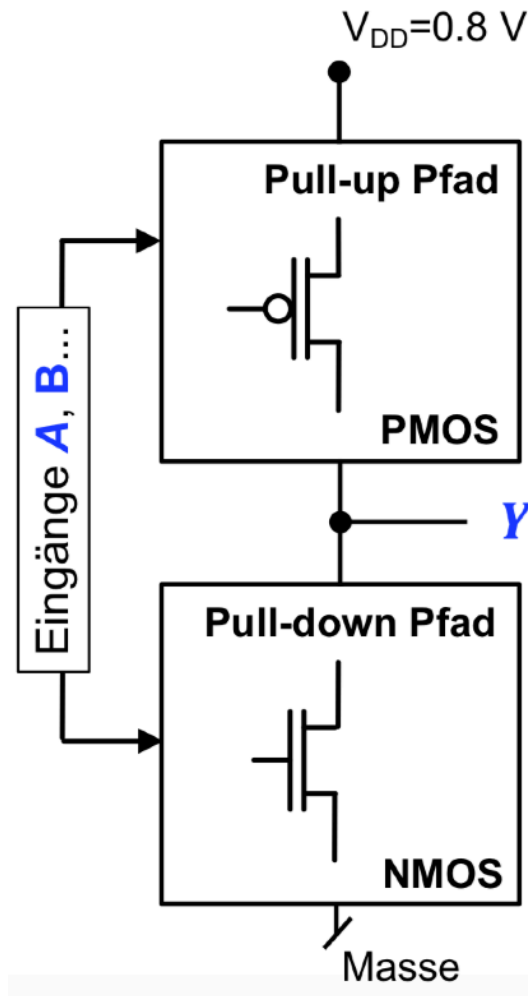
Pull-down Schaltung

- NMOS-Transistoren
- Leitet, wenn $G \neq S$ ($G = 1$)
- Leitet eine 0 weiter (an Masse gebunden)
- NN (Unbestimmter Zustand) wenn Transistor davor gesperrt

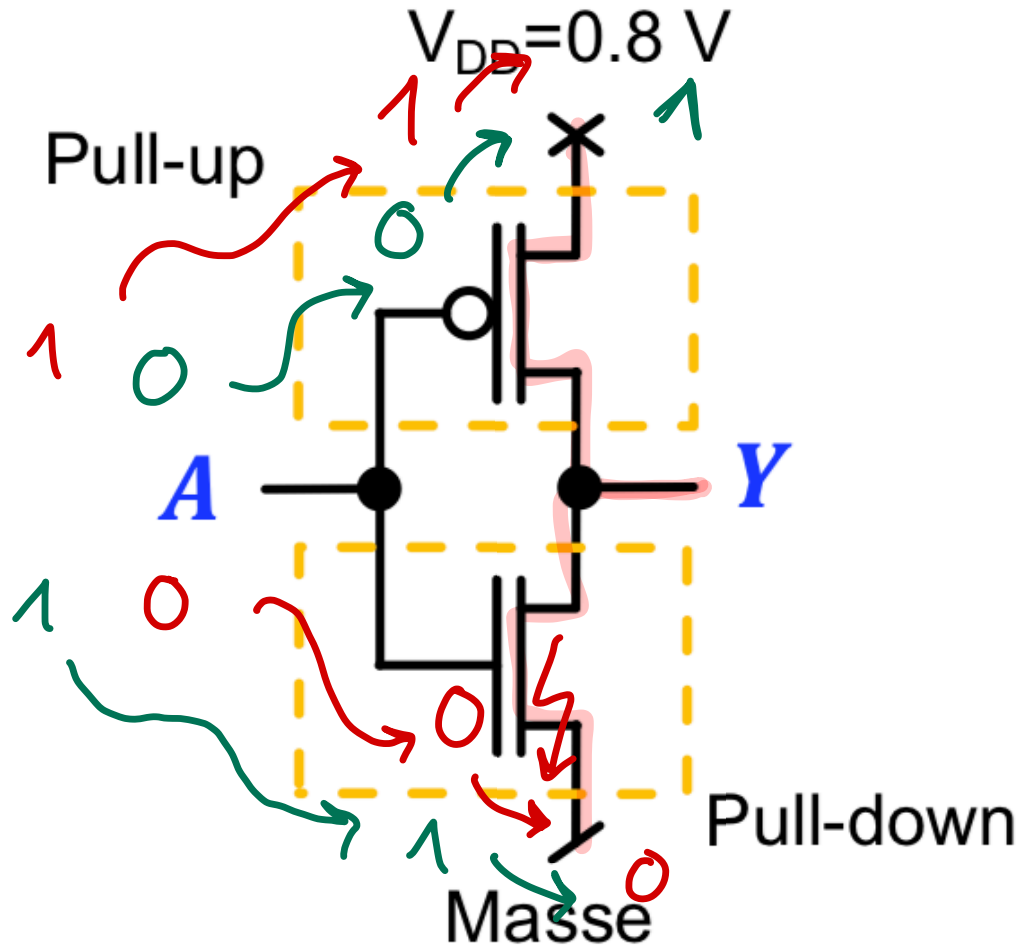


Schaltungen

- # NMOS-Transistoren = # PMOS-Transistoren = # Eingänge

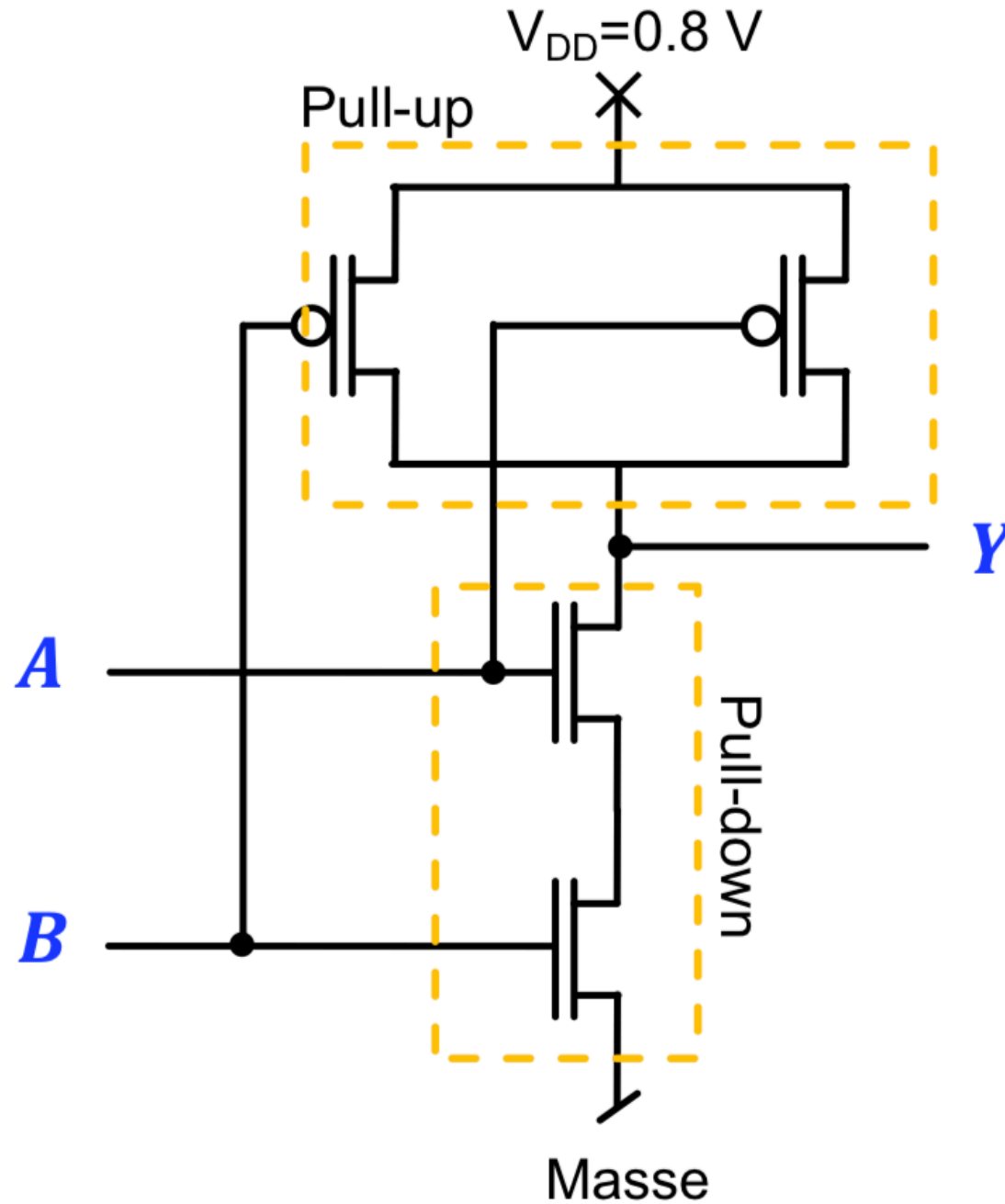


NOT



A	PMOS	NMOS	Y
0	ZU	OFFEN	1
1	OFFEN	ZU	0

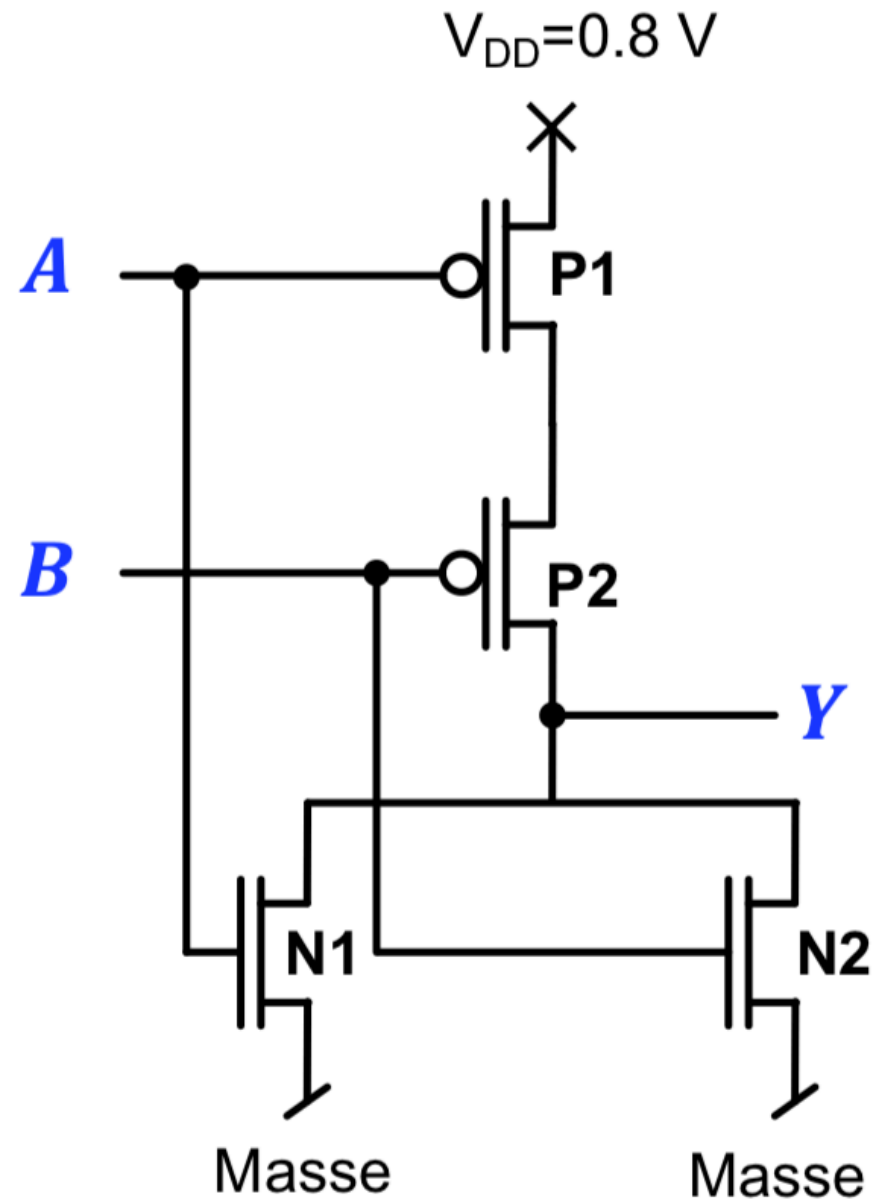
NAND



- PMOS parallel
- NMOS in Serie

0	0		1
0	1		1
1	0		1
1	1		0

NOR



- PMOS in Serie
- NMOS parallel

0	0		1
0	1		0
1	0		0
1	1		0

Pfade umwandeln

1. Gleichung aufstellen

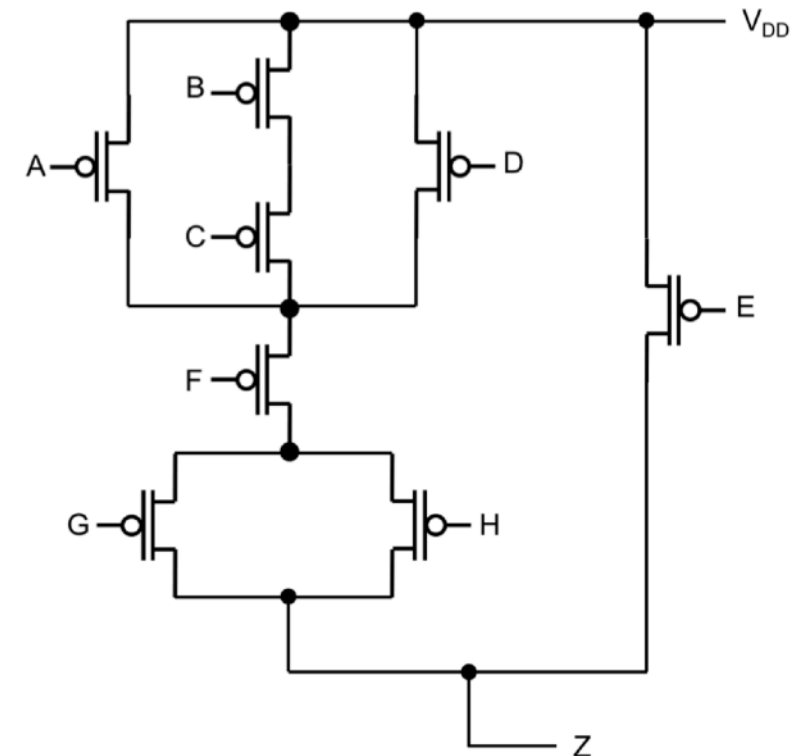
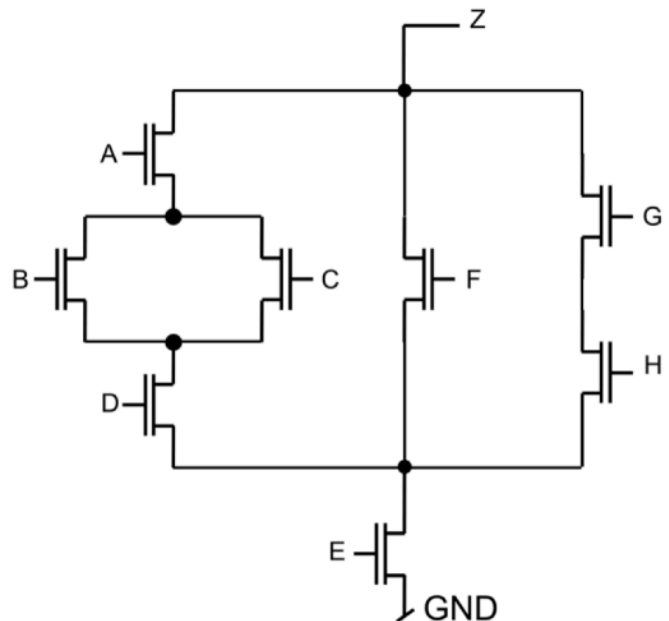
- (Bei Pull-up einzelne Elemente invertiert, bei Pull down ganze Gleichung)
- (UND in Serie, ODER parallel)

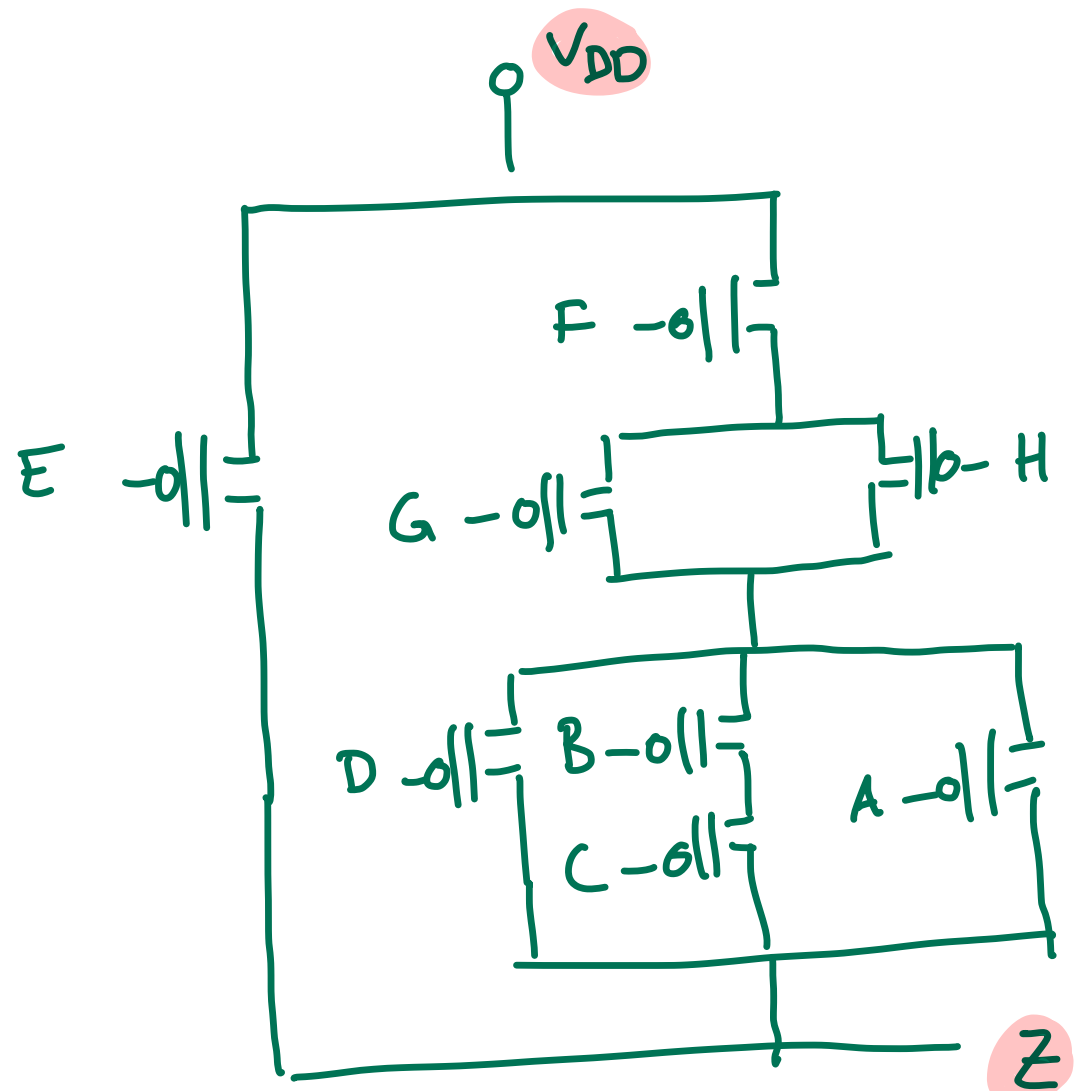
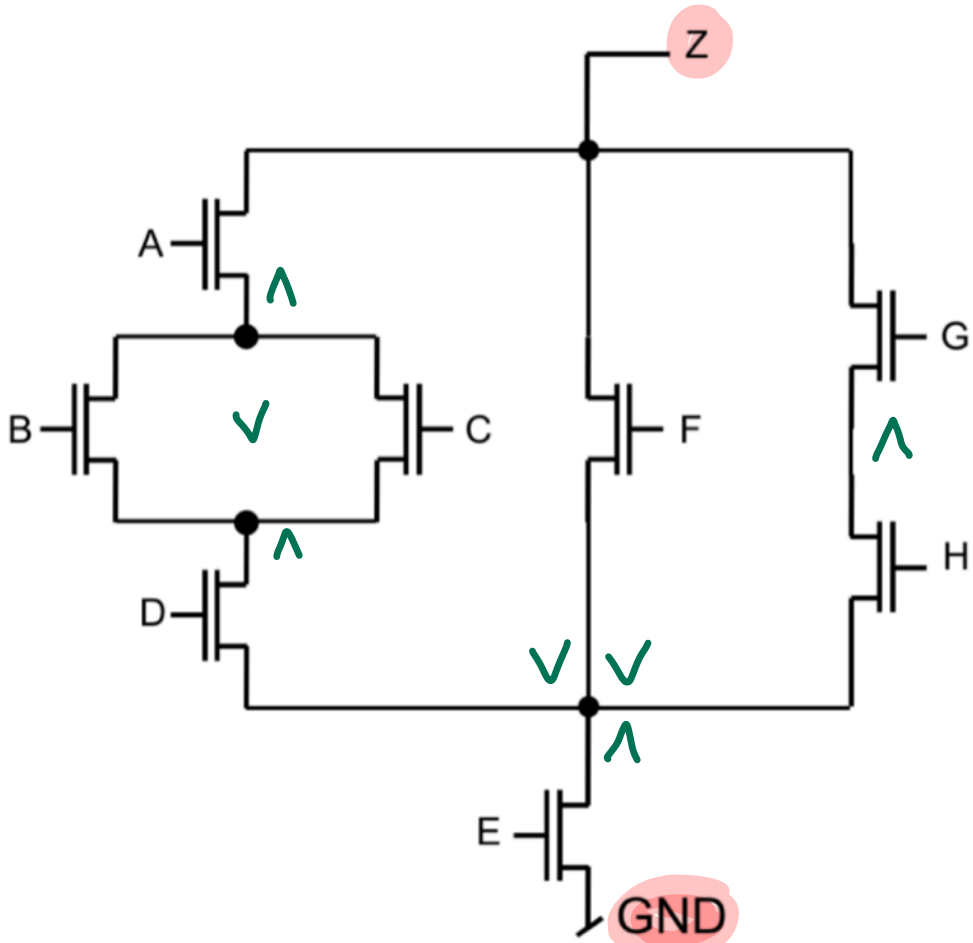
2. Gleichung invertieren

- (UND/ODER vertauschen und Investitionen vertauschen)

3. Neue Gleichung zeichnen

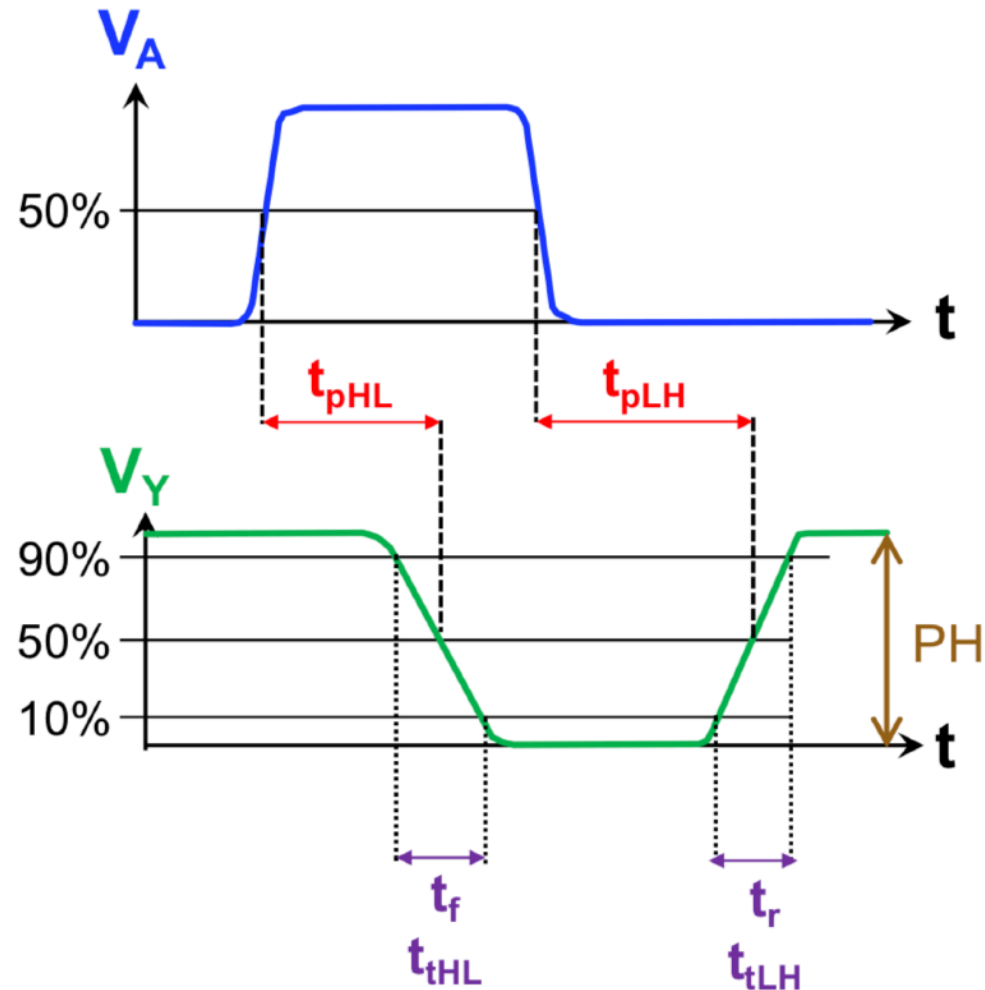
- (UND: Serie, ODER: Parallel)





$$E \wedge (F \vee (G \wedge H) \vee (D \wedge (B \vee C) \wedge A)) \rightarrow \bar{E} \vee (\bar{F} \wedge (\bar{G} \vee \bar{H}) \wedge (\bar{D} \vee (\bar{B} \wedge \bar{C}) \vee \bar{A}));$$

Zeitverzögerung



$$t_d = (t_{pHL} + t_{pLH})/2$$

- Verzögerungszeit, gemessen bei 50%
- “Verschiebung im Block”
- Anstiegs- Abfallszeit, gemessen bei 10%-90%
- “Schräge Übergänge”

Zeitverzögerung

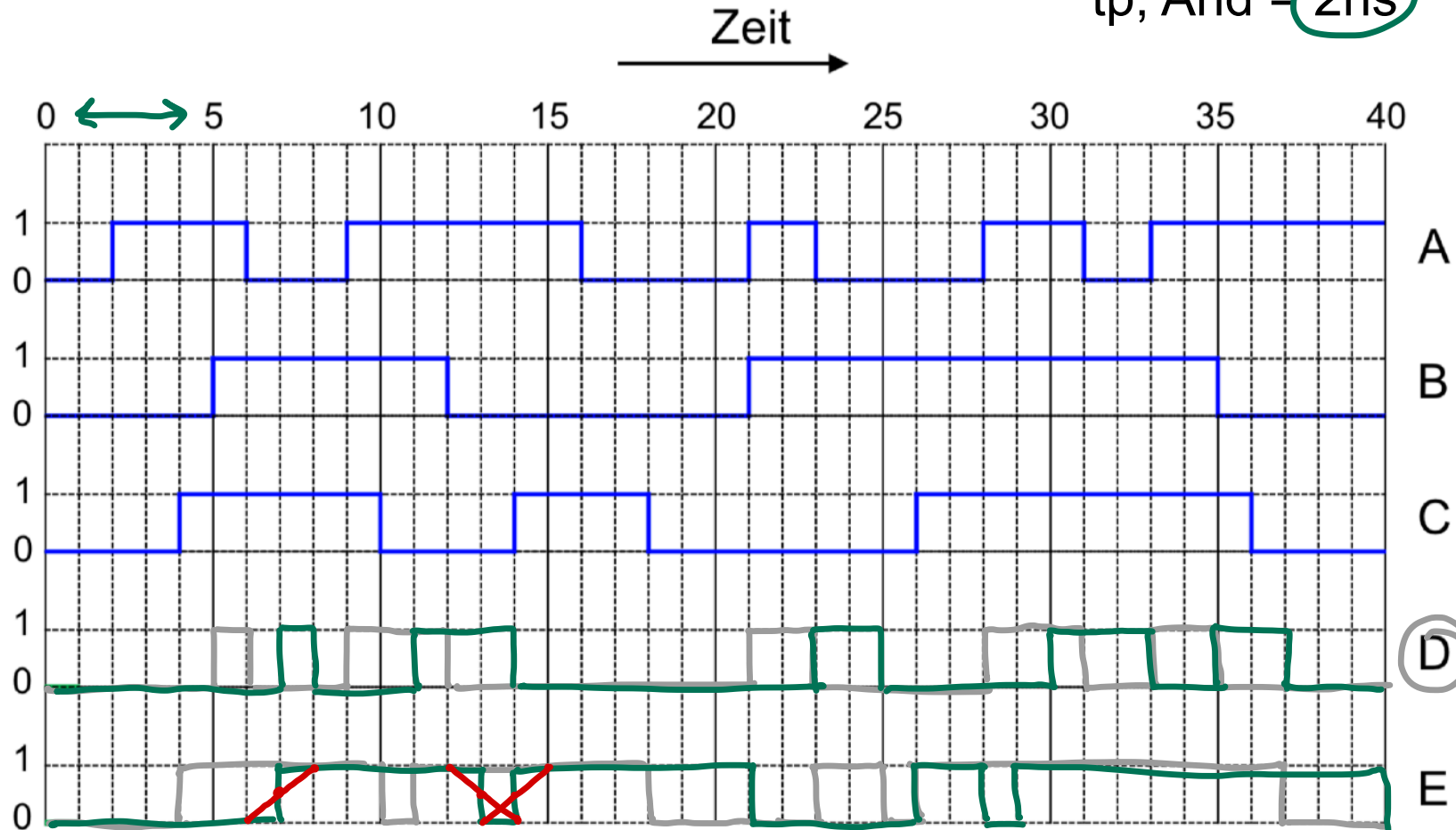
$$t_f = 2ns = t_r$$

$$D = A * B$$

$$t_{p, \text{And}} = 2ns$$

$$E = D + C$$

$$t_{p, \text{OR}} = 3ns$$



Schritt 1: Signal ohne Zeitverzögerung einzeichnen

Bool'sche Algebra

Basisregeln Schaltalgebra

- Kommutativität: $(A + B + C = B + C + A)$ $(A * B * C = B * A * C)$
- Assoziativität: $(A + (B + C) = (A + B) + C)$ $(A * (B * C) = (A * B) * C)$
- Distributivität: $(A * B + A * C = A * (B + C))$ $((A + B) * (A + C) = A + (B * C))$

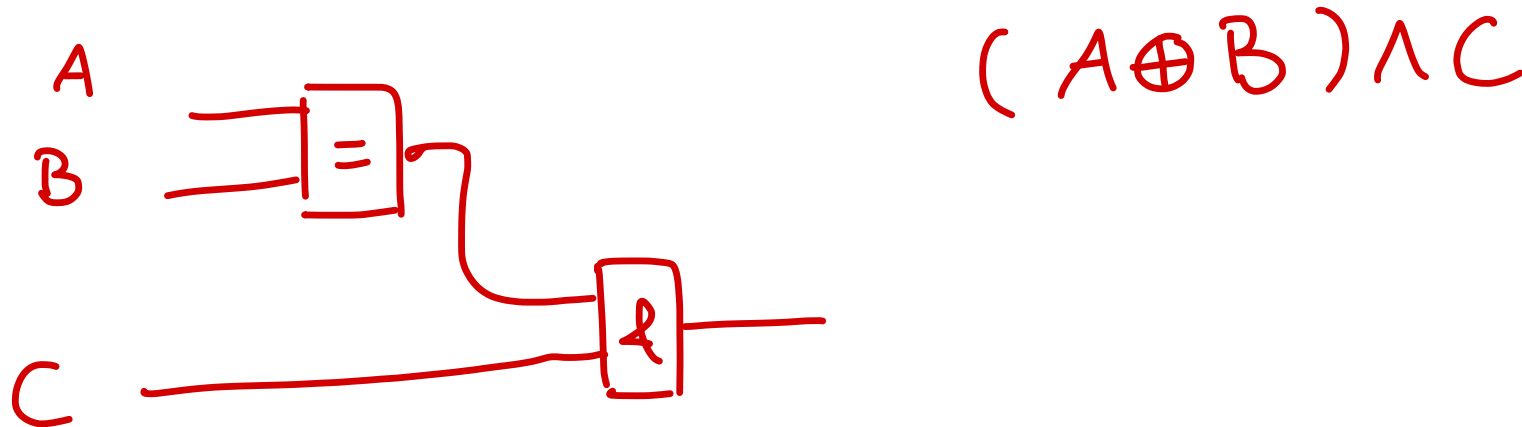
- “Normale” Regeln für Multiplikation und Addition

Bool'sche Grundregeln

NICHT	$!!A = A$	-
NULL	$A + 0 = A$	$A * 0 = 0$
EINS	$A + 1 = 1$	$A * 1 = A$
IDEMPOTENZ	$A + A = A$	$A * A = A$
KOMPLEMENT	$A + !A = 1$	$A * !A = 0$
ADSORPTION	$A + (!A * B) = A + B$	$A * (!A + B) = A * B$
ABSORPTION	$A + (A * B) = A$	$A * (A + B) = A$
NACHBARSCHAFT	$(A * B) + (!A * B) = B$	$(A + B) * (!A + B) = B$

Vorrangsregeln

- Klammern
- {AND; OR; NOR; NAND} vor {XOR; XNOR}
- {AND; OR; NOR; NAND} und {XOR; XNOR} sind untereinander gleichwertig



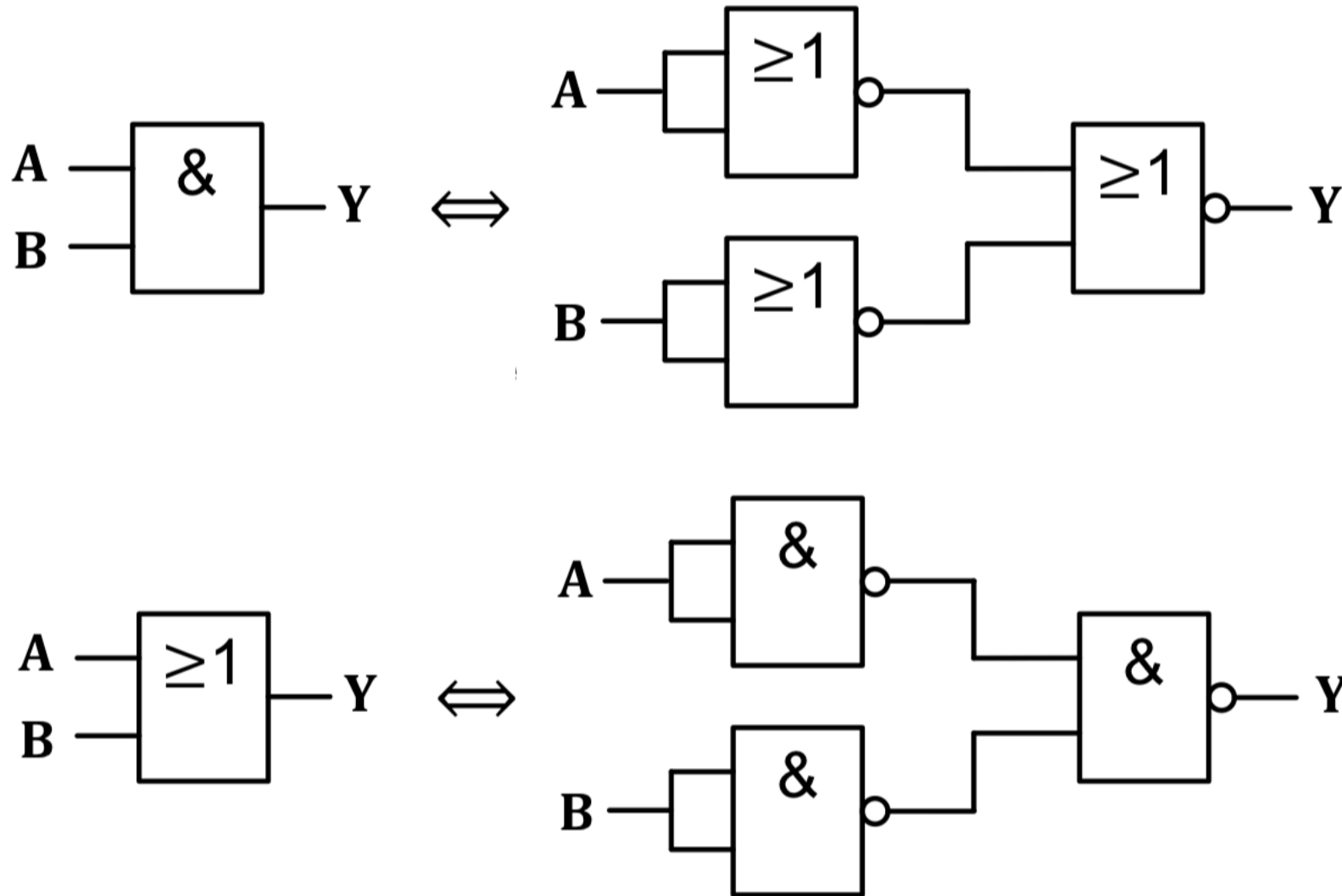
De Morgan'sche Regeln

Die De Morgan'schen Gesetze sind verallgemeinerbar auf mehreren Variablen:

Erstes Gesetz: $Y = \overline{A \wedge B \wedge C \wedge \dots} = \bar{A} \vee \bar{B} \vee \bar{C} \vee \dots$

Zweites Gesetz: $Y = \overline{A \vee B \vee C \vee \dots} = \bar{A} \wedge \bar{B} \wedge \bar{C} \wedge \dots$

De Morgan'sche Regeln



Schaltungssynthese

Minterm

- UND-Verknüpfung von Schaltvariablen
- Minterm gibt nur bei einer Variablenkombination 1 (Minimum)
- Bei n Variablen 2^n Minterme

- Bildung und Variablenfindung:
 - Invertierte = 0
 - Nicht-invertierte = 1

Maxterm

- ODER-Verknüpfung von Schaltvariablen
- Maxterm gibt nur bei einer Variablenkombination 0 (Maximum)
- Bei n Variablen 2^n Maxterme

- Bildung und Variablenfindung:
 - Invertierte = 1
 - Nicht-invertierte = 0

Normalformen

Disjunktive Normalform

- ODER-Verknüpfung von allen **Mintermen = 1**
-

Konjunktive Normalform

- UND-Verknüpfung von allen Maxtermen = 0
-

Kanonische Form

- Nur Min- oder Maxterme mit jeder Variable genau 1 Mal

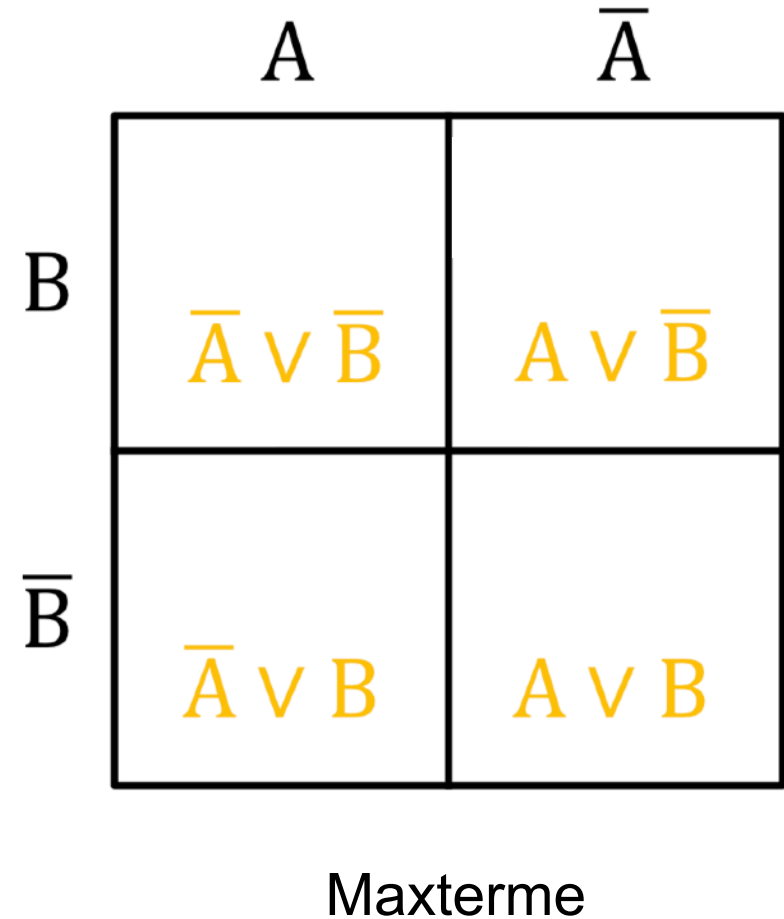
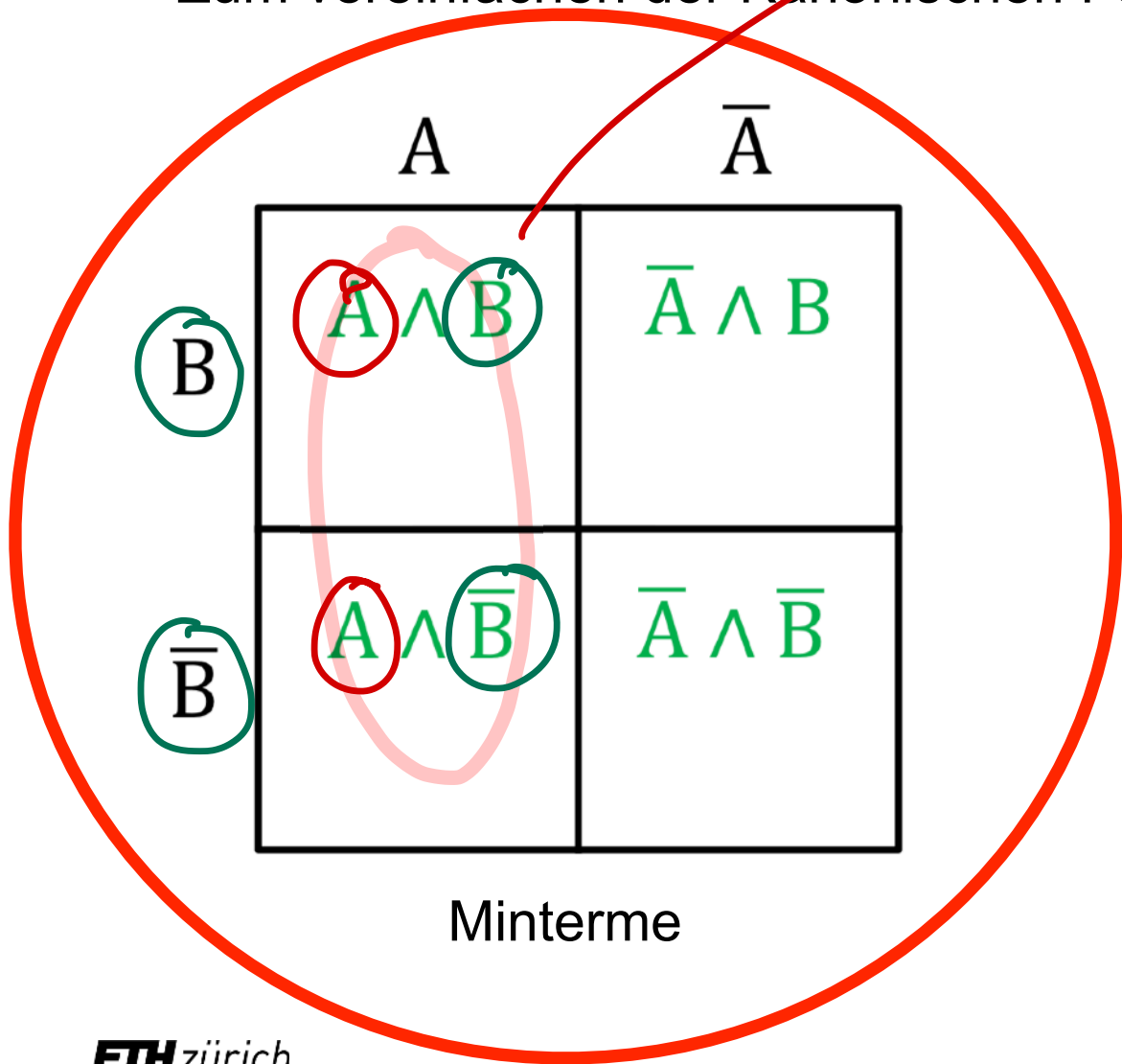
A	B	C	Y = f(A, B, C)	Minterm	Maxterm
0	0	0	1	$\bar{A} \wedge \bar{B} \wedge \bar{C}$	
0	0	1	1	$\bar{A} \wedge \bar{B} \wedge C$	
0	1	0	0		$A \vee \bar{B} \vee C$
0	1	1	1	$\bar{A} \wedge B \wedge C$	
1	0	0	1	$A \wedge \bar{B} \wedge \bar{C}$	
1	0	1	1	$A \wedge \bar{B} \wedge C$	
1	1	0	0		$\bar{A} \vee \bar{B} \vee C$
1	1	1	0		$\bar{A} \vee \bar{B} \vee \bar{C}$

$$\text{DNF: } (\bar{A} \wedge \bar{B} \wedge \bar{C}) \vee (\bar{A} \wedge \bar{B} \wedge C) \vee (\bar{A} \wedge B \wedge C) \vee (A \wedge \bar{B} \wedge \bar{C}) \vee (A \wedge \bar{B} \wedge C)$$

$$\text{KNF: } (A \vee \bar{B} \vee C) \wedge (\bar{A} \vee \bar{B} \vee C) \wedge (\bar{A} \vee \bar{B} \vee \bar{C})$$

Karnaugh Diagramme

- Zum vereinfachen der Kanonischen Formen



Vereinfachen mit dem Karnaugh Diagramm (DNF)

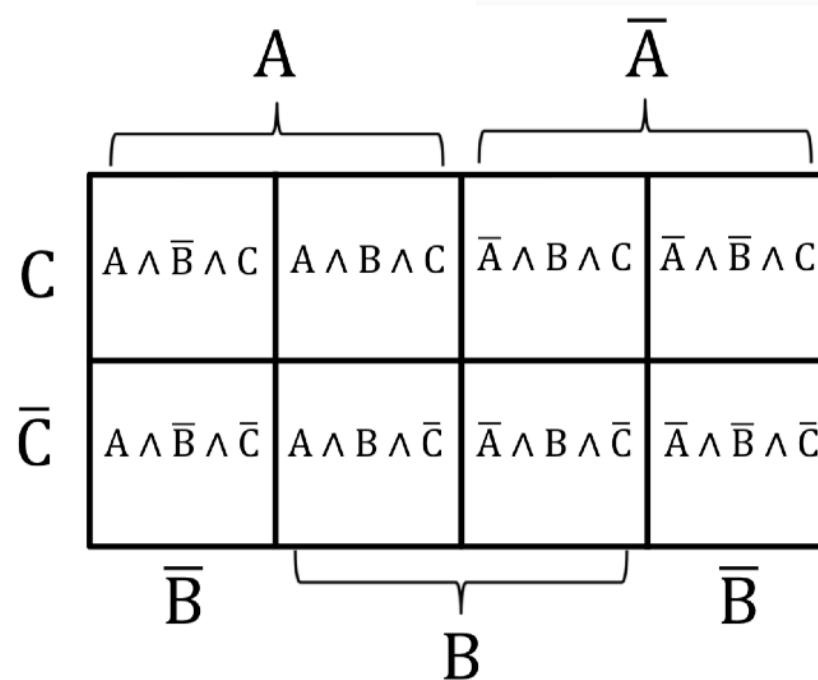
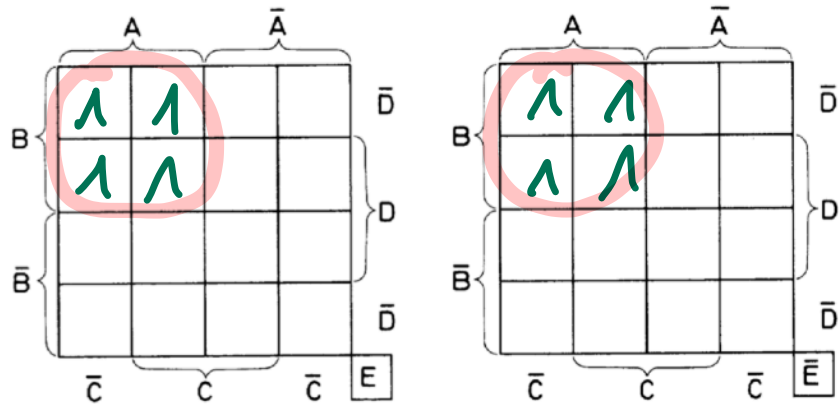
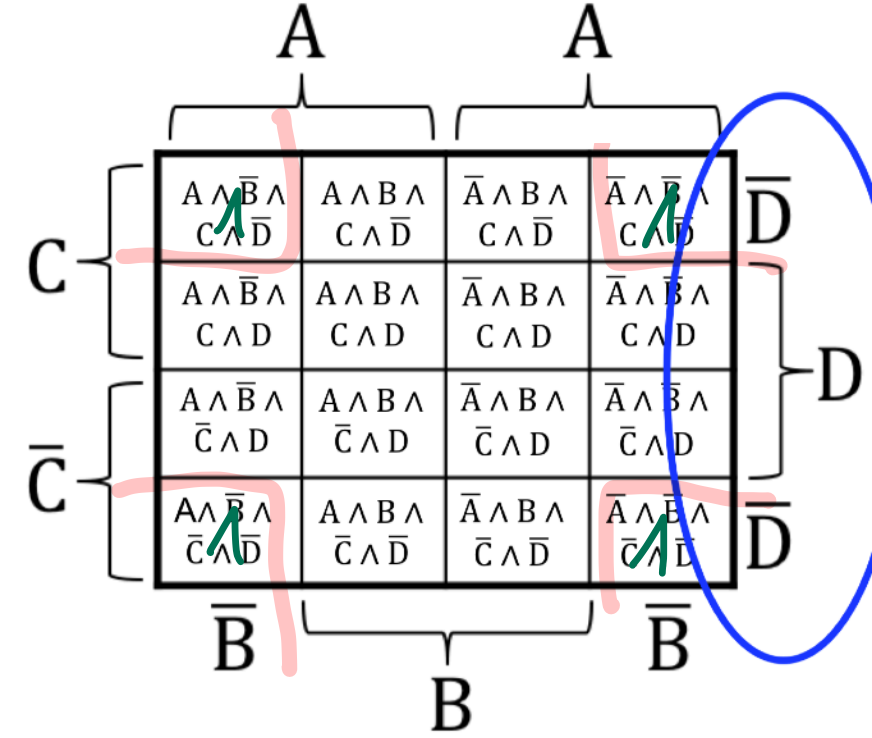
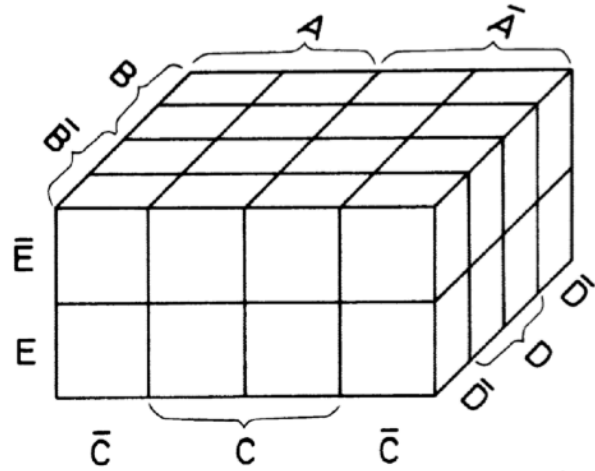
- 1 eintragen, wenn der Minterm existiert.
- 0 eintragen, wenn der Minterm nicht existiert.
- Päckchen können gebildet werden, wenn:
 - mind. eine Variable negiert und nicht negiert vorkommt
 - mind. eine Variable konstant bleibt
 - Felder könne zu mehreren Päckchen gehören
 - Aufgeschrieben wird die Variable, die konstant bleibt

AUSGANG = 1

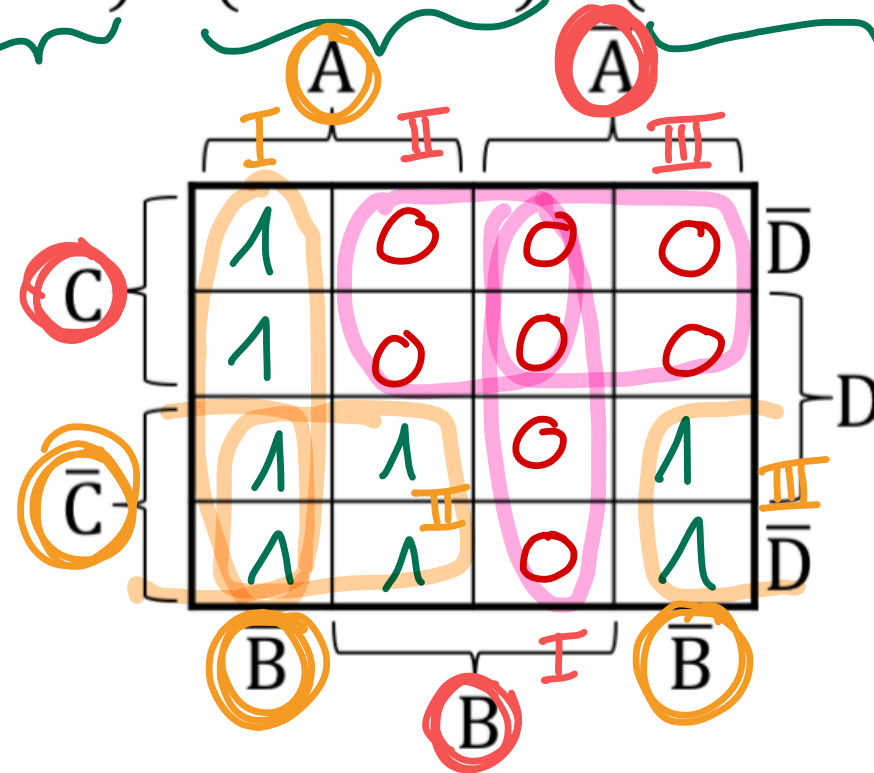
= 0

A	B	C	D	Y	
0	0	0	0	0	
0	0	0	1	1	$\rightarrow \bar{A} \wedge \bar{B} \wedge \bar{C} \wedge D$
0	0	1	0	0	
0	0	1	1	1	$\rightarrow \bar{A} \wedge \bar{B} \wedge C \wedge D$

Mehr Variablen möglich...



$$f = (\bar{B} \wedge \bar{C}) \vee (A \wedge \bar{B}) \vee (A \wedge B \wedge \bar{C}) \vee (A \wedge \bar{B} \wedge C \wedge \bar{D}) \vee (\bar{A} \wedge \bar{B} \wedge \bar{C} \wedge \bar{D})$$



$$I \quad II \quad III$$

$$(\bar{A} \wedge \bar{B}) \vee (A \wedge \bar{B}) \vee (\bar{C} \wedge A) \vee (\bar{B} \wedge \bar{C})$$

$$I \quad II \quad III$$

$$(\bar{B} \vee A) \wedge (\bar{C} \vee \bar{B}) \wedge (\bar{C} \vee A)$$

Vereinfachung mit der KNF

- 0-Päckchen
- Maxterme formen und mit UND verknüpfen
- Eingänge invertieren
- Siehe Serie

Don't Care Zustände

- Nicht benutzte Zustände können im Karnaugh Diagramm mit X gekennzeichnet werden.
- X können bei der Päckchenbildung mitbenutzt werden.

$$2^2 = 4$$

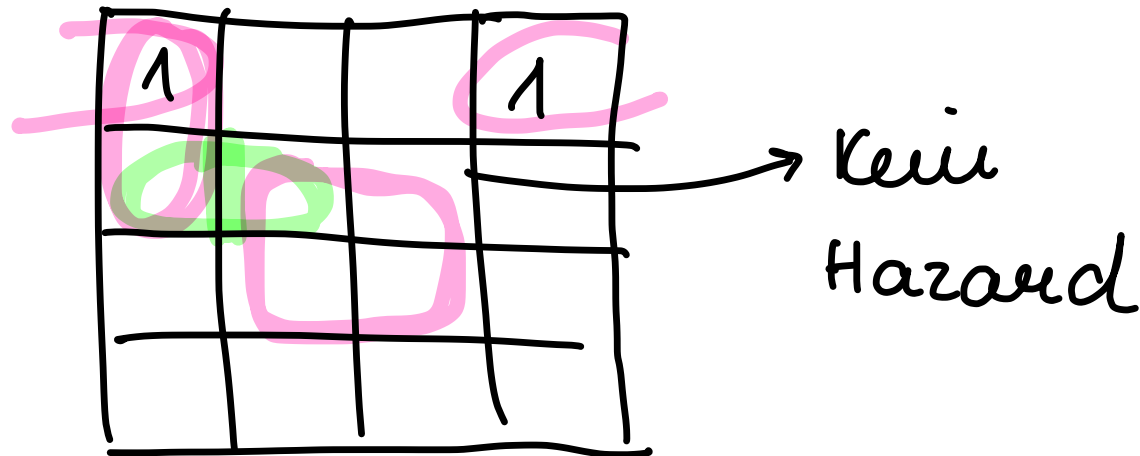
$$2^3 = 8$$

$$2^4 = 16$$

12 gebrauchte + 4 unbenutzte

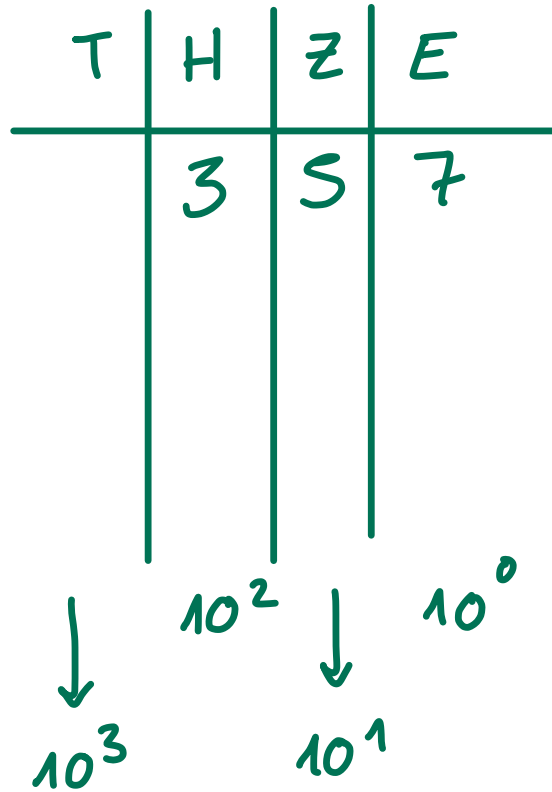
Hazards

- Durch Zeitverzögerung
- Bei Päckchen, die sich orthogonal berühren
- Behebung: Zusätzliches Päckchen um die betroffenen Stellen



Zahlen & Codes

Zahlensysteme Definition (Das Dezimalsystem)



$$D = \sum_{i=-\infty}^{i=\infty} b_i \cdot R^i$$

Basis

Zwischen 0 und R-1

$$357 = 7 * 1 + 5 * 10 + 3 * 100 = 3 * 10^0 + 5 * 10^1 + 3 * 10^2$$

Gebräuchliche Zahlensysteme

- Dezimal
 - Basis = 10
- Binär (0b)
 - Basis = 2
- Hexadezimal (0x)
 - Basis = 16
 - $b = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F\}$
- Oktal (0o)
 - Basis = 8

Umwandlung Dezimal zu System mit Basis R

- Ganze Zahl
 - Division mit R
 - Rest eintragen (von Rechts nach Links)
 - Quotient wieder Teilen bis dieser == 0
- Zahl zwischen 0 und 1
 - Mit R Multiplizieren
 - Zahl vor dem Komma eintragen (von Links nach Rechts)
 - Zahl vor dem Komma wegnehmen und wieder Multiplizieren bis ganze Zahl

Beispiel Dezimal zu System mit Basis R

21

10 R 1

5

2 R 1

1

0 R 1

42 =

0b 101010

0o 52

0x 2A

Umwandlung Binär zu Hex

- Viererpäckchen bilden (von hinten aus)
- Viererpäckchen umwandeln in Dezimal
- Zahlen >9 als Buchstaben schreiben und alles als Hex schreiben

Umwandlung Binär zu Oct

- Dreierpäckchen bilden (von hinten aus)
- Dreierpäckchen umwandeln in Dezimal
- Als Oct schreiben

Beispiel Binär zu Hex

A = 10
B = 11
C = 12

$$200 = 0b11001000 = 0x C8$$

$2^3 + 2^2 \quad 2^3 = 8$
 $8 + 4 = 12$

Beispiel Binär zu Oct

$$200 = 0o310$$

$2^0 + 2^1 \quad 2^0 \quad 0$
 $= 3 \quad = 1$

Binärzahlen addieren

$$1 + 1 = 2 = (10)_2$$
$$3 = (11)_2$$

- “Normales” schriftliches addieren

	1	1	1		1	1						
		0	1	1	0	0	1	0	1	1		
+		1	1	1	0	1	1	1	0	0		
=	1	0	1	0	1	0	0	1	1	1		

Binärzahlen subtrahieren

- Negatives Zweierkomplement zur zu subtrahierenden Zahl bilden
- “Normales” schriftliches addieren

Binärzahlen subtrahieren – Beispiel

$$A = 1101.111$$

$$A = 00001101,111$$

$$B = 111000.001$$

$$B = 0011000,001$$

$$\begin{array}{r} \hline 1100011,110 \\ + 1 \\ \hline 1100011,111 \end{array}$$

$$C = A - B =$$

$$\begin{array}{r} 00001101,111 \\ + 11000111,111 \\ \wedge \wedge \wedge \wedge \wedge \wedge \\ \hline 11010101,110 \\ \hline \hline \end{array}$$

Codes

Binär	BCD	Excess-3	Aiken	4-2-2-1	Gray	O'Brien
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				
1001	9	6				
1010		7				9
1011		8	5			
1100		9	6	6	8	5
1101			7	7	9	6
1110			8	8		7
1111			9	9		8

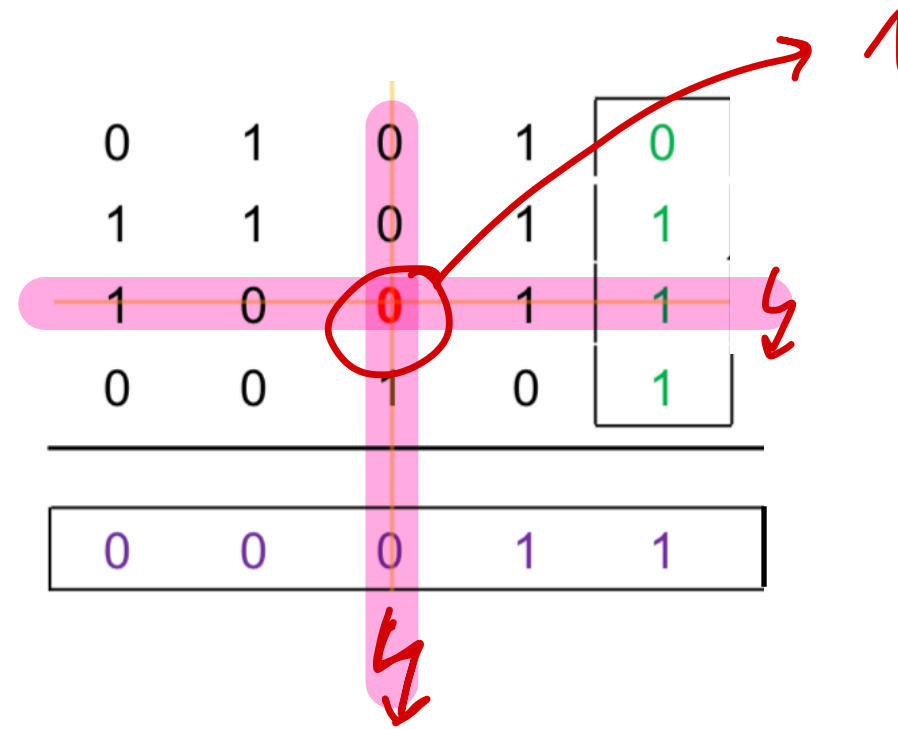
Einzelne Ziffern in Binärdarstellung

Beim Zählen verändert sich immer nur eine Zahl

Fehlererkennung

01010 →
11011 →
10111 →
00101 →
00011 →

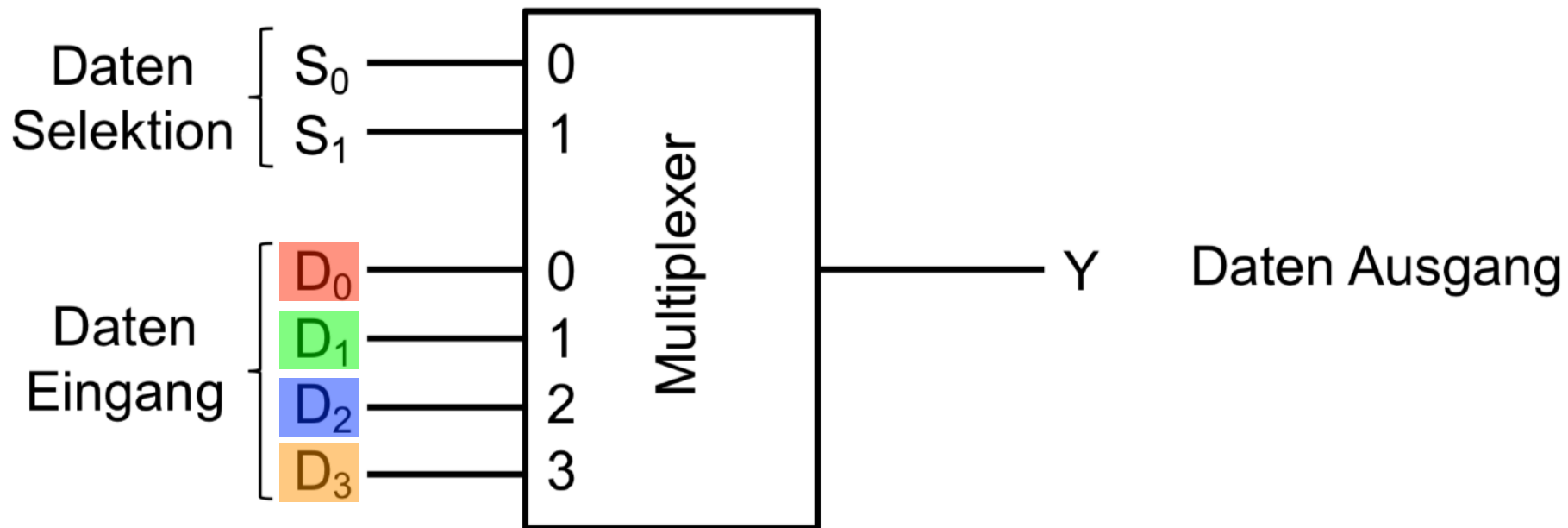
- Code wird in regelmässigen Abständen (Tetrade) um ein Parity Bit ergänzt
 - **Even Parity Bit** = 0 wenn Anzahl 1 gerade
 - Odd Parity Bit = 0 wenn Anzahl 1 ungerade
- Am Ende des übertragenen Codes wird ein **Prüfwort** angehängt



Rechenschaltungen

Multiplexer

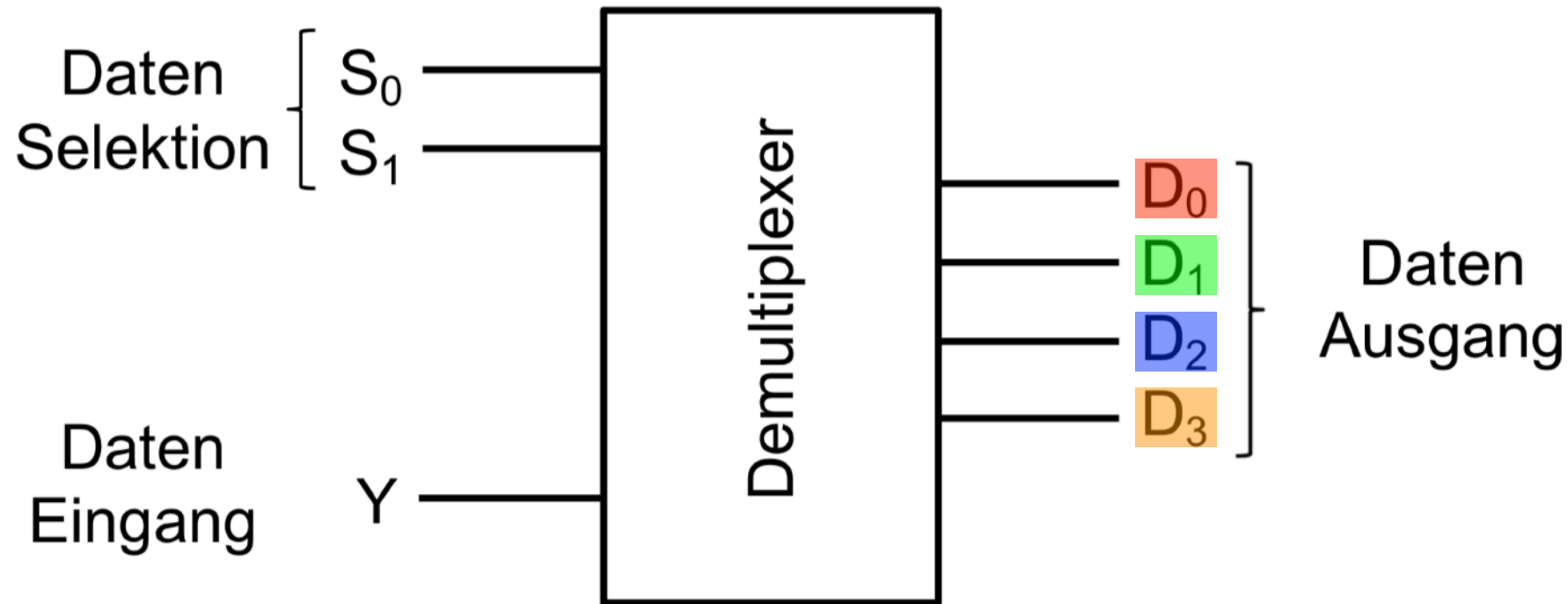
Mehrere Eingänge auf einen Ausgang



$$Y = (\bar{S}_0 \wedge \bar{S}_1 \wedge D_0) \vee (\bar{S}_0 \wedge S_1 \wedge D_1) \vee (S_0 \wedge \bar{S}_1 \wedge D_2) \vee (S_0 \wedge S_1 \wedge D_3)$$

Demultiplexer

Ein Eingang auf mehrere Ausgänge



$$D_0 = \overline{S_0} \wedge \overline{S_1} \wedge Y,$$

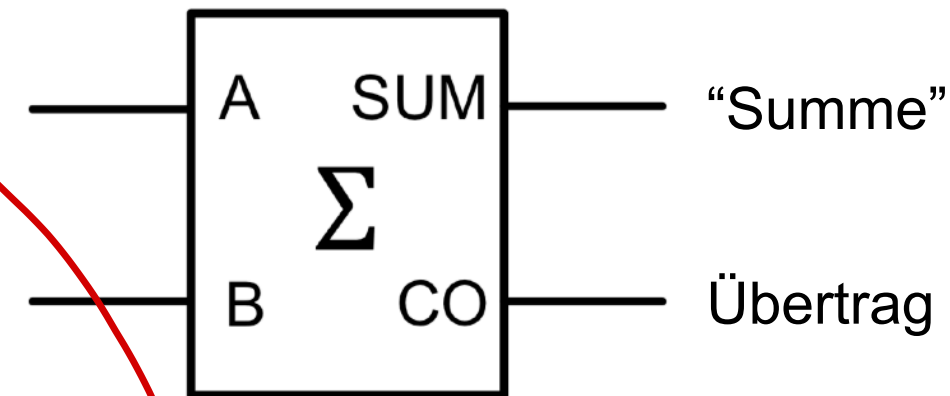
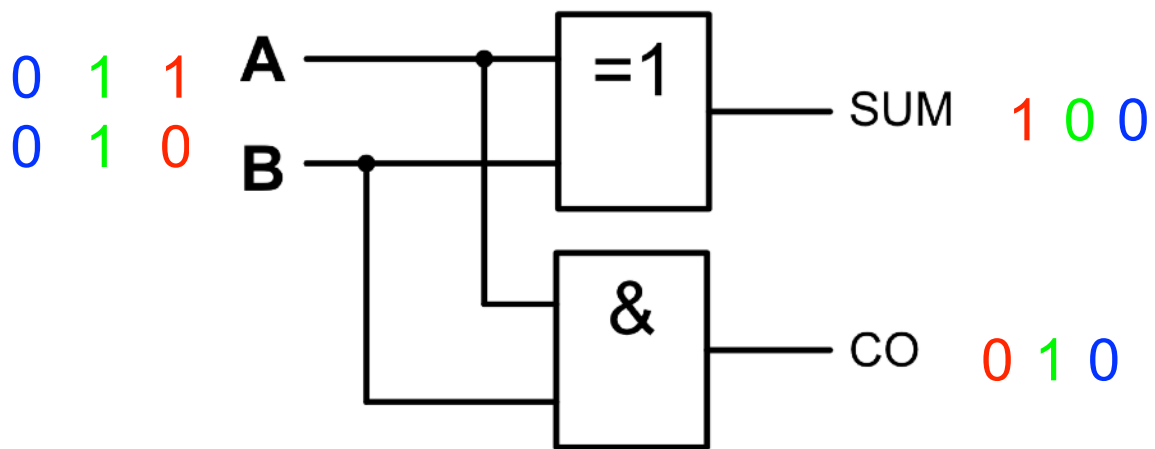
$$D_1 = \overline{S_0} \wedge S_1 \wedge Y,$$

$$D_2 = S_0 \wedge \overline{S_1} \wedge Y,$$

$$D_3 = S_0 \wedge S_1 \wedge Y$$

Halbaddierer

- Um 2 Bits zu addieren

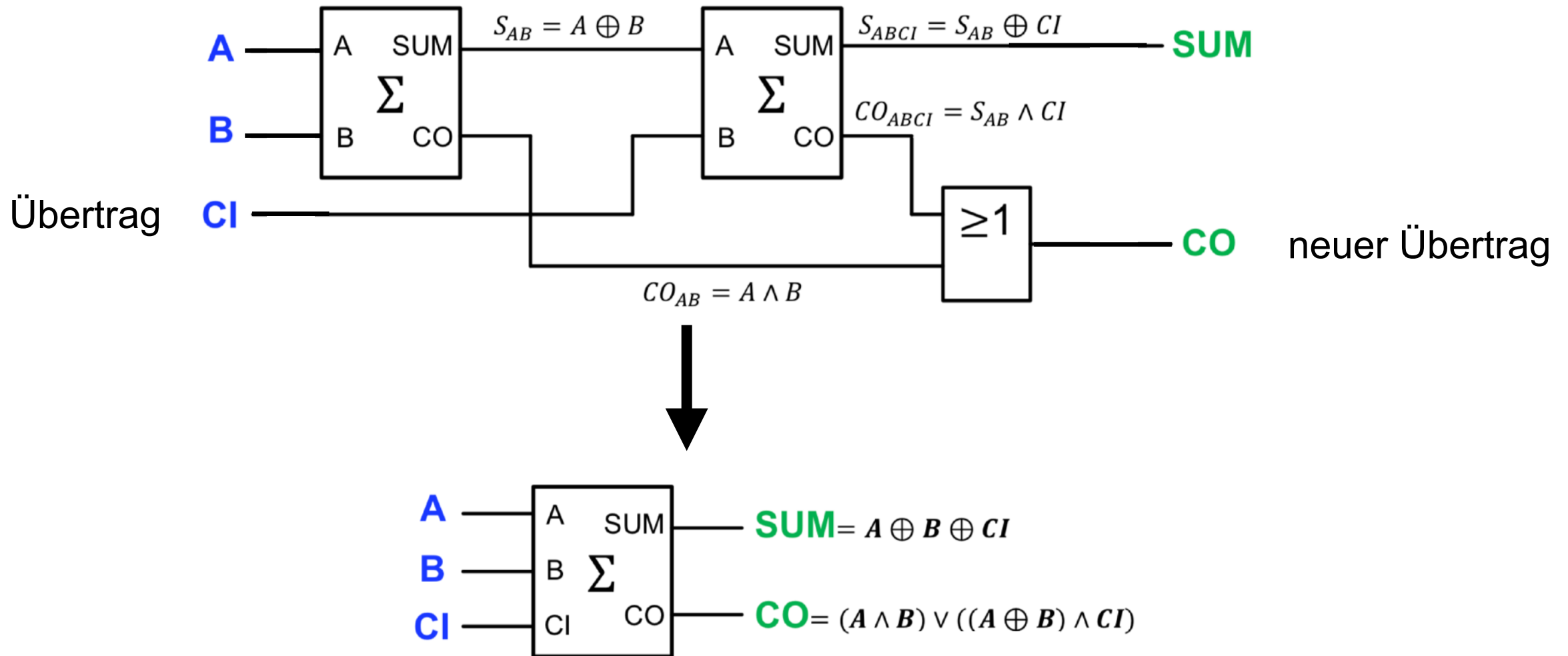


$$\begin{array}{r} 1 \\ 0 \\ 1 \\ \hline \end{array} + \begin{array}{r} 1 \\ 1 \\ \hline \end{array} = \begin{array}{r} 0 \\ \hline \end{array}$$

Was machen wir, wenn wir 3 Bits addieren wollen?

Volladdierer

- Um 3 Bits zu addieren
- $A + B + C = (A + B) + C$



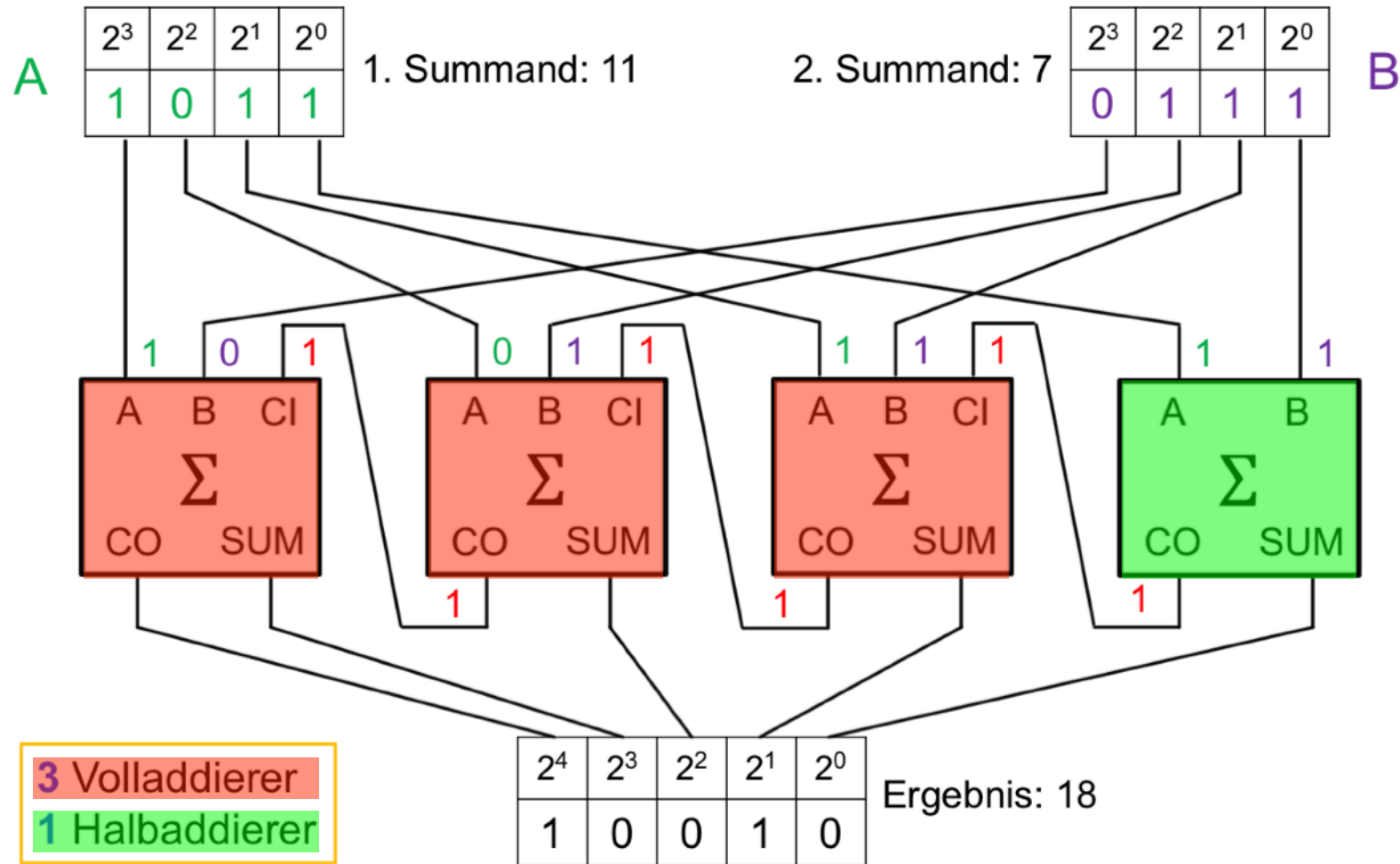
Mehrbit-Addierer

- Wir wollen ganze Dualzahlen addieren
- Serienaddierer: in jedem Taktschritt wird eine Stelle addiert
- Paralleladdierer: in einem Taktschritt werden alle Stellen addiert

Paralleladdierer

- Für jedes Bit eine Schaltung generieren
 - $S_0 = A_0 + B_0$
 - $S_1 = A_1 + B_1 + C0_0$
- Vorteile
 - Schnell, da die Daten max. durch 3 Gatter müssen
- Nachteile
 - sehr Schaltungsaufwendig

Ripple-Carry Addierer



Ripple-Carry Addierer II

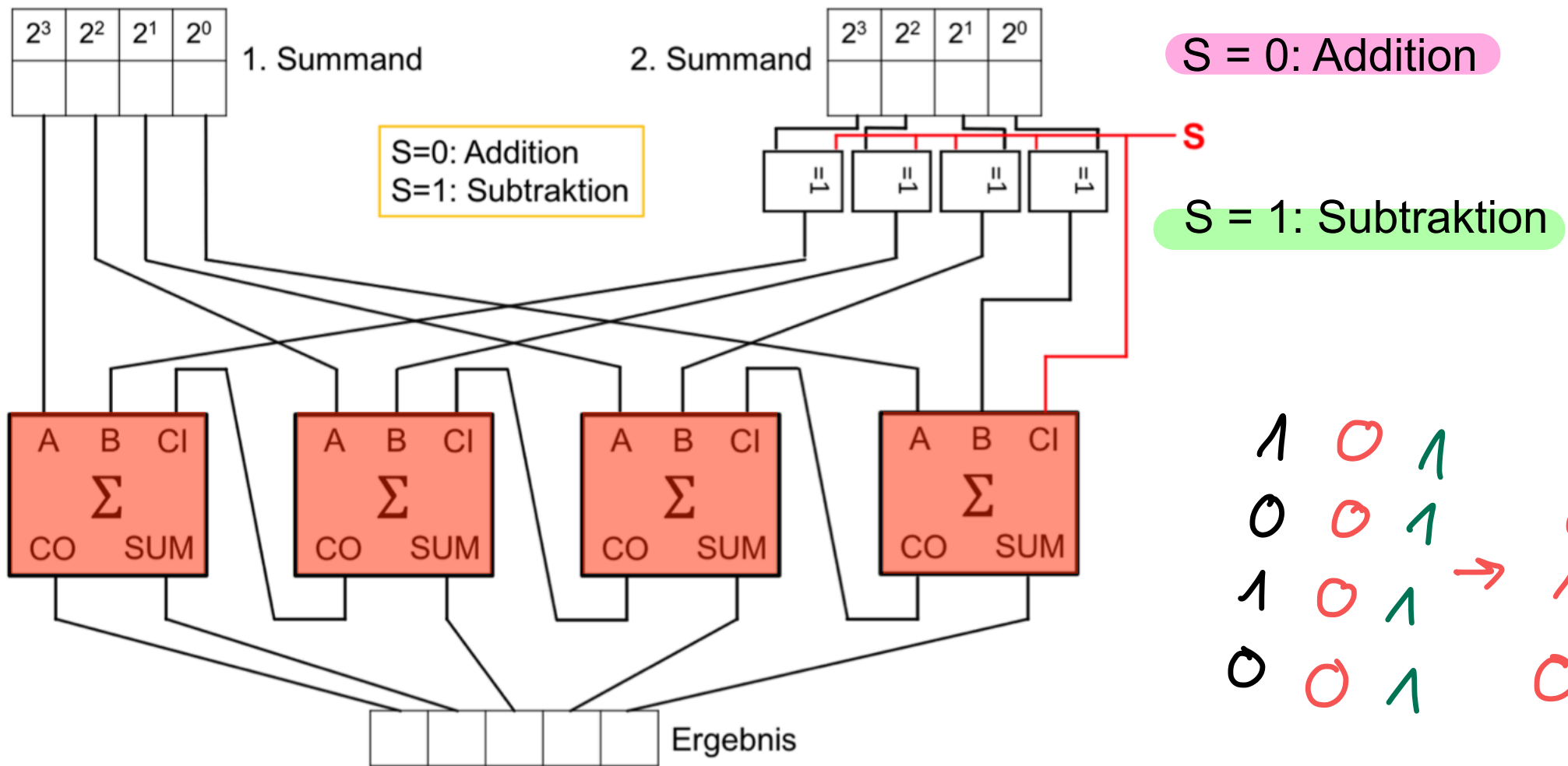
- Vorteile
 - Schaltungsaufwand wächst linear
 - Einfach erweiterbar
- Nachteile
 - langsam, da die Daten durchrieseln müssen

Carry-Lookahead Addierer

- Addierer werden kaskadiert
- Überträge werden parallel berechnet

- Vorteile
 - Gleich schnell wie der Paralleladdierer
 - Schaltungsaufwand wächst linear (wie bei Ripple-Carry)

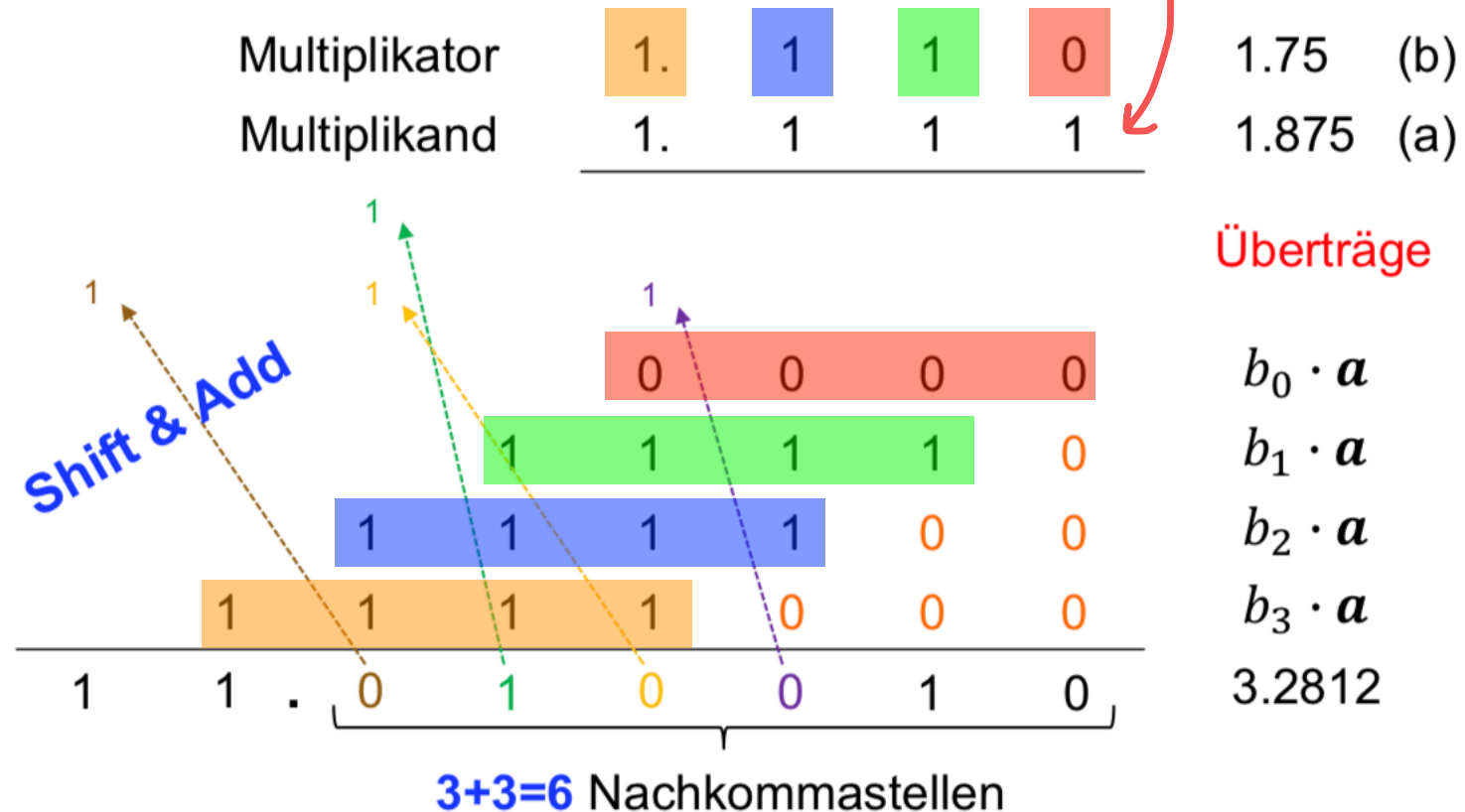
Subtrahierer



Multiplikation von Binärzahlen

A · B

- 1. Übertrag
 - 2. Übertrag
 - 3. Übertrag
 - 4. Übertrag



Booth-Algorithmus

$A \cdot B$

$$A = 1010$$

$$B = 111$$

$$= 0111$$

(0 0): nichts machen

(1 0): -B

(0 1): +B

Shift

$$A = 0100$$

$$B = 0111$$

$$-B = 1001$$

Init.



	Produkt	LB
Init.	0000'0100	0)
→	00000010	0)
→	00000001	0)
	+ 1001	
	10011000	
→	11001100	1)
	+ 0111	
	00111000	
→	00011100	0)

Latches & Flipflops

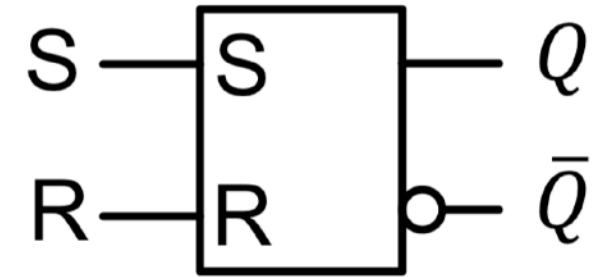
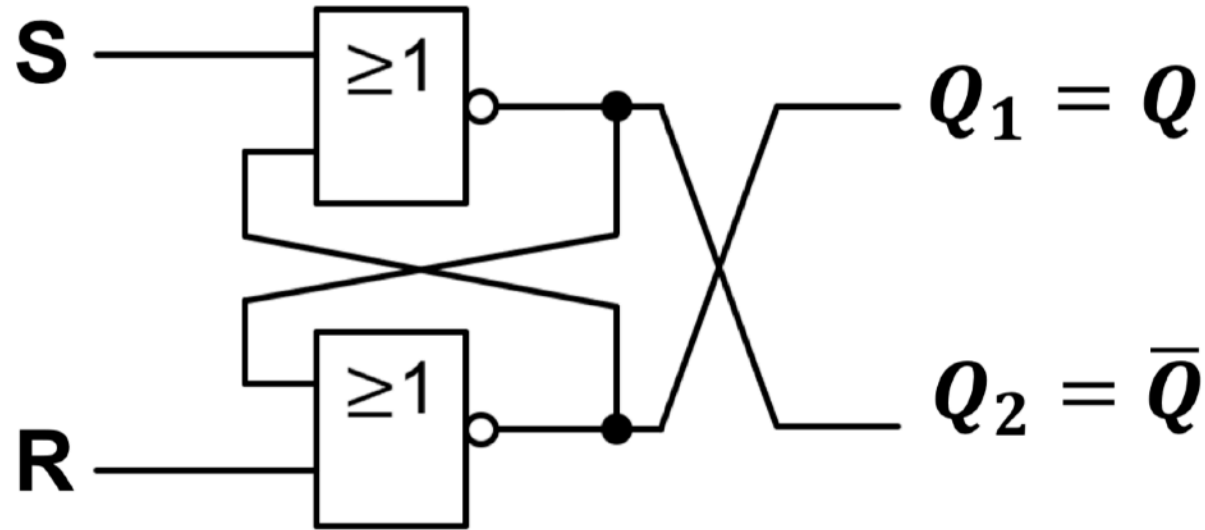
Kombinatorische Schaltungen

- Ausgang hängt nur von den Eingängen und Gattern ab
- Bisher nur solche betrachtet

Sequentielle Schaltungen

- Besitzen Rückkoppelungen
- jetzt neu

SR-Latch



Fall	S	R	Q_{1n}	Q_{1n+1}
1	0	0	0	0
2	0	0	1	1
3	0	1	0	0
4	0	1	1	0
5	1	0	0	1
6	1	0	1	1
7	1	1	0	X
8	1	1	1	X

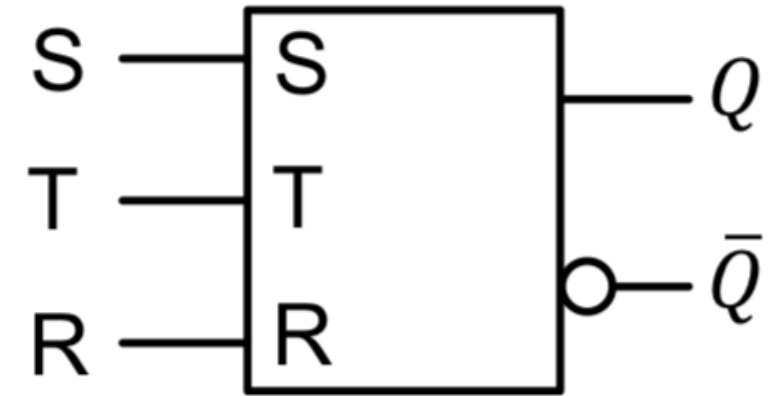
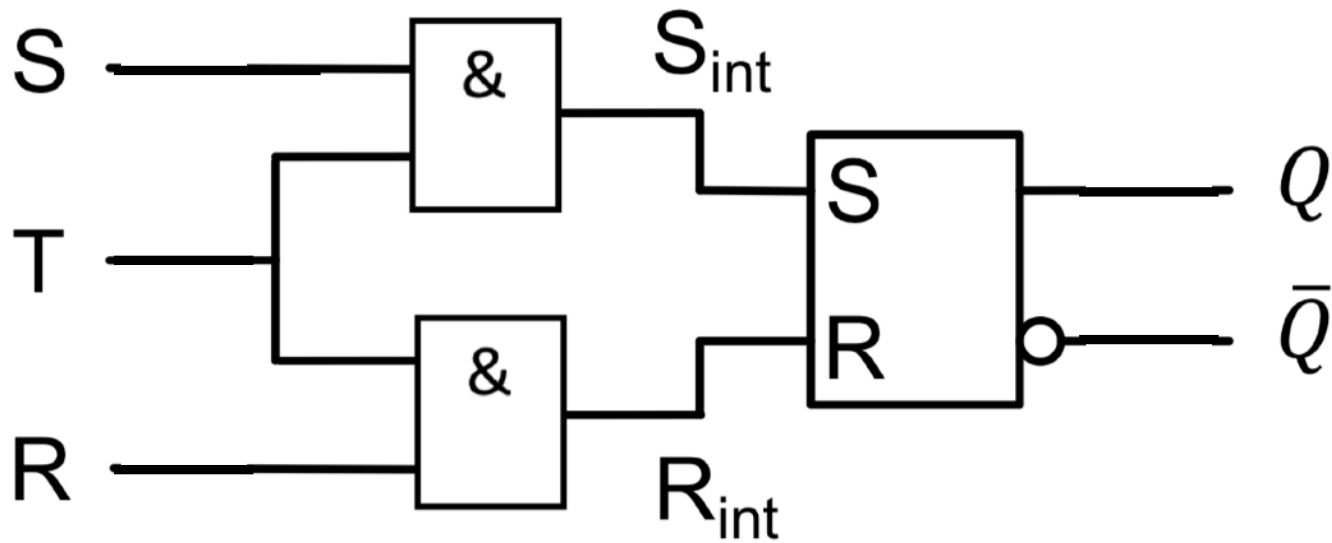
speichern

rücksetzen

setzen

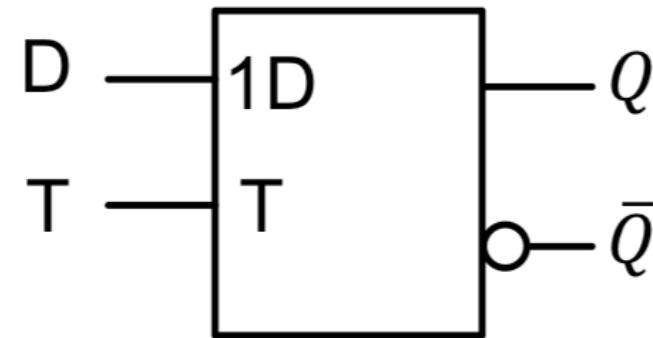
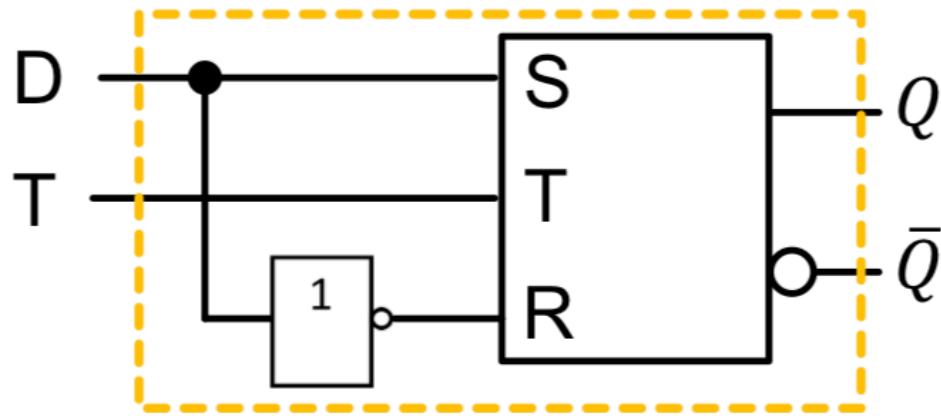
unzulässig

Taktzustandgesteuertes SR-Latch



Gleich wie SR-Latches, nur dass sich Q nur verändern kann, wenn $T = 1$

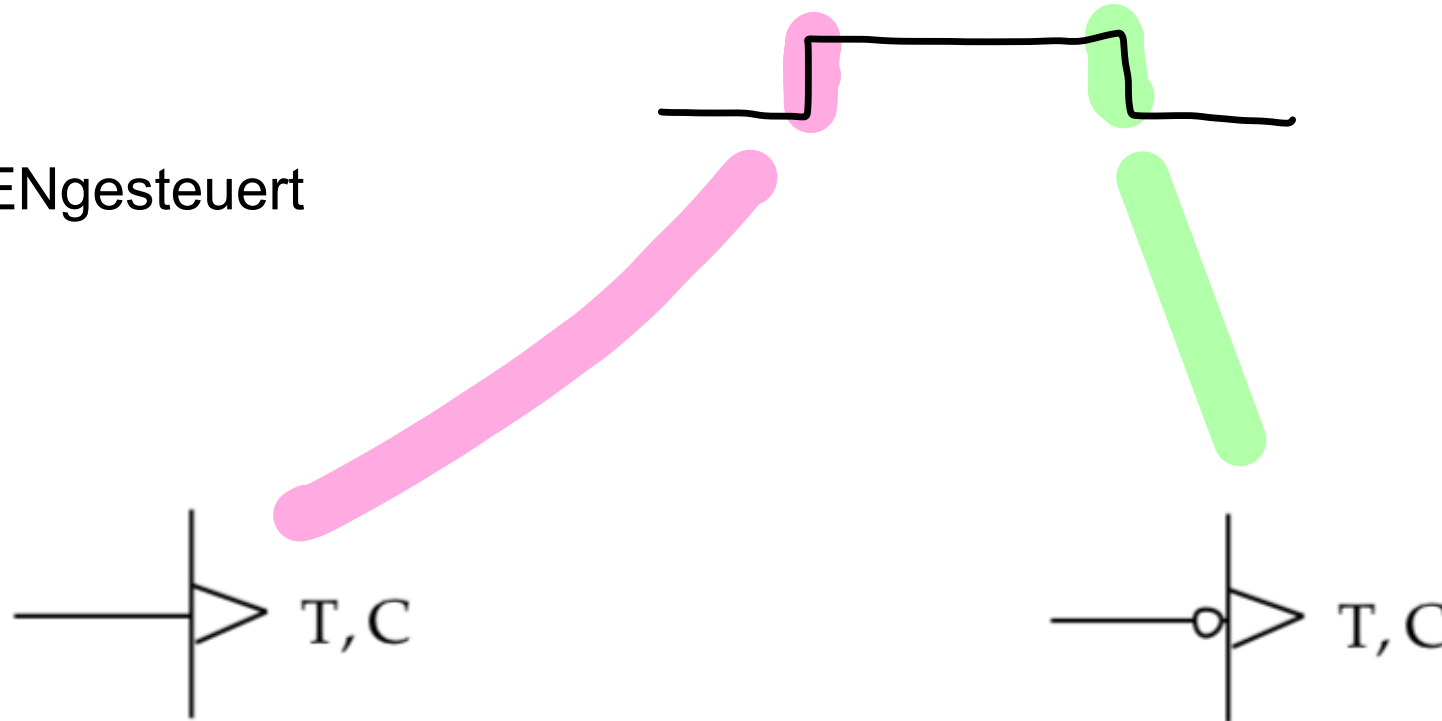
D-Latches



- Die Eingangskombination $R = S = 1$ wird verhindert
- $Q = D$ während $T = 1$
- Bei $T = 0$ wird der letzte Wert vor dem Übergang beibehalten

Flipflops

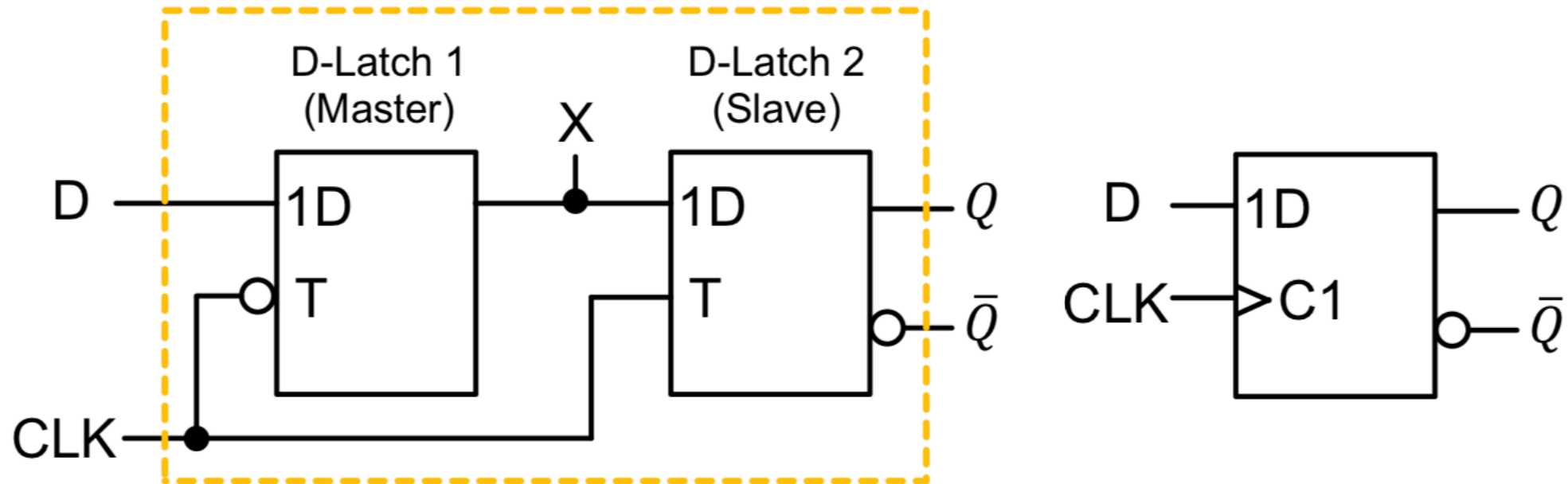
- TaktFLANKENgesteuert



Eingangsvariable werden beim
0 - 1 Übergang von C wirksam

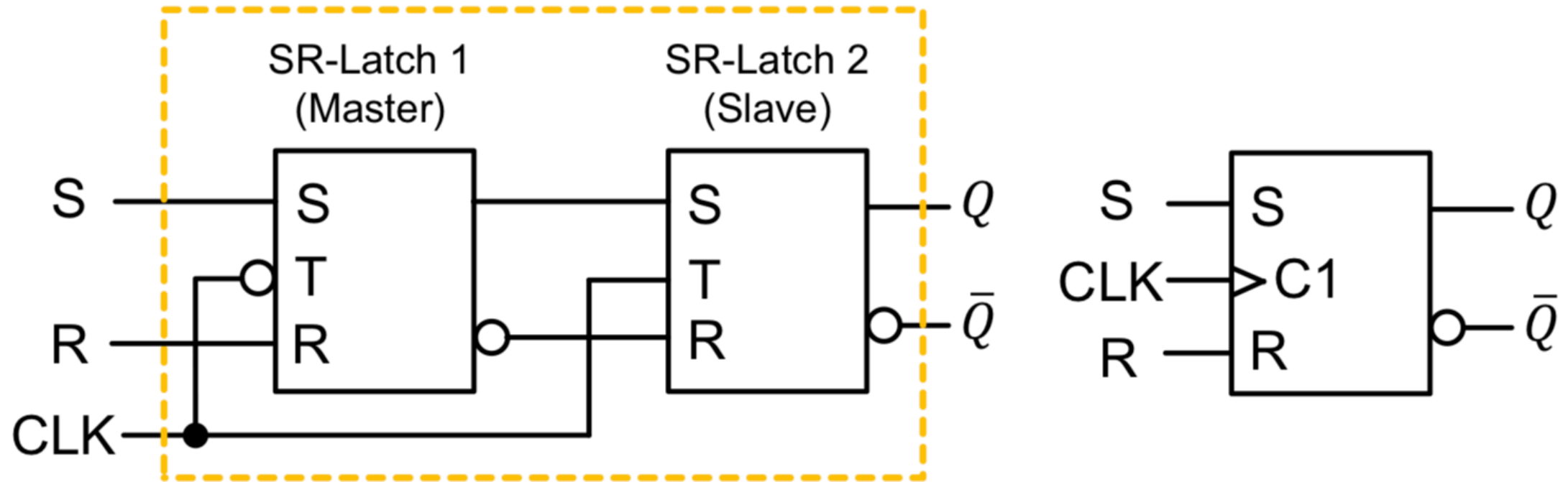
Eingangsvariable werden beim
1 - 0 Übergang von C wirksam

D-Flipflop

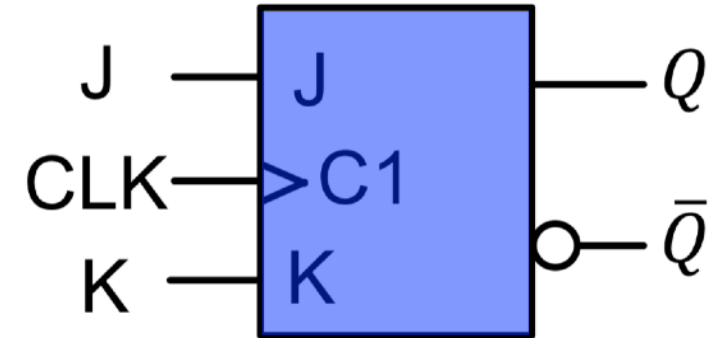
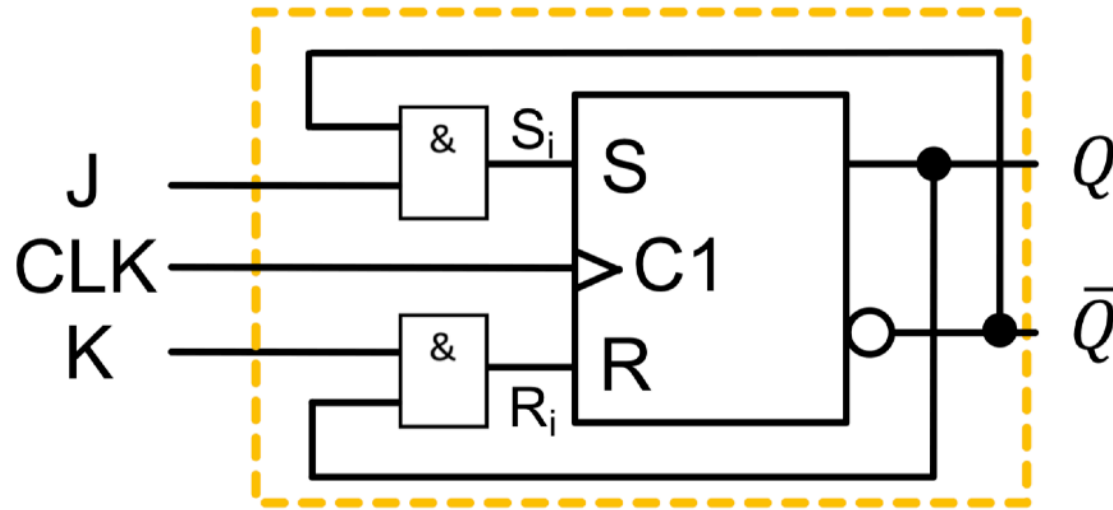


Ausgang wird zu D bei jeder positiven Taktflanke

SR-Flipflop



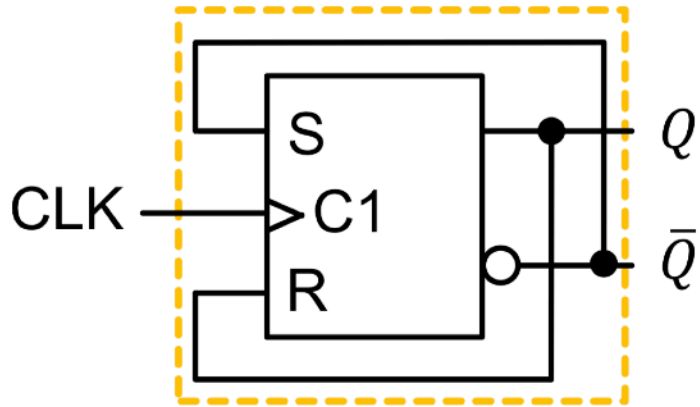
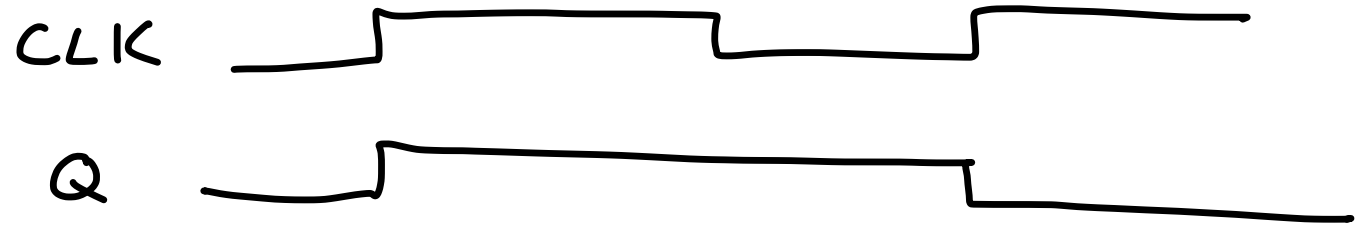
JK-Flipflop (Jump - Kill)



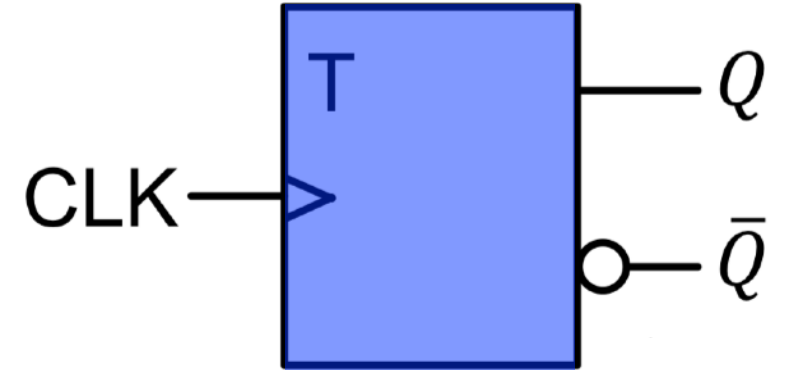
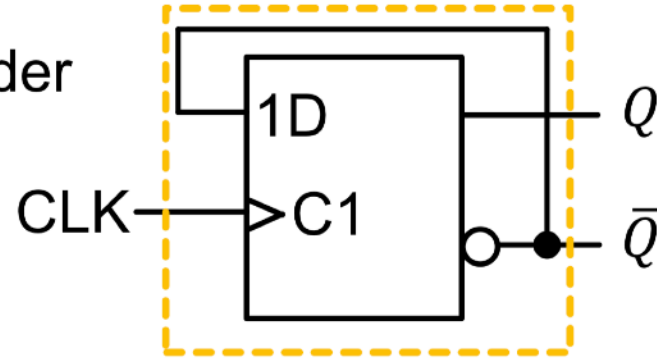
Fall	J	K	Q_{1n+1}	Q_{2n+1}
1	0	0	Q_{1n}	Q_{2n}
2	0	1	0	1
3	1	0	1	0
4	1	1	$\overline{Q_{1n}}$	$\overline{Q_{2n}}$

speichern
rücksetzen
setzen
wechseln

T-Flipflop (Toggle)

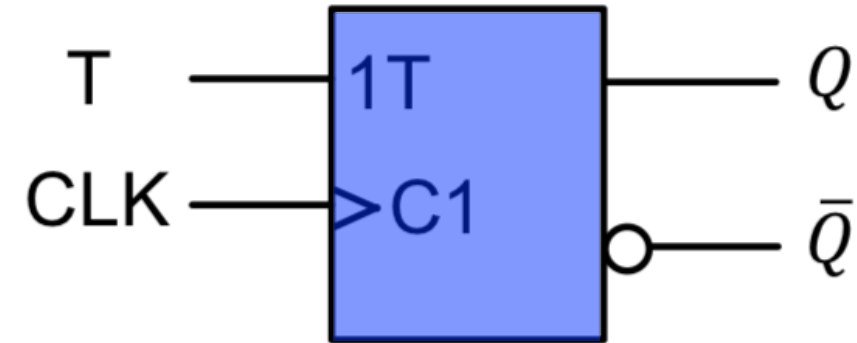
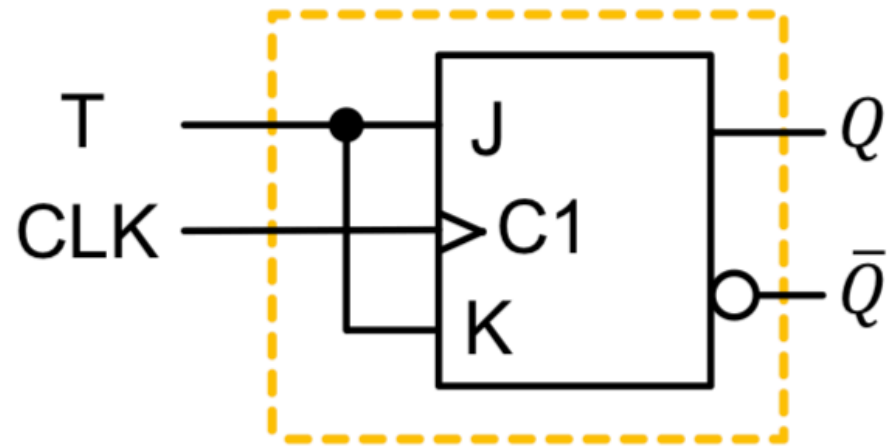


oder



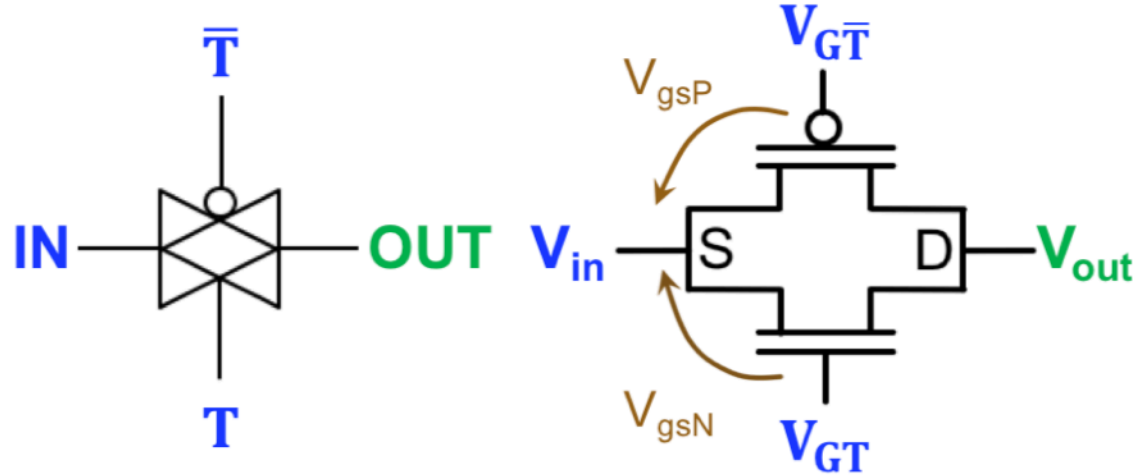
Signal "toggelt" (= wechselt) bei jeder Taktflanke

T-Flipflop mit 2 Eingängen



Signal "toggelt" nur, wenn $T = 1$

Transmission Gate



Schaltsymbol

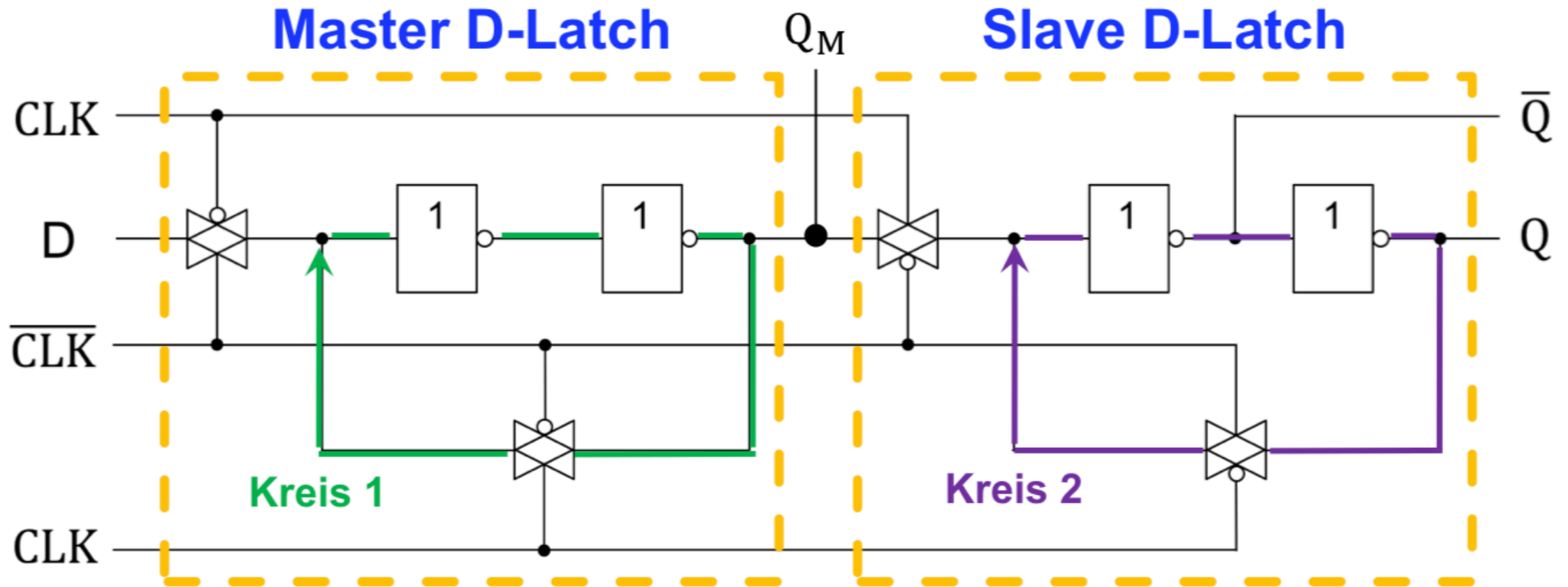
CMOS Schaltung

IN	T	Widerstand	OUT
0	0	hochohmig	-
0	1	niederohmig	0
1	0	hochohmig	-
1	1	niederohmig	1

PMOS leitet, wenn $V_s = 1$ und $V_g = 0$

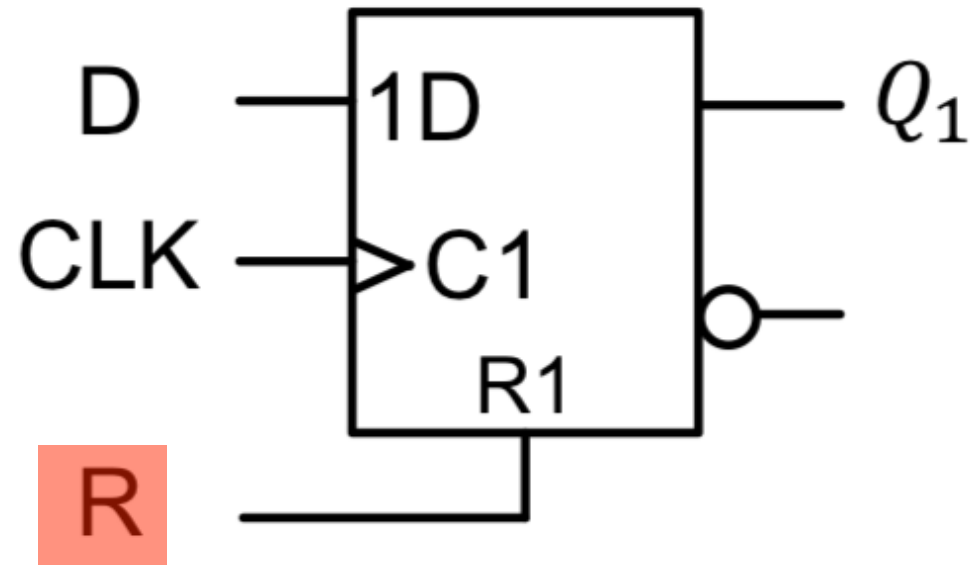
NMOS leitet, wenn $V_s = 0$ und $V_g = 1$

D-Flipflop mit Transmission Gates



8 NMOS und 8 PMOS

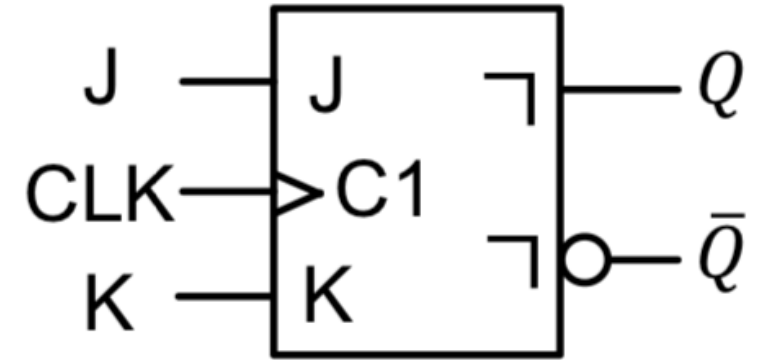
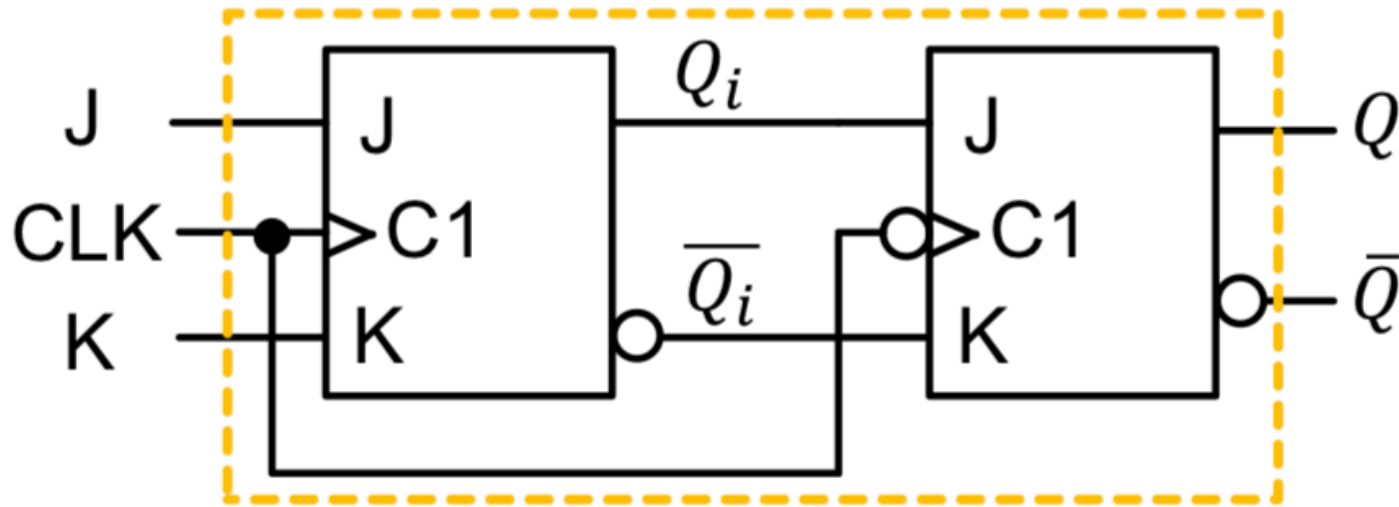
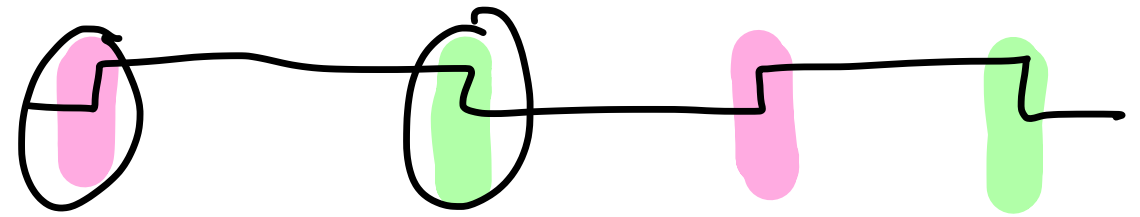
Rücksetz-Eingang



Speicherzustand kann jederzeit auf 0 zurückgesetzt werden

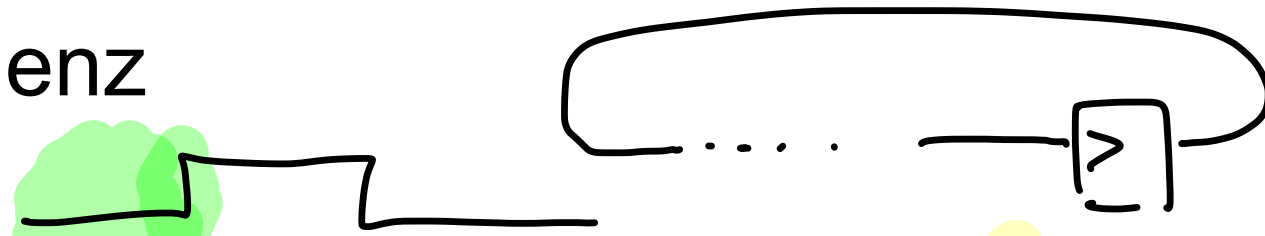
Master-Slave Flipflop

CLK



Informationen werden bei steigender Taktflanke eingelesen aber erst bei fallender ausgegeben

Maximale Taktfrequenz



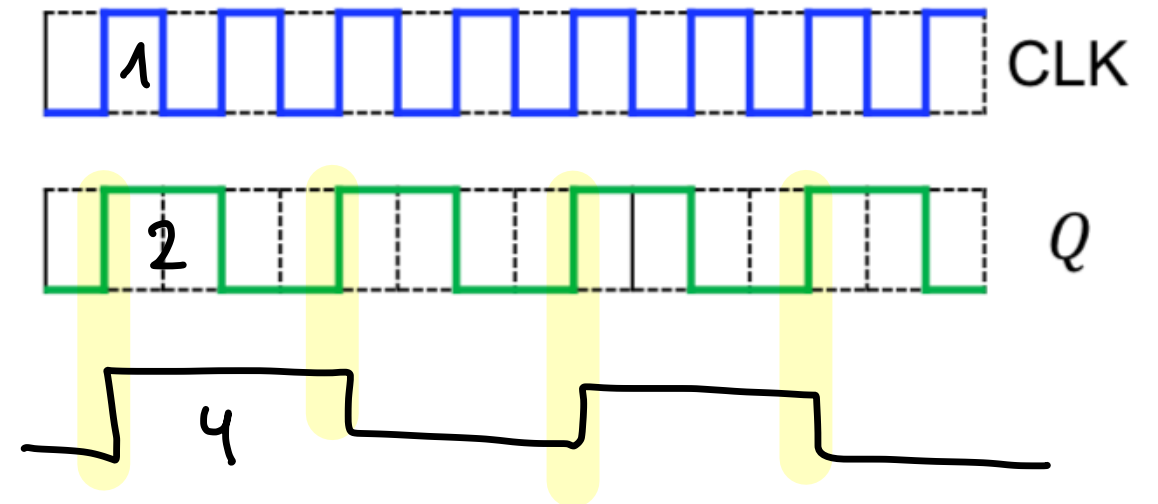
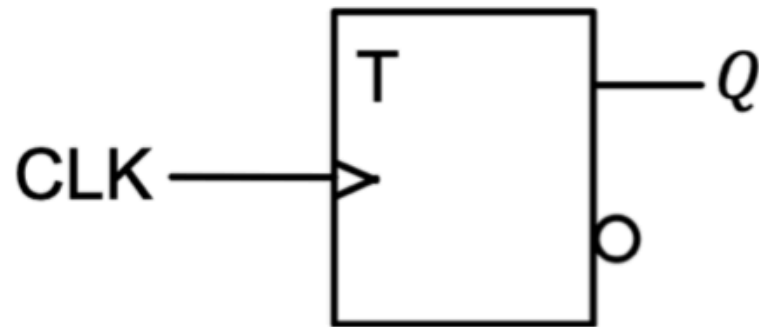
- Wenn mindestens 2 Flipflops in Serie
- t_{pd} = Verzögerungszeit (Durchlaufzeit)
- t_s = Setup-Zeit (Wie lange ein Signal vor einer Taktflanke unverändert sein muss)
- t_h = Hold-Zeit (Wie lange ein Signal nach einer Taktflanke unverändert sein muss)

$$T_{min} - t_{s,ff2} \geq t_{pd,ff1} + t_{pd,ks}$$

$$\underline{\underline{f_{max} = 1/T_{min}}}$$

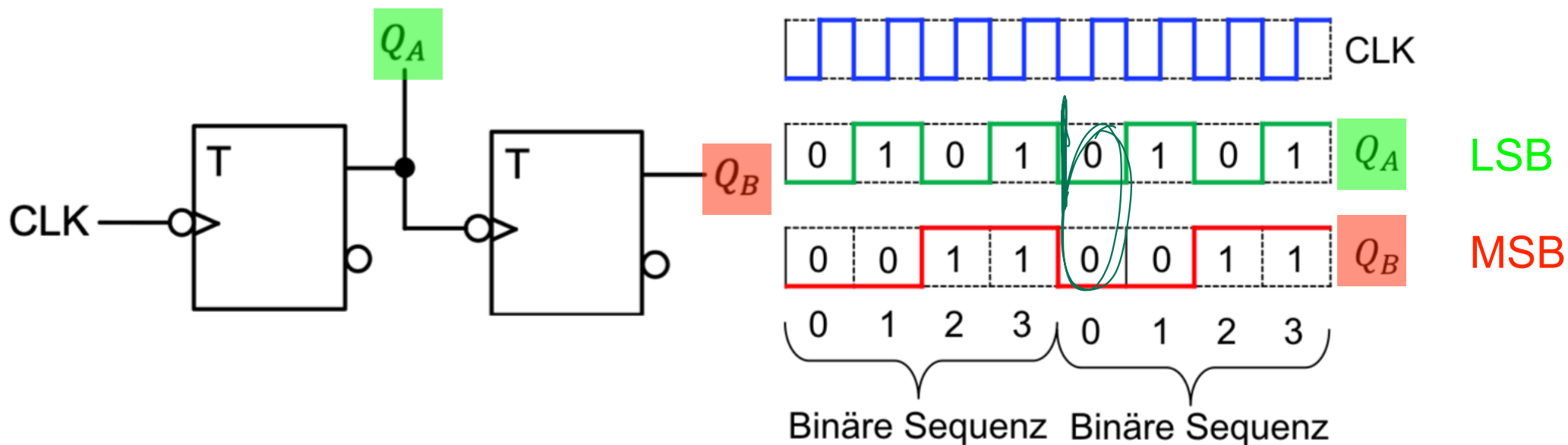
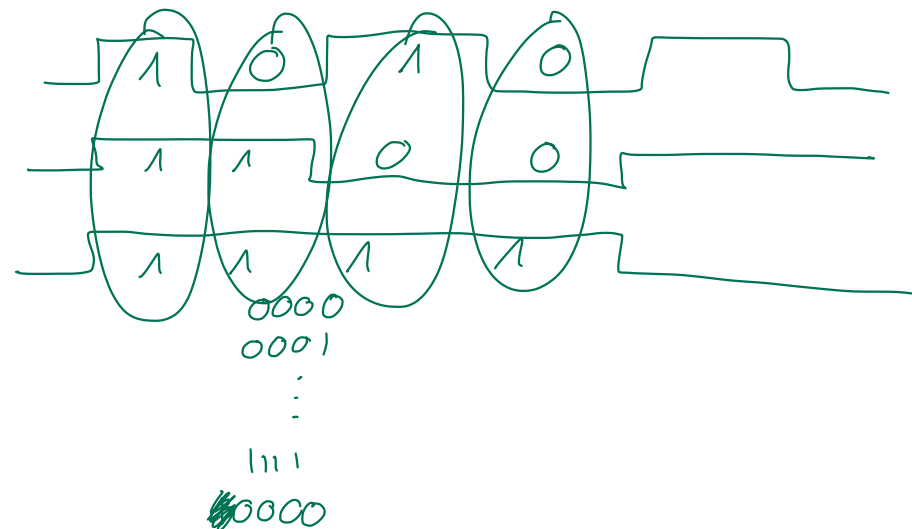
Frequenzteiler

- T-Flipflops oder J-K-Flipflops mit $J = K = 1$
- n Flipflops in Serie führen zu einer Reduktion um 2^n



Dualzähler

- T-Flipflops oder JK-Flipflops mit $J = K = 1$
- Bei n Flipflops Zahlen bis $2^n - 1$



Automaten

Automaten



Hier nur synchrone Automaten betrachtet (alle FF haben das gleiche Taktsignal)

Finite State Machine (= Endliche Automaten)

- mögliche Eingabebezeichen
 - mögliche Ausgabebezeichen
 - intern gespeicherte Zustände
- } sind endlich

Beschreibung von Automaten

- X = Eingabealphabet
- Y = Ausgabealphabet
- Z = Zustandsmenge
- Z_0 = Anfangszustand
- F_{c1} = Übergangsfunktionen
- F_{c2} = Ausgangsfunktionen

Mealy - Automat

- Ausgang von Eingang und internem Zustand abhängig
 - $Y_n = f_{C2}(X_n, Z_n)$
 $Z_{n+1} = f_{C1}(X_n, Z_n)$
-

Moore - Automat

- Sonderfall vom Mealy-Automat
- Ausgang hängt nur vom internen Zustand ab
- $Y_n = f_{C2}(Z_n)$
 $Z_{n+1} = f_{C1}(X_n, Z_n)$
- Medwedjew-Automat, wenn Ausgang = interner Zustand ($Y_n = Z_n$)

Zustandsfolgetabelle

$$n \rightarrow 2^n$$

$$v_{max} = 2^{e+m}$$

No	Eingang X_n	Momentaner Zustand Z_n	Folgezustand Z_{n+1}	Ausgang Y_n
1	x_1, x_2, \dots, x_e	$Z_{1n}, Z_{2n}, \dots, Z_{mn}$	$Z_{1n+1}, Z_{2n+1}, \dots, Z_{mn+1}$	y_1, y_2, \dots, y_b
\vdots				
v				

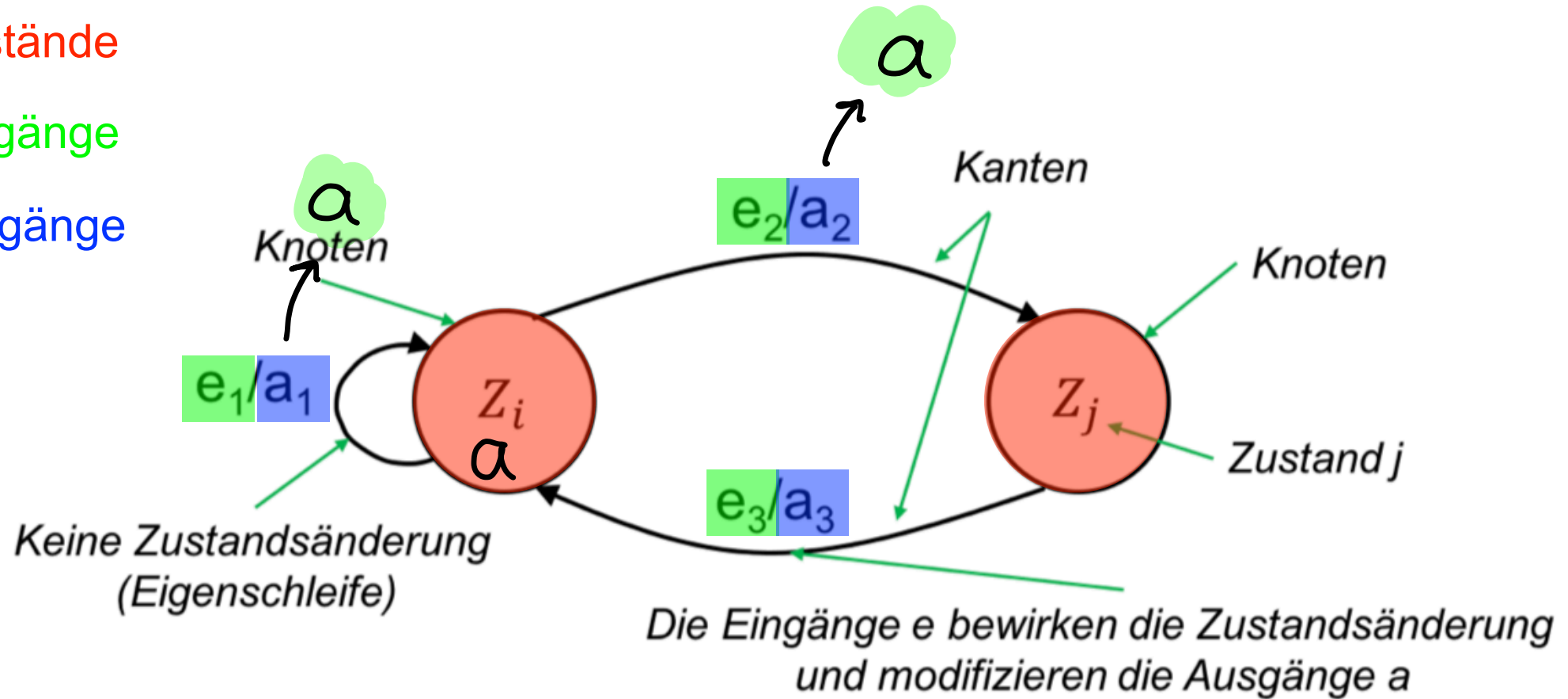
$$e+2m+b$$

Zustandsdiagramm – Mealy

Zustände

Eingänge

Ausgänge

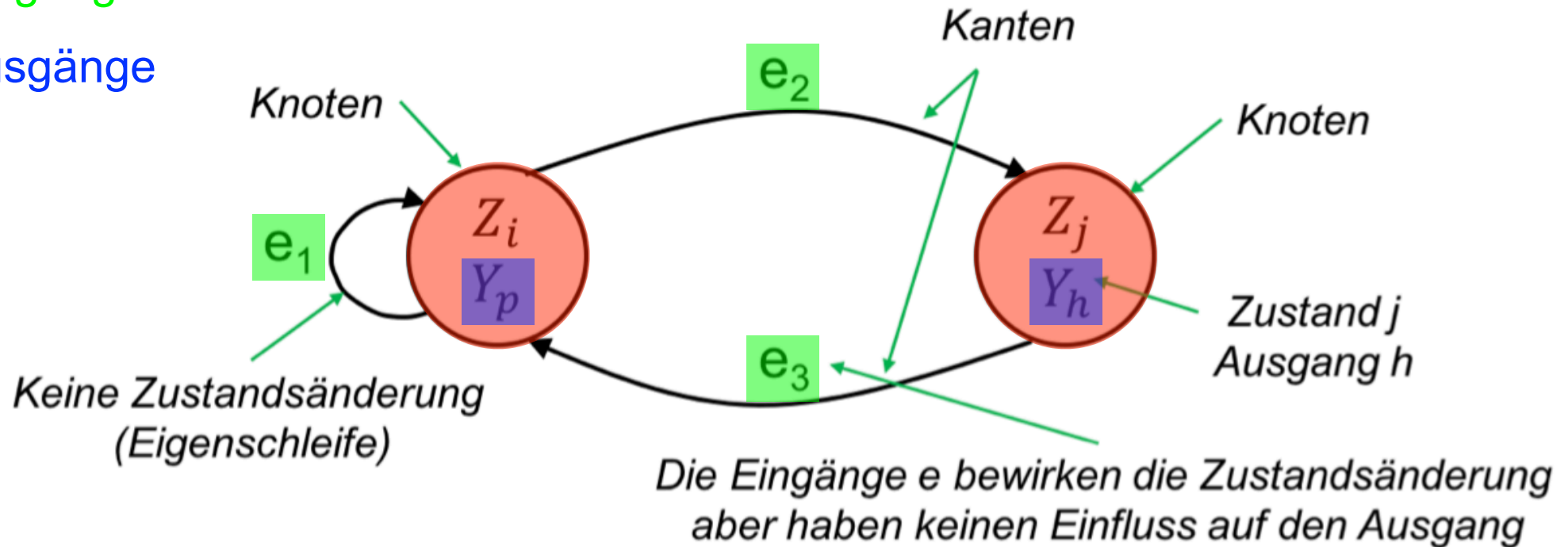


Zustandsdiagramm – Moore

Zustände

Eingänge

Ausgänge



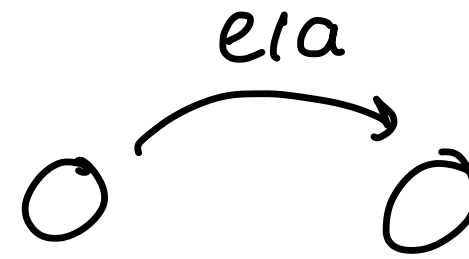
Automatensynthese

- Zustandsmenge bestimmen
- Ein- und Ausgangvariablen bestimmen und Zustände kodieren
- Zustandsdiagramm zeichnen
- Zustandsfolgetabelle aufstellen
- Die Ausgangs- und Zustandsübertragungsfunktionen herausfinden
- Schaltplan zeichnen

Mealy - Moore - Umwandlung

- Mealy lässt sich immer in Moore umwandeln
- Einfach wenn die Zustände immer den gleichen Ausgang produzieren
- Sonst neue Zustände definieren

Dynamisches Verhalten

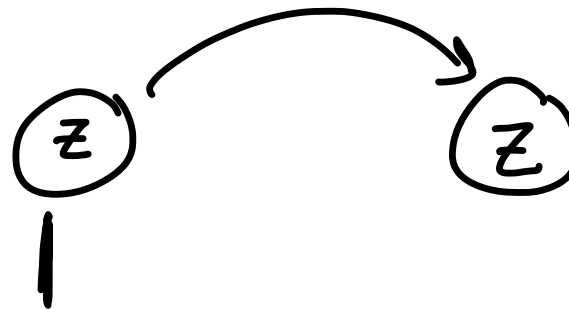


Mealy-Automat

- Änderung des Eingangs wird sofort am Ausgang sichtbar

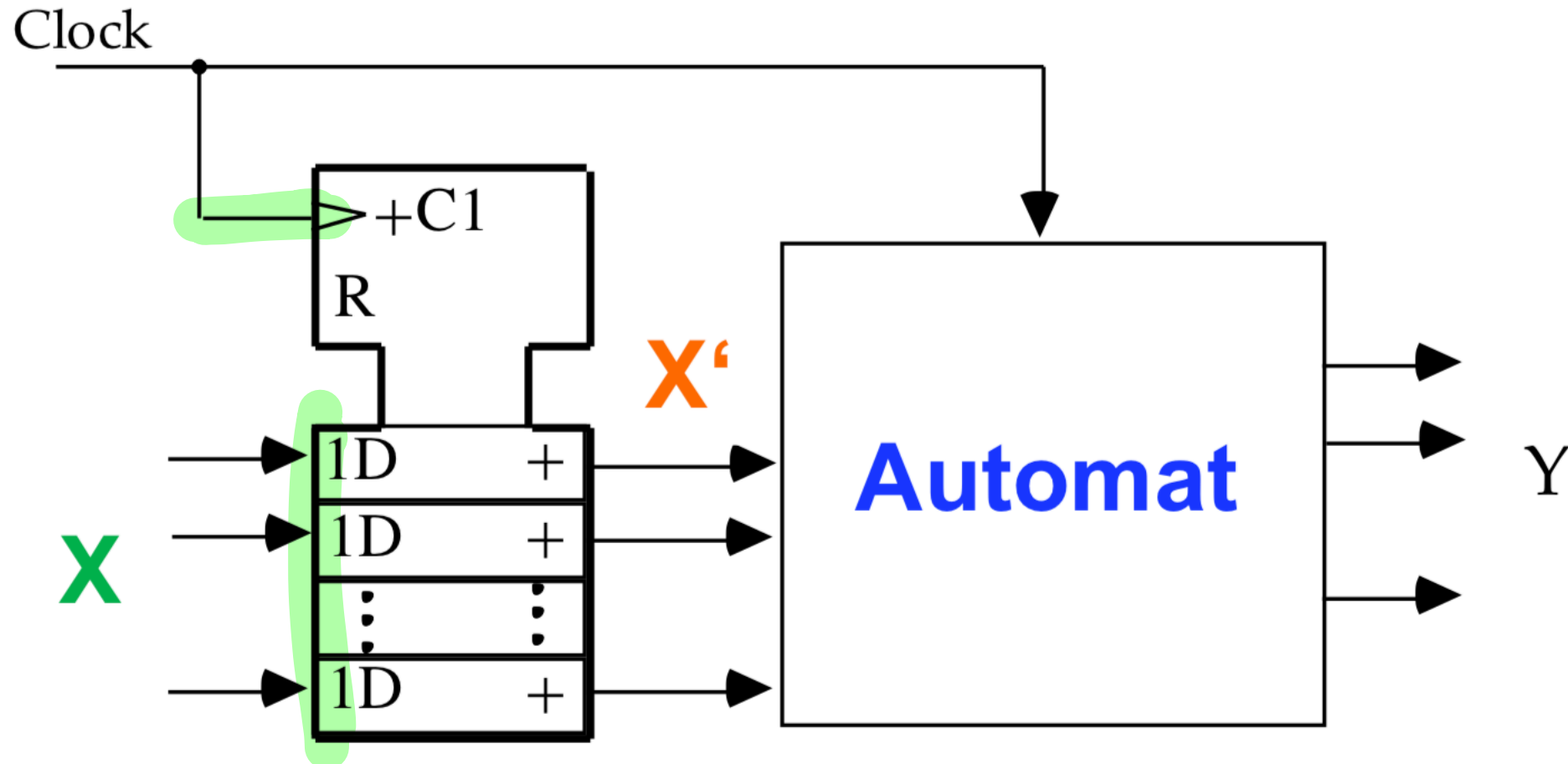
Moore-Automat

- Änderung des Eingangs wird erst bei der nächsten Taktflanke am Ausgang sichtbar



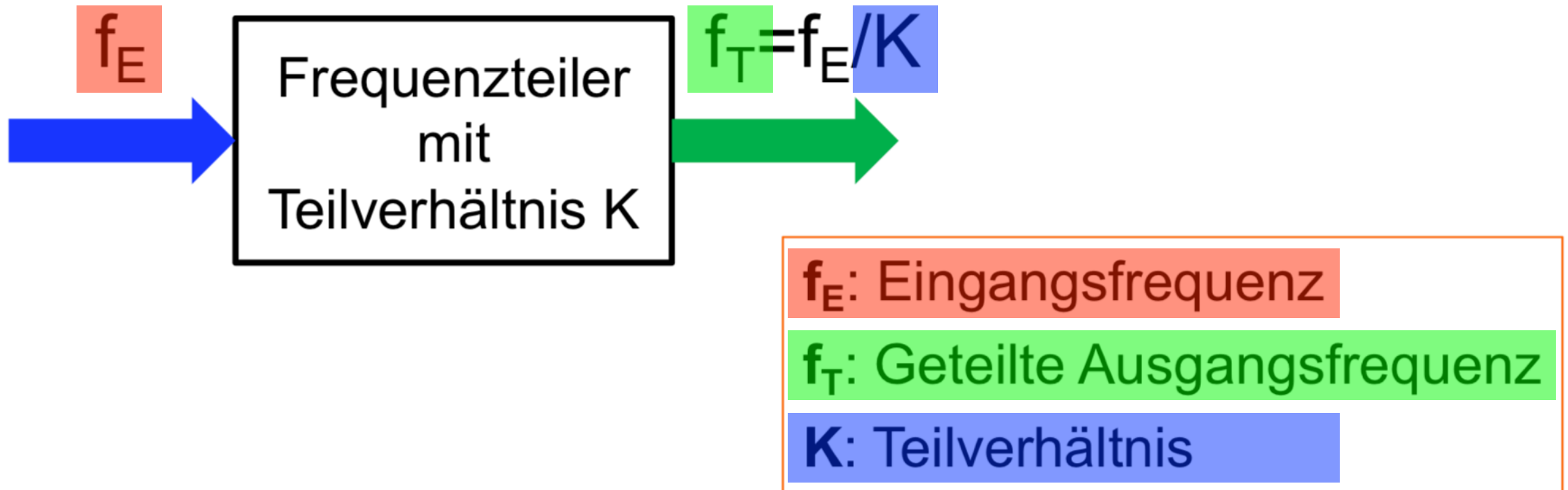
Synchronisierung der Eingänge

- Eingänge werden gespeichert und erst mit der nächsten Aktiven Taktflanke eingegeben



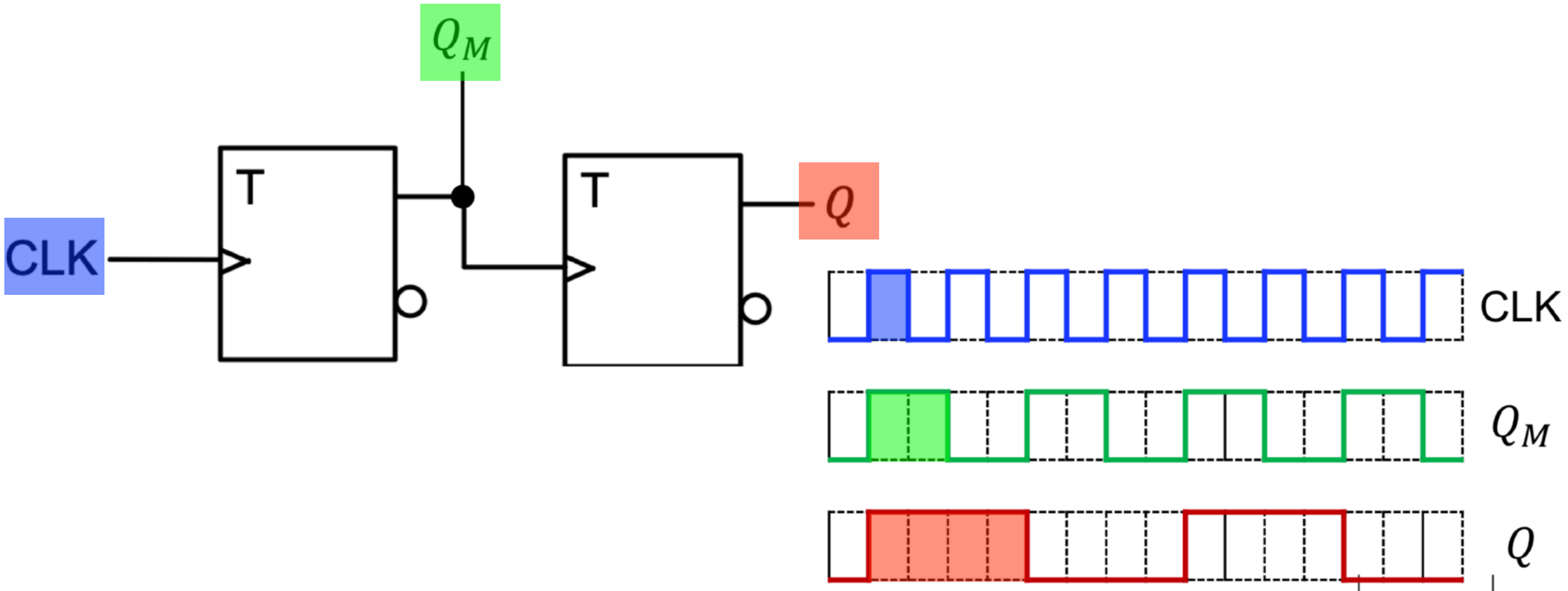
Frequenzteiler

- Zum reduzieren von Frequenzen



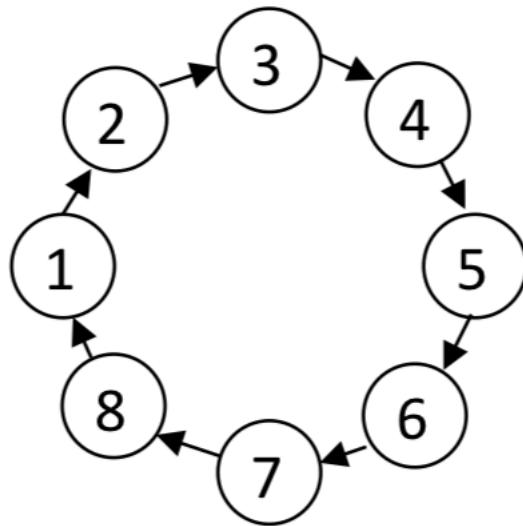
Flipflops als Frequenzteiler

- n Flipflops führen zu einer Reduktion von 2^n



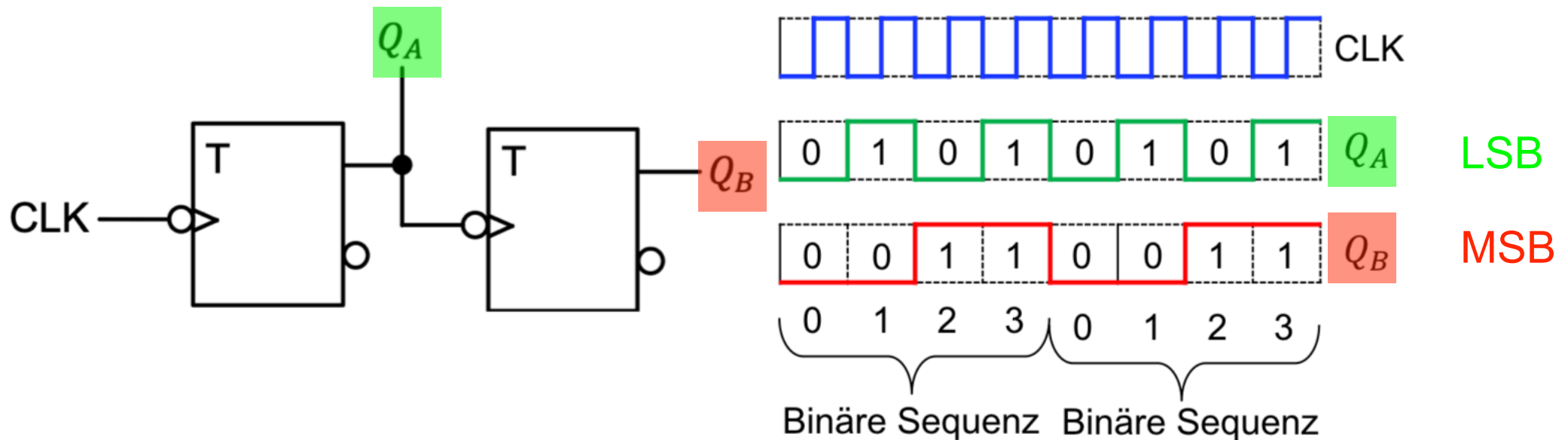
Zähler

- Zählen bis zu einer bestimmten Zahl und fangen dann wieder von vorne an
- Vorwärts zählen = +1 in jedem Schritt
- Rückwärts zählen = -1 in jedem Schritt



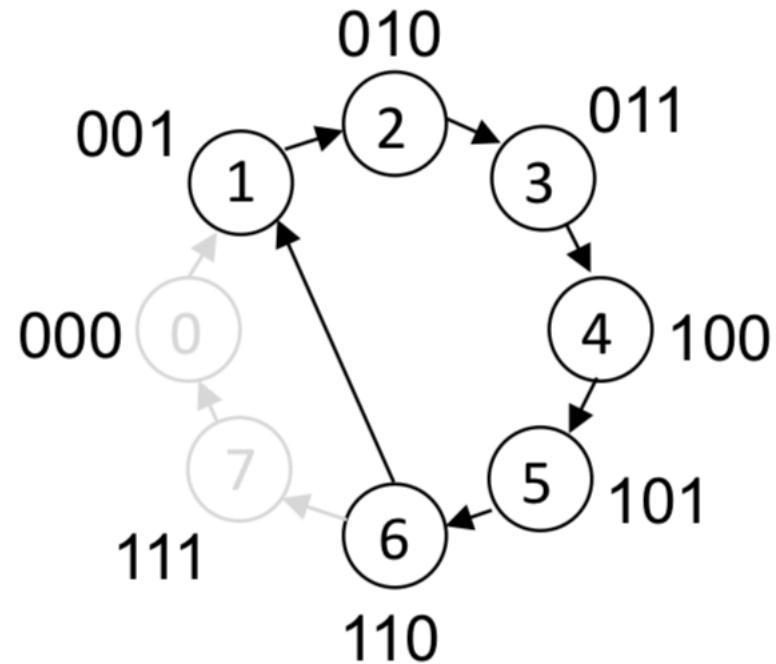
Dualzähler = Asynchrone Zähler

- Kaskadierung von T-Flipflops
- Mit n Flipflops kann man bis $2^n - 1$ zählen
- Einzelverzögerungen der Flipflops kumulieren sich \longrightarrow Asynchron
- Maximale Taktfrequenz: $f_{max} = \frac{1}{\sum_{i=1}^n t_{pd,i}}$



Modulo-n Zähler

- Zählt bis n und springt dann zurück auf einen vorgegebenen Startzustand
- Umsetzbar als Asynchronzähler und Synchronzähler



Modulo-n Synchronzähler

- Alle Eingänge liegen auf dem gleichen CLK (schalten gleichzeitig)
- Sind Medwedjew-Automaten
- Entwerfen wie Automaten

Zustandsgraph → Folgezustandstabelle → Karnaugh Diagramme → Schaltplan

Karnaugh Diagramm & Flipflops

D - Flipflops

- Karnaughdiagramm normal

JK - Flipflops

- Karnaughdiagramm mit Felder für Q und !Q

Q_{1n+1}

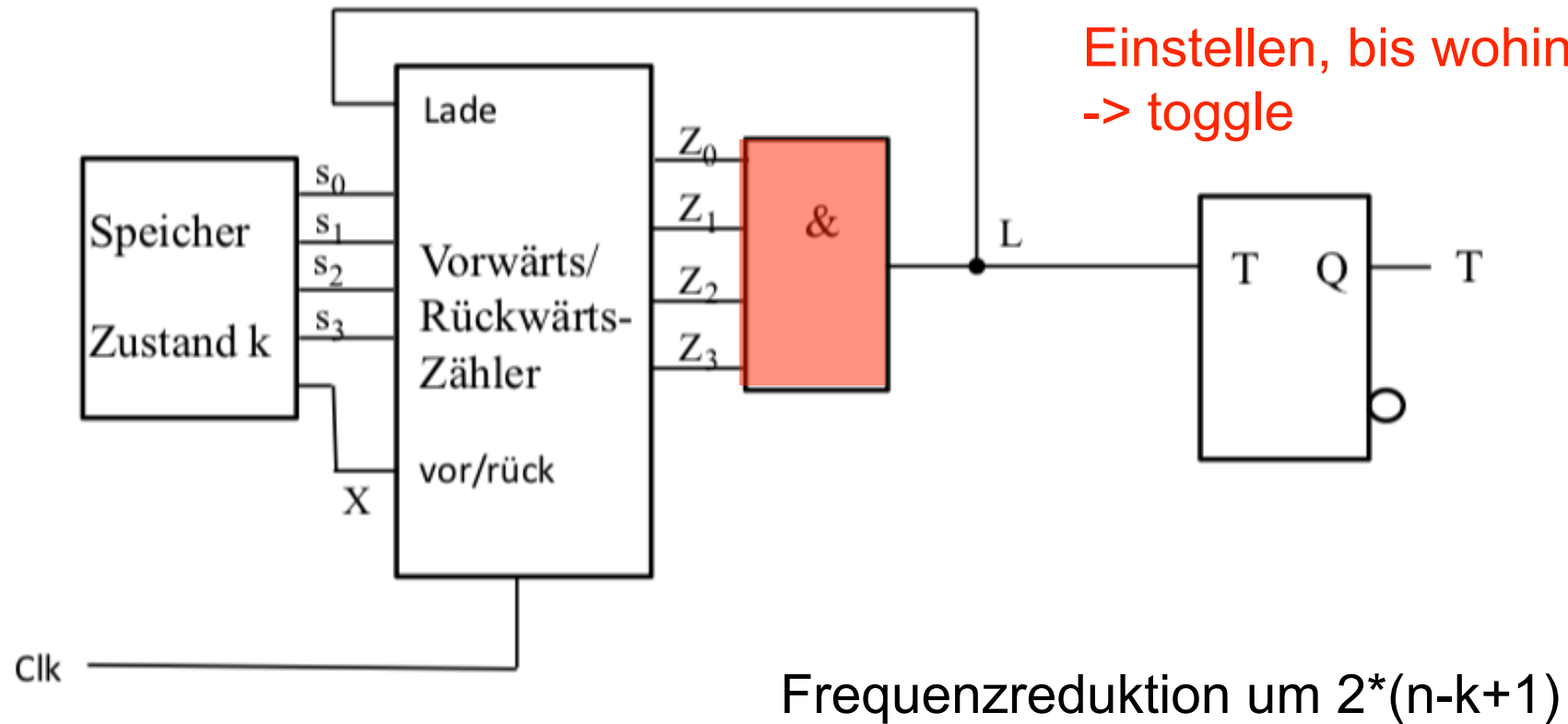
Q_3Q_2 Q_1	00	01	11	10
0				
1				

Q_3Q_2 Q_1	00	01	11	10
0				
1				

Q_{2n+1}

Frequenzteiler mit Zähler

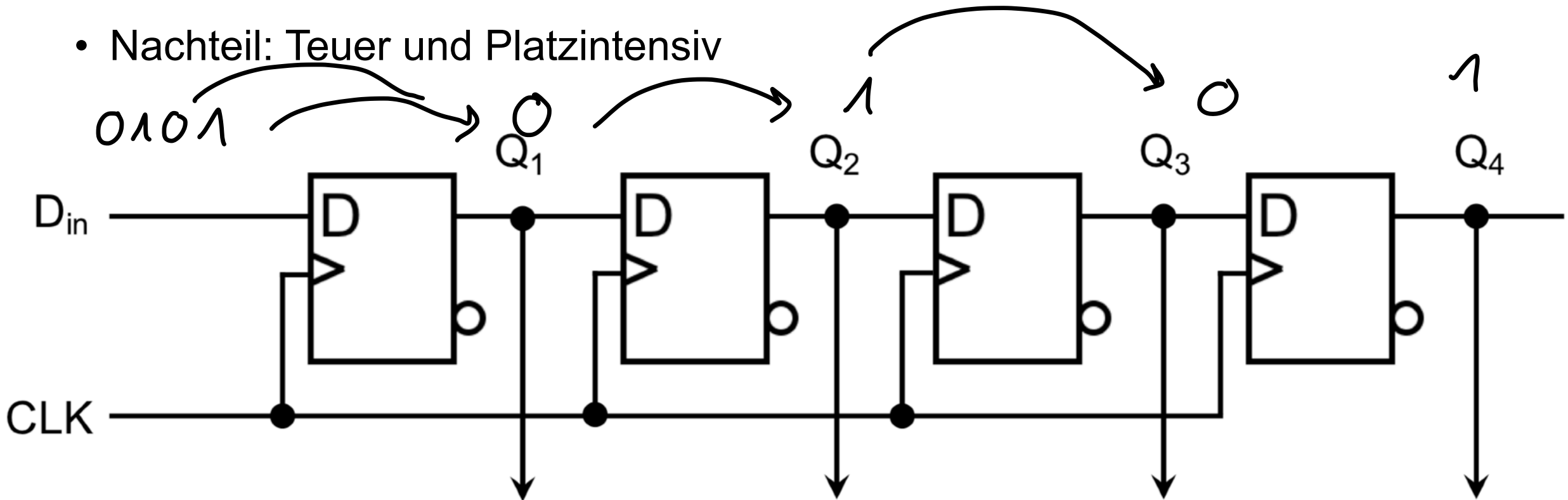
- Um Frequenz beliebig einzustellen



Speicher

Schieberegister

- n = Anzahl Flipflops
 - n Bits können gespeichert werden
 - n Taktflanken um Bits einzulesen
- Nachteil: Teuer und Platzintensiv



Speichermedien

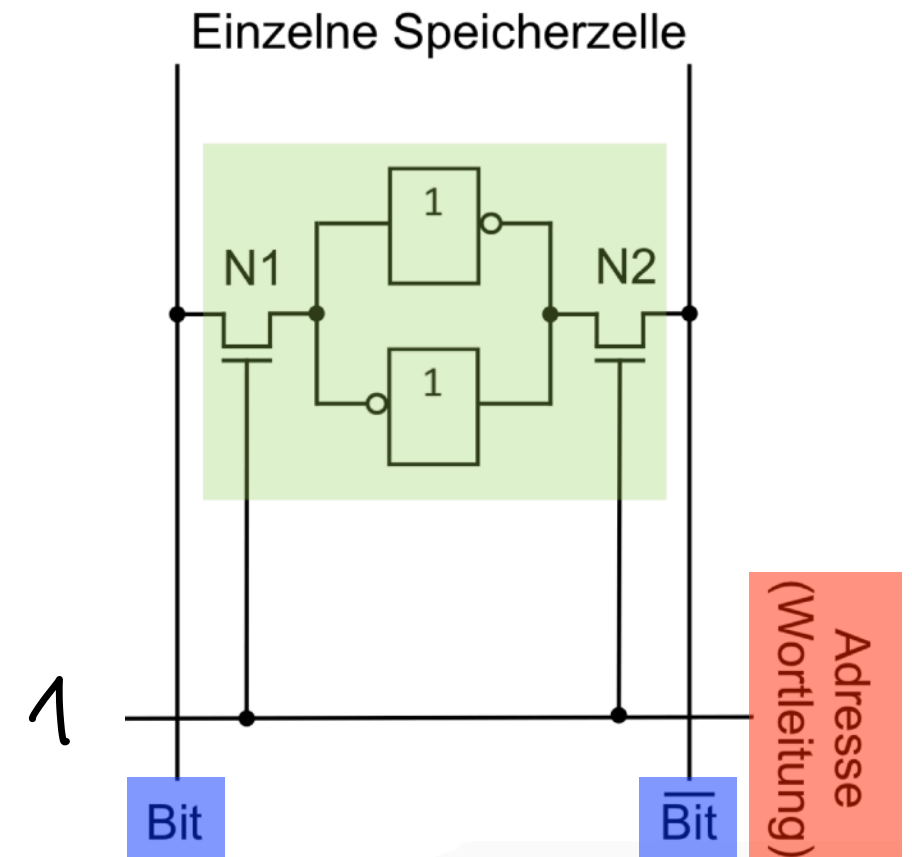
- Magnetband
 - Plattenlaufwerk
 - DRAM
 - Flash
-

Speicherfunktionen

- ROM = Read Only Memory
- RAM = Random Access Memory

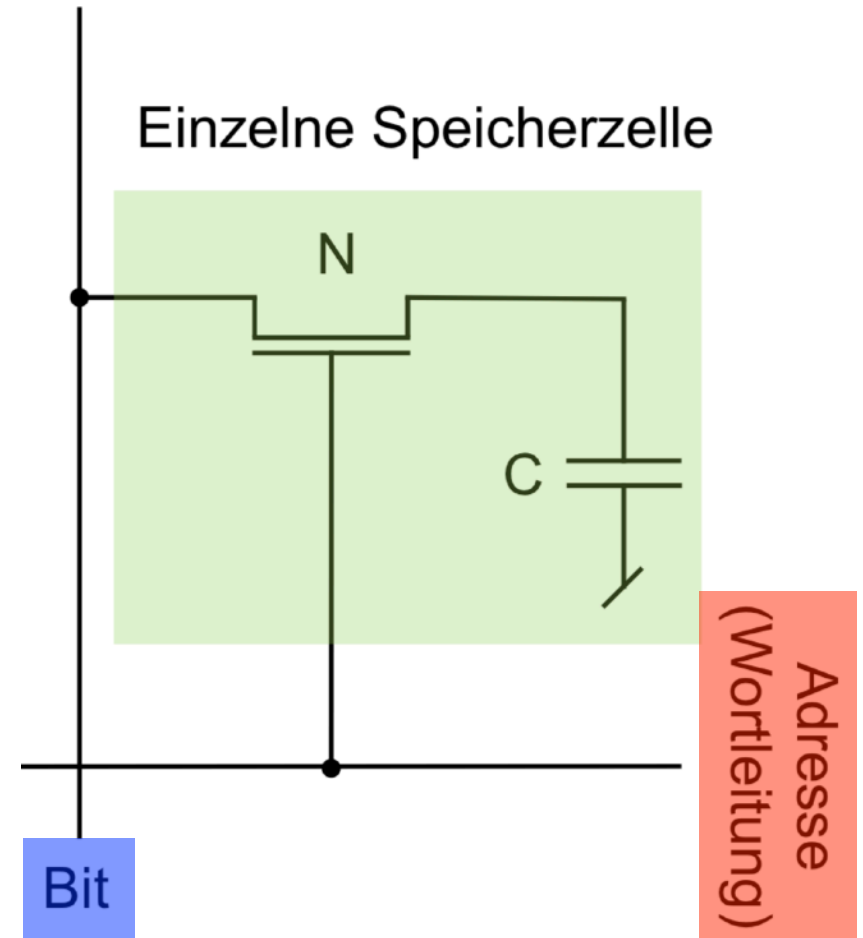
SRAM

- static random access memory
 - Speicherzelle wird angewählt
 - Speicherinhalt wird gesetzt/gelesen
-
- lesen: Bit = 1, !Bit = 1
 - 1 schreiben: Bit = 1, !Bit = 0
 - 0 schreiben: Bit = 0, !Bit = 1



DRAM

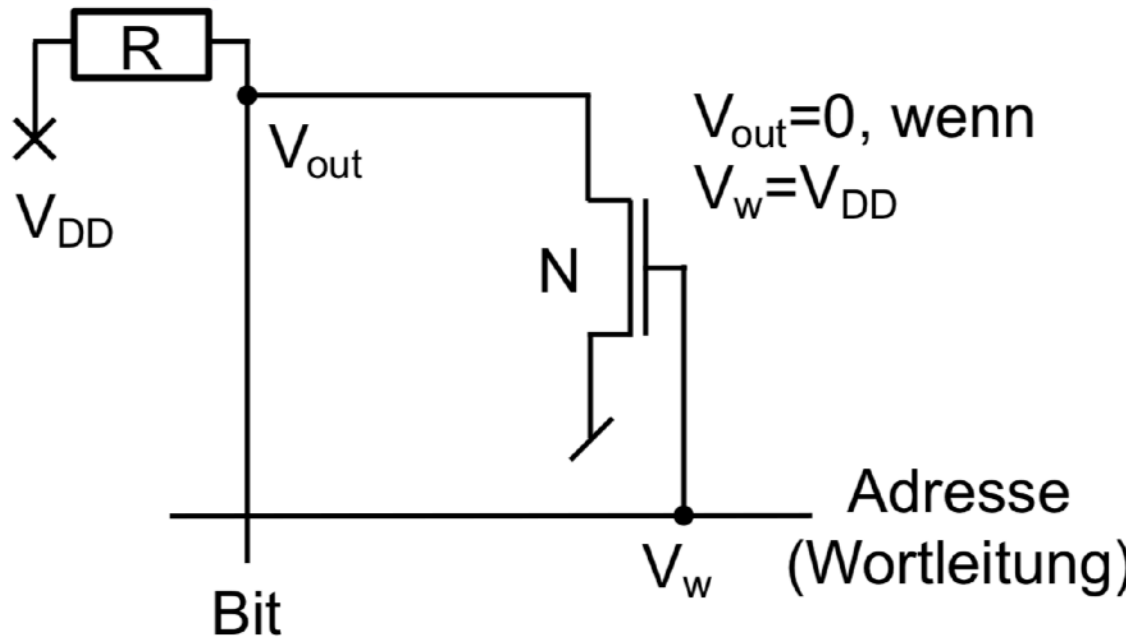
- dynamic random access memory
- Speicherzelle wird angewählt
- Speicherinhalt wird gesetzt/gelesen



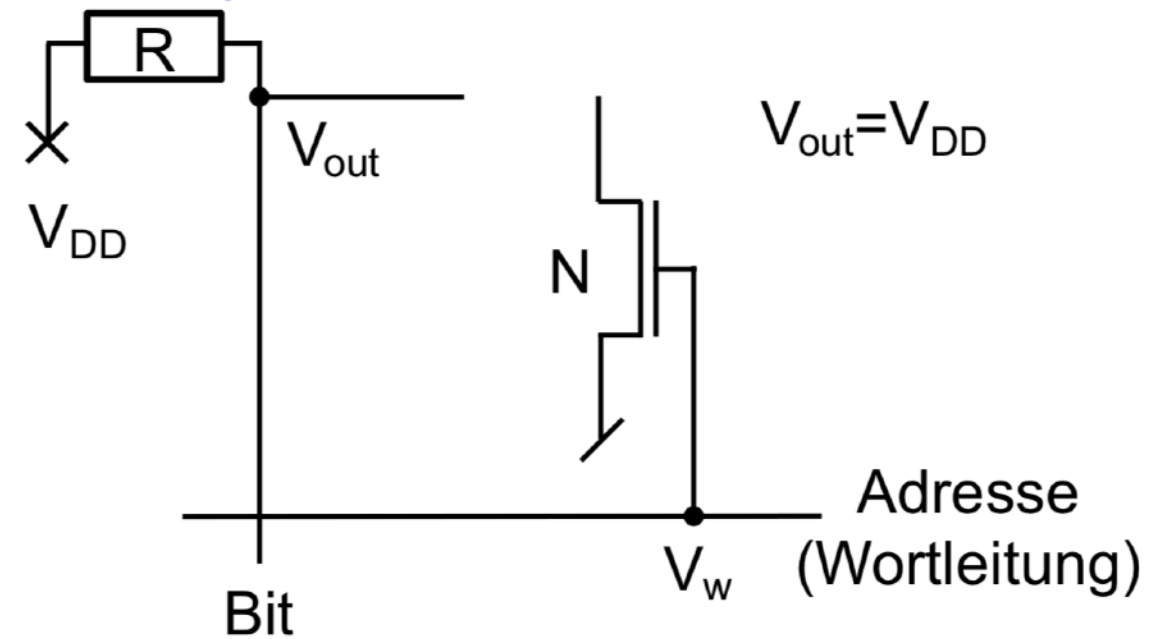
ROM

- Read only memory
- werden bei Produktion programmiert

Speicherzelle für eine 0

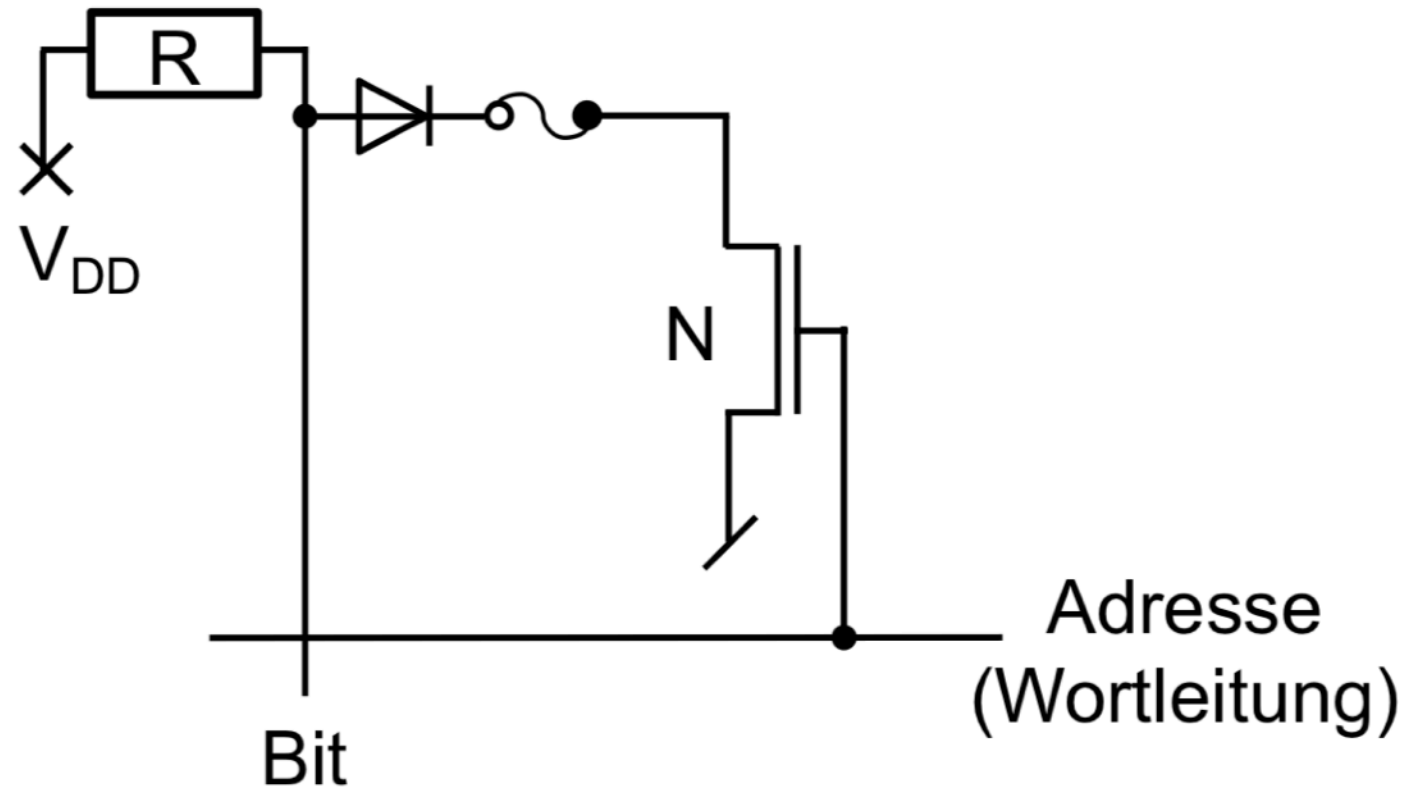


Speicherzelle für eine 1



PROM

- Programmierbare ROM
- Als 1 oder 0 programmierbar



Basisprüfung

Aufgabe A

Automaten (30 Punkte)

Die Aufgabe besteht aus **zwei unabhängigen** Teilaufgaben. In Teilaufgabe 1 geht es um das Erstellen des Zustandsgraphen eines Automaten, und in Teilaufgabe 2 soll das Verhalten eines durch ein Zustandsdiagramm beschriebenen Automaten analysiert werden.

Teilaufgabe 1: Entwurf eines Automaten

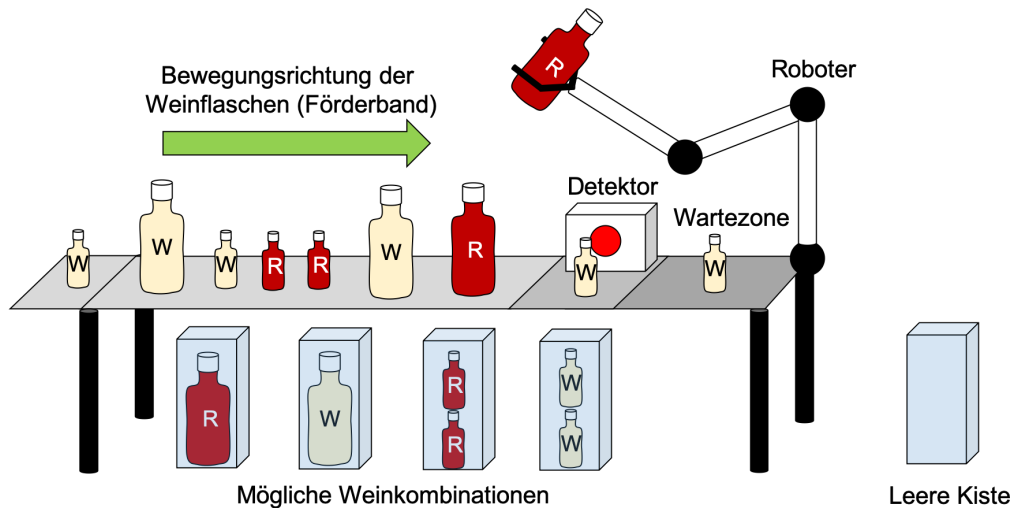


Abbildung A1: Anlage, um Weinflaschen in Kisten zu sortieren und zu packen.

Ein Winzer hat eine neue Anlage bestellt, um die Weinflaschen, die er auf seinem Weinberg produziert, in Kisten zu sortieren und zu packen. In seinem Keller wird **Rot- und Weisswein erzeugt**. Beide Weinsorten können entweder in **kleinen (0.5 Liter) oder in grossen (1 Liter) Flaschen** abgefüllt werden. Da alle Kisten die gleiche Grösse haben, werden **immer zwei kleine oder eine grosse Flasche(n) zusammengepackt**. **Rot- und Weisswein werden nie gemischt**. Dem Winzer wurde die Anlage in Abbildung A1 vorgeschlagen, der von Ihrer Firma verkauft wird. Als Ingenieur(in) haben Sie jetzt die Aufgabe, einen Automaten zu entwerfen, um diese Anlage zu steuern. Er besteht aus einem Detektor, einem Roboter, einer Wartezone und einer Kontrollstation.

Die Weinflaschen befinden sich zuerst auf einem Förderband und kommen vor dem Detektor vorbei, der die folgenden Signale generiert: **ND**, wenn nichts detektiert wird, **KR** für kleine Rotweinflaschen, **KW** für kleine Weissweinflaschen, **GR** für grosse Rotweinflaschen und **GW** für grosse Weissweinflaschen. Wenn der Automat eine grosse Flasche detektiert, schickt er dem Roboter ein **PF** Signal: **der Roboter packt diese grosse Flasche in eine Weinkiste**. Wenn eine kleine Flasche detektiert wird, prüft zuerst der Automat, ob es schon eine kleine Flasche der gleichen Farbe in der Wartezone gibt:

- wenn ja, dann empfängt der Roboter ein Signal **P2F** und **packt die beiden kleinen Flaschen der gleichen Farbe in eine einzige Kiste zusammen**. Der Roboter kann die Weinsorten automatisch unterscheiden.
- wenn nicht, dann wird die detektierte kleine Flasche vom Roboter mit dem **WZ** Befehl **in die Wartezone übertragen**.

Maximal können sich zwei kleine Weinflaschen von verschiedener Farbe gleichzeitig in der Wartezone befinden. Wenn der Roboter ein **MN** Signal kriegt, **macht er nichts und wartet** auf den nächsten Befehl.

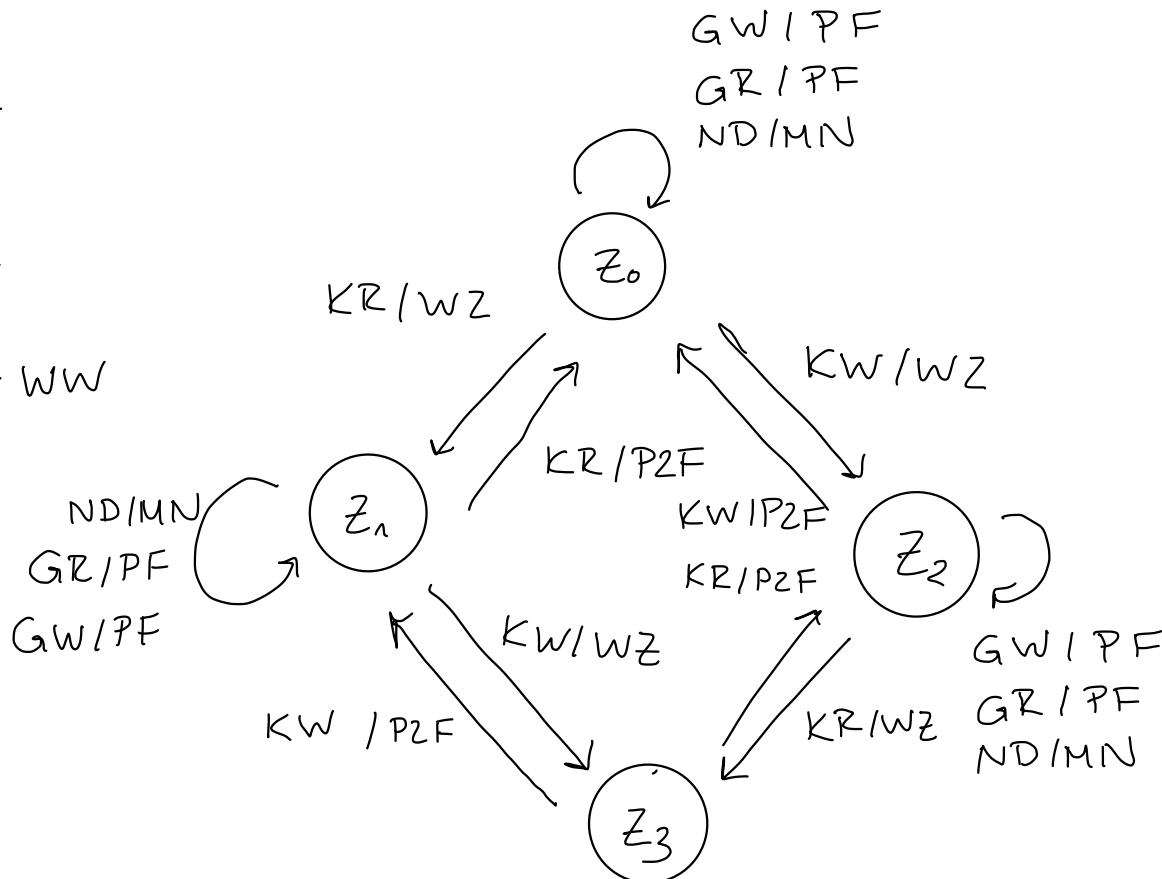
Aufgaben

1. Was sind die Eingänge und die Ausgänge dieses Automaten? Verwenden Sie die folgenden Variablen: GR, GW, KR, KW, MN, ND, PF, P2F und WZ. (2 Punkte)
2. Wie viele Bits brauchen Sie, um die Eingangs- und Ausgangsvariablen zu kodieren? (1 Punkt)
3. Zeichnen Sie das Zustandsdiagramm für diese Anlage als Mealy-Automat mit der minimalen Anzahl an Zuständen. Definieren Sie N Zustände Z_0 bis Z_{N-1} und beschreiben Sie deren Funktionalität ganz kurz. Als Eingangs- und Ausgangsvariablen benutzen Sie die Abkürzungen aus Aufgabe 1 (GR, GW, KR, ...). Hinweis: betrachten Sie den Zustand der Wartezone. (6 Punkte)
4. Wie viele Flipflops benötigen Sie minimal, um diesen Automaten zu realisieren? (1 Punkt)
5. Wie viele Zeilen und Spalten hätte die komplette Zustandsfolgetabelle? Begründen Sie. (2 Punkte)
6. Kann man dieses System auch als Moore-Automaten realisieren? Begründen Sie. (1 Punkt)

1. GR, GW, KR, KW, ND = Eingänge
 MN, P2F, PF, WZ = Ausgänge

2. 3 $\rightarrow 2^3 = 8$
 2 $\rightarrow 2^2 = 4$

3. Z_0 : leer
 Z_1 : RW
 Z_2 : WW
 Z_3 : RW + WW



4. $2^{FF} \rightarrow 2^2 = 4$

5. Zeilen: $2^{e+m} = 2^{3+2} = 2^5 = 32$

Spalten: $e + 2m + b = 3 + 2 \cdot 2 + 2 = 3 + 4 + 2 = 9$

$m = \text{Zustandsbits} = \#FF$

6. Ja.

Teilaufgabe 2: Analyse des Zustandsdiagramms eines Automaten

Bemerkung: Die Aufgaben in diesem Teil sind unabhängig von den Aufgaben in Teilaufgabe 1.

Ein Automat ist durch das Zustandsdiagramm in Abbildung A2 gegeben. Dabei bezeichnet $\{e_1, e_0\}$ die Eingänge, $\{y\}$ den Ausgang und $\{z_1, z_0\}$ die Zustände.

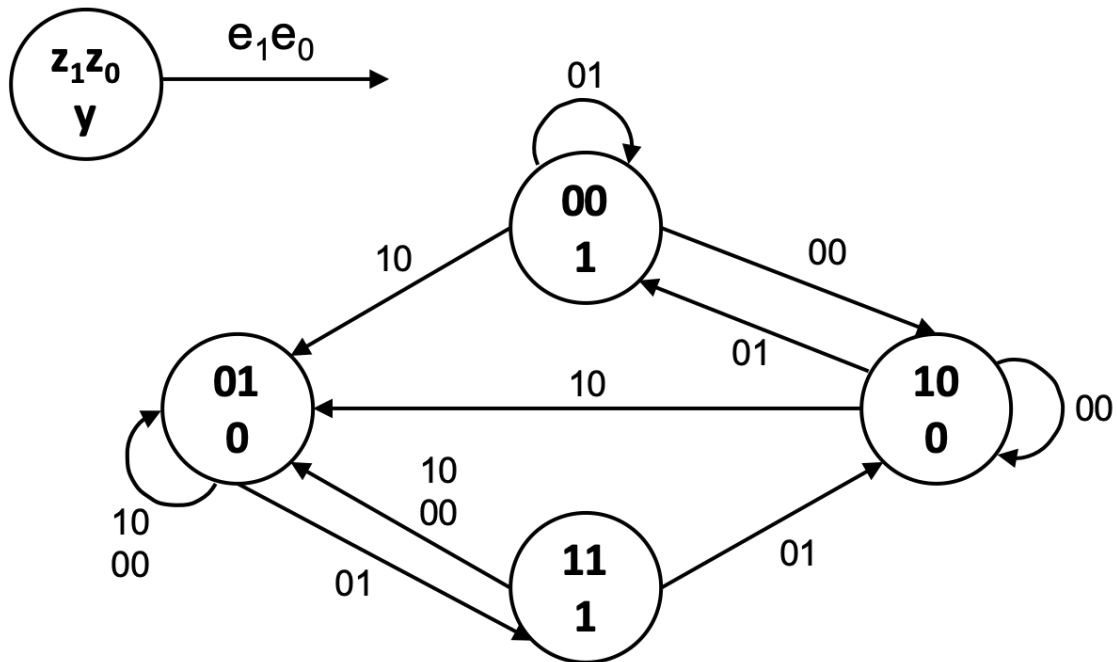


Abbildung A2: Zustandsdiagramm eines Automaten

Aufgaben

7. Um welchen Automatentyp handelt es sich? Begründen Sie Ihre Antwort. (1 Punkt)
8. Ergänzen Sie auf dem Lösungsblatt die Folgezustandstabelle des Automaten. (2 Punkte)
9. Erstellen Sie die Karnaugh-Diagramme für die Zustände $z_{1(n+1)}$ und $z_{0(n+1)}$ sowie für die Ausgangsvariable y_n auf dem Lösungsblatt. Geben Sie die minimale disjunktive Normalform (DNF) der Gleichungen für $z_{1(n+1)}$, $z_{0(n+1)}$ und y_n . (6 Punkte)
10. Zeichnen Sie den Schaltplan dieses Automaten auf dem Lösungsblatt. Ihnen stehen D-Flipflops sowie NOT, AND und OR Gatter mit einer beliebigen Anzahl an Eingängen zur Verfügung. (3 Punkte)

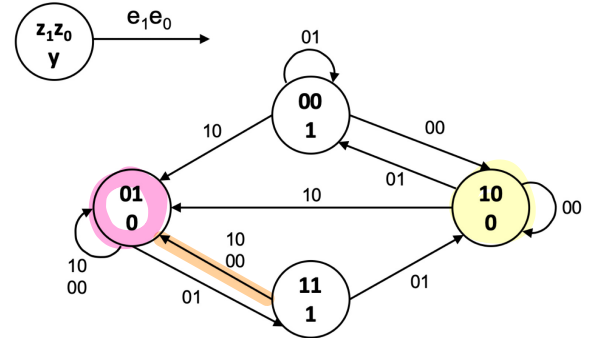
Eine (e_1, e_0) Eingangskombination ist bei diesem Automat unzulässig.

11. Welche ist diese Kombination? (1 Punkt)
12. Wenn diese Eingangskombination trotzdem passiert, sollte der Automat bei der nächsten aktiven Taktflanke in seinem aktuellen Zustand bleiben und in keinen anderen Zustand wechseln. Wie müssen die Gleichungen für $z_{1(n+1)}$ und $z_{0(n+1)}$ modifiziert werden, damit diese Bedingung erfüllt wird? Die zusätzlichen Karnaugh-Diagramme auf dem Lösungsblatt können dafür verwendet werden. (4 Punkte)

7. Moore. Ausgänge nur abhängig von I.Z.

zu 8)

e_1	e_0	z_{1n}	z_{0n}	z_{1n+1}	z_{0n+1}	y_n
0	0	0	0	1	0	1
0	0	0	1	0	1	0
0	0	1	0	1	0	0
0	0	1	1	0	1	1
0	1	0	0	0	0	1
0	1	0	1	1	1	0
0	1	1	0	0	0	0
0	1	1	1	1	0	1
1	0	0	0	0	1	1
1	0	0	1	0	1	0
1	0	1	0	0	1	1
1	0	1	1	0	1	1
1	1	0	0	X	X	1/X
1	1	0	1	X	X	0/X
1	1	1	0	X	X	0/X
1	1	1	1	X	X	1/X



z_{1n+1}

$e_1 e_0$	00	01	11	10
$\bar{z}_1 z_0$	1	0	X	0
$z_1 \bar{z}_0$	0	1	X	0
$z_1 z_0$	0	1	X	0
$\bar{z}_1 \bar{z}_0$	1	0	X	0

z_{0n+1}

$e_1 e_0$	00	01	11	10
$\bar{z}_1 z_0$	0	0	X	1
$z_1 \bar{z}_0$	1	1	X	1
$z_1 z_0$	1	0	X	1
$\bar{z}_1 \bar{z}_0$	0	0	X	1

$$z_{0n+1} = e_1 \vee (\bar{z}_1 \wedge z_0) \vee (\bar{e}_0 \wedge z_0)$$

$$z_{1n+1} = (\bar{e}_1 \wedge \bar{e}_0 \wedge \bar{z}_0)$$

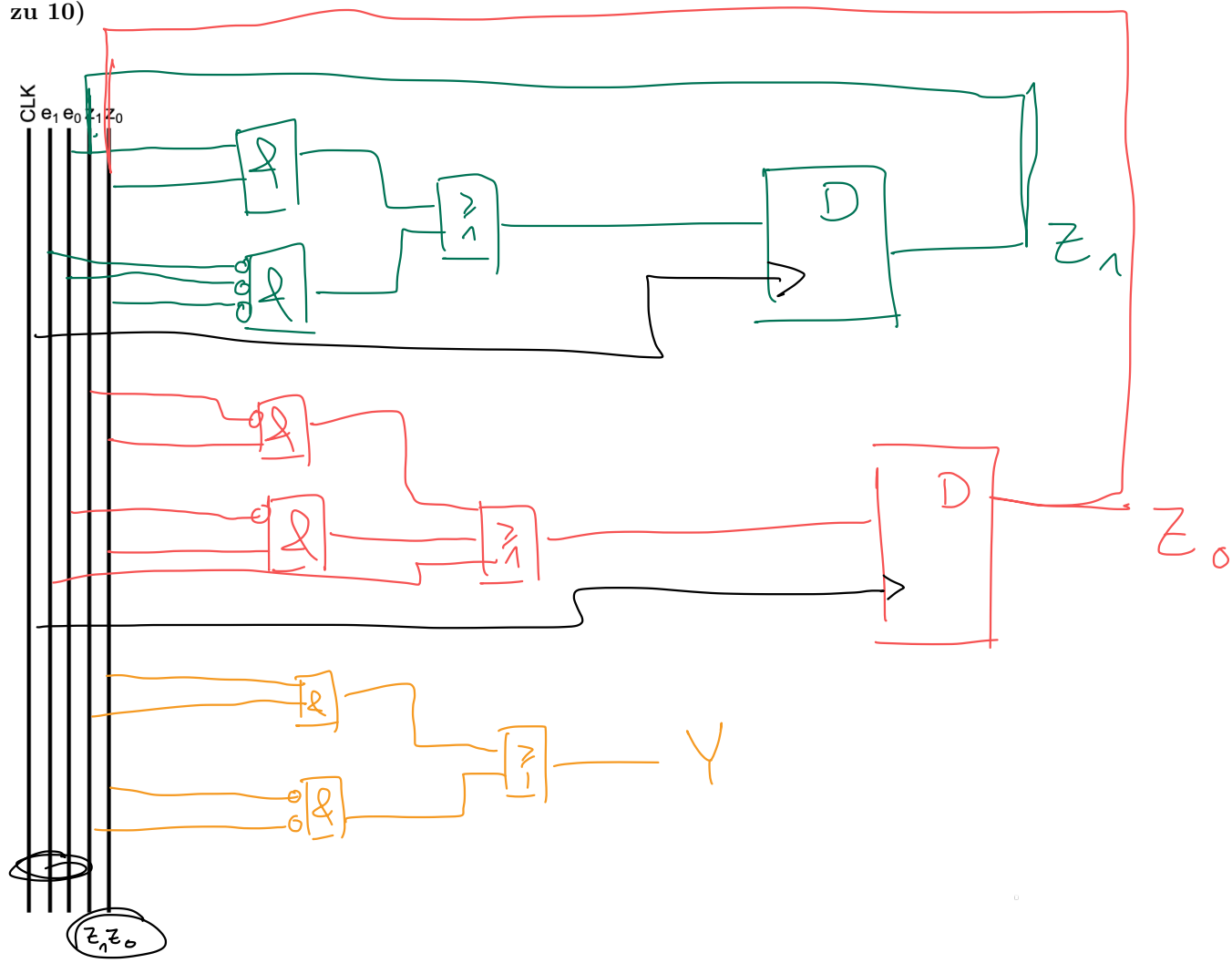
$$\vee (e_0 \wedge z_0)$$

y_n

$e_1 e_0$	00	01	11	10
$\bar{z}_1 z_0$	1	1	X	1
$z_1 \bar{z}_0$	0	0	X	0
$z_1 z_0$	1	1	X	1
$\bar{z}_1 \bar{z}_0$	0	0	X	0

$$y = (z_1 \wedge z_0) \vee (\bar{z}_1 \wedge \bar{z}_0)$$

zu 10)



$$z_{n+1} = (e_0 \wedge z_0) \vee (\bar{e}_n \wedge \bar{e}_0 \wedge \bar{z}_0)$$

$$z_0 = e_n \vee (\bar{z}_n \wedge z_0) \vee (\bar{e}_0 \wedge z_0)$$

$$Y = (\bar{z}_n \wedge \bar{z}_0) \vee (z_n \wedge z_0)$$

11. (1, 1)

Aufgabe B

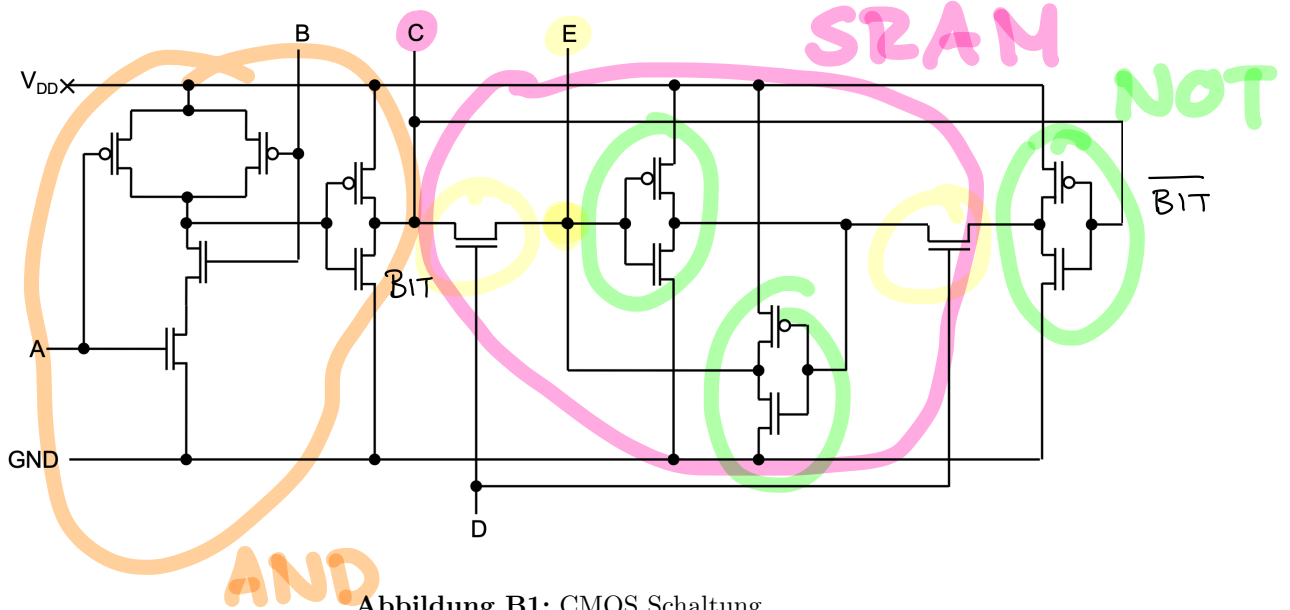
CMOS (16 Punkte)

In dieser Aufgabe sollen verschiedene Aspekte von CMOS Schaltungen untersucht werden.

Hinweis: Die Teilaufgaben 1 und 2 sind voneinander unabhängig.

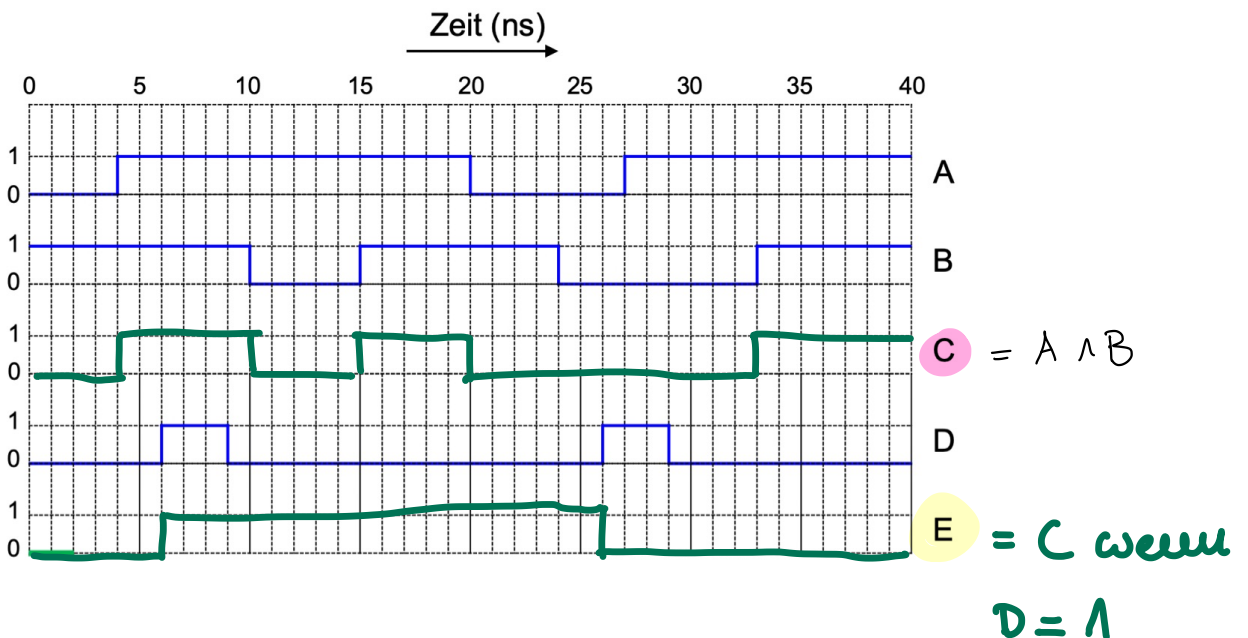
Teilaufgabe 1: Analyse einer CMOS Schaltung

In dieser Teilaufgabe soll die gegebene CMOS Schaltung in Abbildung B1 analysiert werden.



1. Markieren und bezeichnen Sie die CMOS Blöcke in der Schaltung auf dem Lösungsblatt. Wenn mehrere Blöcke gruppiert werden können, um ein einziges Bauelement zu realisieren, dann muss nur der gesamte Block markiert werden, z.B. NOR+INVERTER→OR (6 Punkte)
2. Vervollständigen Sie das Zeitdiagramm auf dem Lösungsblatt für die Variablen D und E als Funktion von A, B und C. Es wird angenommen, dass die Bauelemente keine Verzögerungszeiten haben. (4 Punkte)

2. Zeitdiagramm



Teilaufgabe 2: Pull-up/Pull-down Umwandlung

In dieser Teilaufgabe soll die Schaltung in Abbildung B2 analysiert werden. Damit wird der Pull-down Pfad einer CMOS Schaltung realisiert.

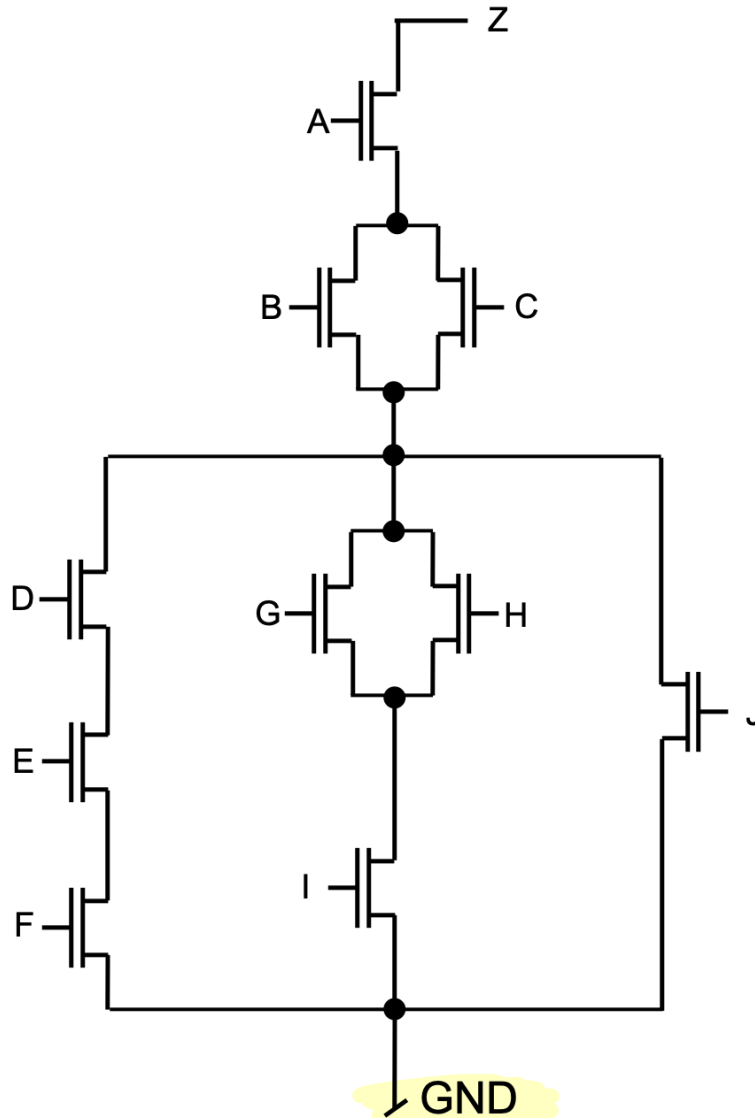


Abbildung B2: Pull-down CMOS Schaltung für die Funktion Z

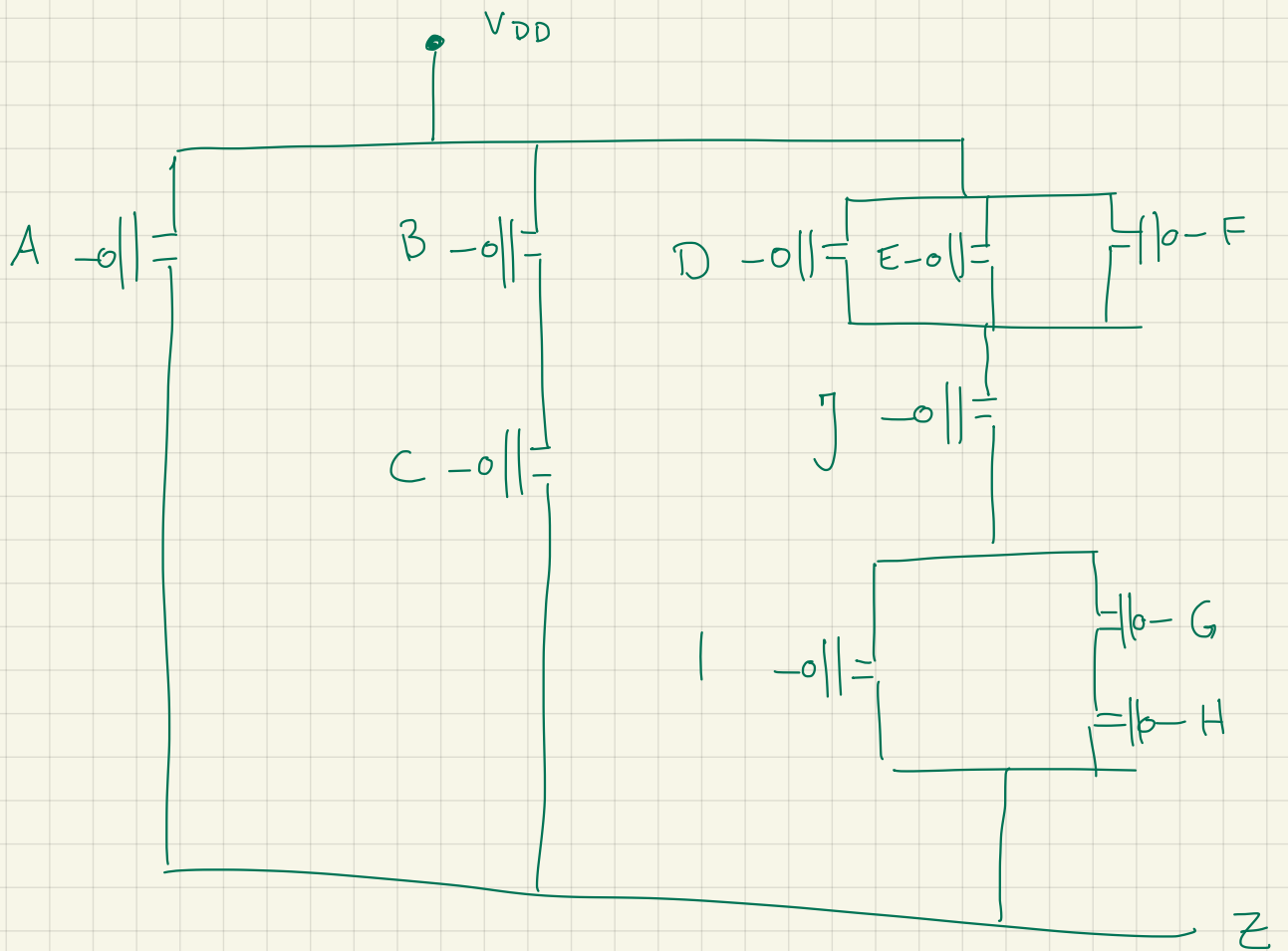
3. Zeichnen Sie den Pull-up Pfad der Schaltung in Abbildung B2 auf dem Lösungsblatt. (4 Punkte)
4. Geben Sie die Pull-up und Pull-down Funktionsgleichungen für Z an. (2 Punkte)

$$PD: A \wedge (B \vee C) \wedge ((D \wedge E \wedge F) \vee J \vee (I \wedge (G \vee H)))$$

$$PU: \bar{A} \vee (\bar{B} \wedge \bar{C}) \vee ((\bar{D} \vee \bar{E} \vee \bar{F} \wedge \bar{J} \wedge (\bar{I} \vee (\bar{G} \wedge \bar{H})))$$

$$PU : \bar{A} \vee (\bar{B} \wedge \bar{C}) \vee ((\bar{D} \vee \bar{E} \vee \bar{F}) \wedge \bar{J} \wedge (\bar{I} \vee (\bar{G} \wedge \bar{H})))$$

3.



Aufgabe C

Zahlensysteme, Rechnen und Minimieren (20 Punkte)

Die Aufgabe befasst sich im ersten Teil mit der Behandlung von Zahlen aus verschiedenen Zahlensystemen. **Achtung: Beachten Sie, dass die Teilaufgaben 1, 2 und 3 unabhängig voneinander zu lösen sind!**

Teilaufgabe 1: Umrechnung zwischen Zahlensystemen und Dualzahloperationen

1. Wandeln Sie -33_{10} in eine 8-stellige Zweierkomplementzahl um. (1 Punkt)
2. Wandeln Sie die 8-stellige Zweierkomplementzahl 10001000_2 in die entsprechende ganzzahlige Dezimalzahl um. (1 Punkt)
3. Wandeln Sie -7.3_{10} so genau wie möglich in eine Zweierkomplementzahl mit 4 Bits vor und 4 Bits nach dem Komma um. Was ist der Absolutbetrag des resultierenden Dezimalzahl Fehlers? (2 Punkte)
4. Wandeln Sie die folgende positive Binärzahl 11111001111.101_2 in die entsprechende Hexadezimalzahl um. Bitte merken Sie sich, dass es keine Zweierkomplementzahl ist. (1 Punkt)
5. Zwei positive Binärzahlen $A=11100.1_2$ und $B=11101.001_2$ sind gegeben. Gesucht ist $C=A-B$. Geben Sie A, -B und C als Zweierkomplementzahlen (6 Bits vor dem Komma, 4 nach dem Komma) sowie C als Dezimalzahl an. (3 Punkte)

$$\begin{array}{r}
 1. \quad 00100001 = 33 \\
 \hline
 11011110 \\
 + \quad \quad \quad 1 \\
 \hline
 11011111 = -33
 \end{array}$$

$$\begin{array}{l}
 2. \quad 7,3 \\
 7 = 0111 \\
 0,3 = (0,0101)_2 = \frac{1}{4} + \frac{1}{16} \\
 0,6 = 0,3125 \\
 1,2 \rightarrow 0,2 \\
 0,4 \\
 0,8 \\
 \text{Fehler} = 0,0125
 \end{array}$$

$$\begin{array}{r}
 2. \quad 10001000 \\
 \hline
 01110111 \\
 + \quad \quad \quad 1 \\
 \hline
 01111000 \\
 \begin{array}{cccc}
 2^6 & 2^5 & 2^4 & 2^3 \\
 | & & & \\
 2^3 & & &
 \end{array} \\
 \hline
 2^6 + 2^5 + 2^4 + 2^3 = 120 \\
 \hline
 -2^7 + 2^3 = -120
 \end{array}$$

$$\begin{array}{r}
 7,3 \approx 0111,0101 \\
 \hline
 1000,1010 \\
 + \quad \quad \quad 1 \\
 \hline
 1000,1011 \approx -7,3
 \end{array}$$

$$4. \quad \underbrace{01111000}_{7} \underbrace{1111}_{12}, \underbrace{1010}_{10} = 7CF, A_{16}$$

$\begin{matrix} C & F & A \end{matrix}$

$$\begin{aligned} 10 &= A \\ 11 &= B \\ 12 &= C \\ &\vdots \end{aligned}$$

$$5. \quad 01100, 1000 = A$$

$$\begin{array}{r} B = \quad 01101, 0010 \\ \hline \quad 100010, 1101 \\ + \quad \quad \quad 1 \\ \hline \quad 100010, 1110 = -B \end{array}$$

$$\begin{array}{r} \quad 011100, 1000 \\ + \quad 100010, 1110 \\ \hline C = 111111, 0110 \end{array}$$

Teilaufgabe 2: Logik Minimierung mit Karnaugh-Diagramm

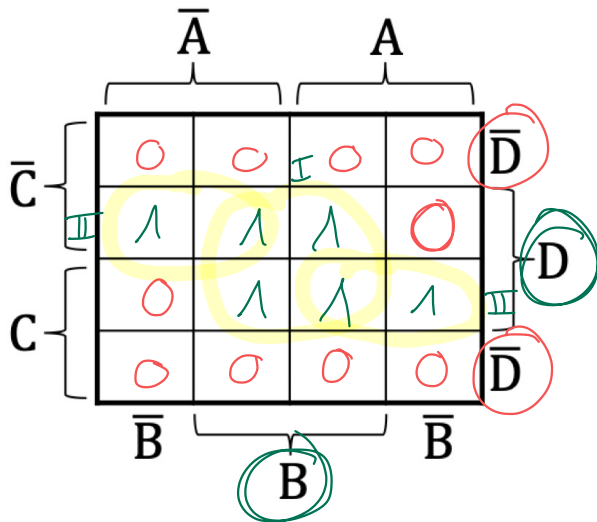
In dieser Aufgabe leiten Sie die minimale disjunktive Normalform (DNF) einer gegebenen Funktion Z mittels Karnaugh-Diagramm her.

Eine kombinatorische Schaltung hat vier Eingänge A, B, C und D und einen Ausgang Z . Der Ausgang ist durch die folgende Funktion beschrieben:

$$Z = \overline{(\bar{B} \vee \bar{D}) \wedge ((\bar{D} \vee (A \wedge \bar{C})) \vee (\bar{A} \wedge C))} \leftarrow$$

Aufgaben

6. Füllen Sie das Karnaugh-Diagramm für Z auf dem Lösungsblatt aus und identifizieren Sie alle Päckchen, die aus Mintermen bestehen. (4 Punkte)
7. Geben Sie die minimale DNF (disjunktive Normalform) von Z als Gleichung an. (2 Punkte)
8. Gibt es Hazards? Wenn ja, bei welchem Übergang (welchen Übergängen)? (1 Punkt)



$$\begin{aligned}
 z &= \overline{(\bar{B} \vee \bar{D}) \wedge ((\bar{D} \vee (A \wedge \bar{C})) \vee (\bar{A} \wedge C))} \\
 &= (\bar{B} \wedge D) \vee \overline{((\bar{D} \vee (A \wedge \bar{C})) \wedge (\bar{A} \wedge C))} \\
 &= (\bar{B} \wedge D) \vee \overline{(\bar{D} \wedge (A \wedge \bar{C})) \wedge (\bar{A} \wedge C)} \\
 &= (\bar{B} \wedge D) \vee \overline{(\bar{D} \wedge (A \wedge \bar{C}))} \wedge \overline{(\bar{A} \wedge C)} \\
 &= (\bar{B} \wedge D) \vee (D \wedge (A \wedge \bar{C})) \wedge (A \vee C)
 \end{aligned}$$

$B=1 \ \& \ D=1$

$D=0$

$$\bar{A} \vee C = 0 \Rightarrow A = 1, C = 0$$

$$A \vee \bar{C} = 0 \Rightarrow A = 0, C = 1$$

$$7. (\bar{B} \wedge D) \vee (\bar{A} \wedge \bar{C} \wedge D) \vee (C \wedge A \wedge D)$$

Teilaufgabe 3: Logik Minimierung mit Bool'scher Algebra

In dieser Aufgabe müssen zwei Gleichungen anhand der Regel der Bool'schen Algebra minimiert werden.

Aufgaben

9. Vereinfachen Sie die folgende logische Gleichung (2 Punkte)

$$\overline{(A \wedge \bar{B}) \vee (\bar{A} \wedge B)} \wedge (A \vee B)$$

10. Vereinfachen Sie die folgende logische Gleichung (3 Punkte)

$$((A \wedge \bar{B} \wedge (C \vee (B \wedge D))) \vee \bar{A} \vee B) \wedge \bar{C}$$

$$\begin{aligned}
 9. \quad & \overline{(A \wedge \bar{B}) \vee (\bar{A} \wedge B)} \wedge (A \vee B) = (\overline{A \wedge \bar{B}}) \wedge (\overline{\bar{A} \wedge B}) \wedge (A \vee B) \\
 & = (\bar{A} \vee B) \wedge (A \vee \bar{B}) \wedge (A \vee B) = (\bar{A} \vee B) \wedge (A \vee (\bar{B} \wedge B)) \\
 & = (\bar{A} \vee B) \wedge (A \vee 0) = (\bar{A} \vee B) \wedge A = \underline{\underline{A \wedge B}}
 \end{aligned}$$

$$10. ((A \wedge \bar{B} \wedge (C \vee (B \wedge D))) \vee \bar{A} \vee B) \wedge \bar{C}$$

$$\begin{aligned}
 & = ((A \wedge \bar{B} \wedge C) \vee (A \wedge \bar{B} \wedge B \wedge D)) \vee \bar{A} \vee B) \wedge \bar{C} \\
 & \quad \underbrace{\hspace{10em}}_0 \\
 & = ((A \wedge \bar{B} \wedge C) \vee \bar{A} \vee B) \wedge \bar{C} = \bar{C} \wedge ((\bar{A} \vee A) \wedge (\bar{A} \vee \bar{B}) \wedge (\bar{A} \vee C)) \vee B) \\
 & = \bar{C} \wedge (B \vee ((\bar{A} \vee \bar{B}) \wedge (\bar{A} \vee C))) = \bar{C} \wedge (B \vee (\bar{A} \vee \bar{B}) \wedge (B \vee \bar{A} \vee C)) \\
 & \quad \underbrace{\hspace{10em}}_0 \\
 & = \bar{C} \wedge (\bar{A} \wedge (B \vee \bar{A} \vee C)) \\
 & = (\bar{A} \vee B) \wedge \bar{C}
 \end{aligned}$$

Aufgabe D

Sequenzielle Schaltungen (20 Punkte)

Die Aufgabe besteht aus **drei unabhängigen** Teilaufgaben.

Teilaufgabe 1: Latches und Flipflops



D-M-S-FF

steigende Flanke
aktiv

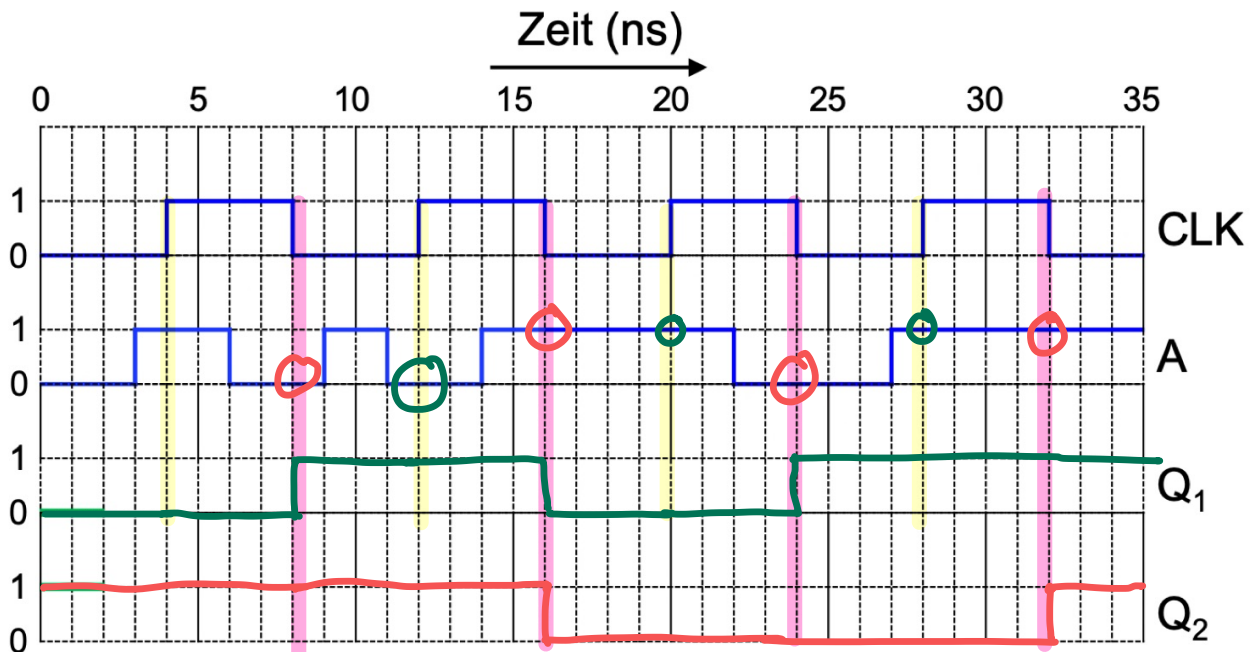
Abbildung D1: Bauelemente

T-FF

fallende Flanke
aktiv

Aufgaben

1. Benennen Sie die Bauelemente aus Abbildung D1. Beschreiben Sie auch, wie sie vom CLK Signal gesteuert werden. (2 Punkte)
2. Vervollständigen Sie das Zeitdiagramm auf dem Lösungsblatt für die Bauelemente aus Abbildung D1. Es wird angenommen, dass diese Bauelemente keine Verzögerungszeiten haben. (4 Punkte)



$$T_{min} = t_{pd,ff1} + t_{pd,s} + t_{setup,ff2}$$

$$T_{min} = t_{pd,ff} + t_{setup,ff} + t_{pd,inv} + 2t_{pd,AND} + t_{pd,or} = 1/50\text{MHz}$$

$$T_{min} = t_{pd,ff} + t_{setup,ff} + 3t_{pd,AND} + t_{pd,or} + t_{pd,inv} = 1/40\text{MHz}$$

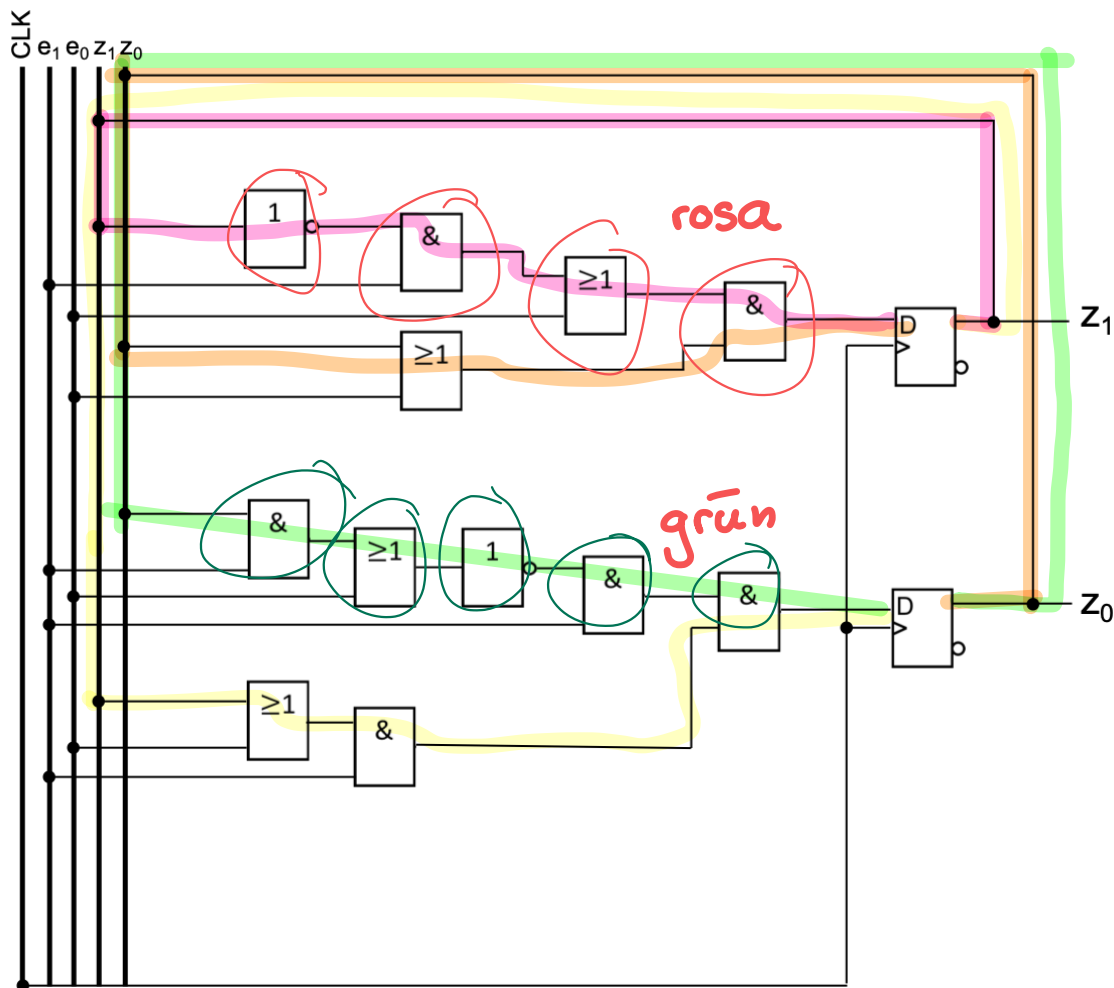


Abbildung D2: Schaltplan eines Automaten

Teilaufgabe 2: Taktfrequenzen von Schaltungen

Die folgende Aufgabe bezieht sich auf die Automaten-schaltung aus Abbildung D2. Es wurde festgestellt, dass der Ausgang z_1 mit einer maximalen Taktfrequenz von 50 MHz betrieben werden kann, der Ausgang z_0 mit maximal 40 MHz. Die Durchlaufzeiten aller Bauelemente sowie die Setup- und Hold-Zeiten der Flipflops sind der Tabelle D1 zu entnehmen, ausser die des UND- und ODER-Gatters, die unbekannt sind. Nehmen Sie an, dass alle Signalpegel der Eingänge e_1 und e_0 rechtzeitig stabil anlagen, als die maximalen Taktfrequenzen von z_1 und z_0 gemessen wurde.

Aufgaben

- Zeichnen Sie auf dem Lösungsblatt den längsten Signalpfad für z_1 und z_0 . (2 Punkte)
- Geben Sie für diese beiden Signalpfade die Gleichung (Bedingung) an, die die entsprechende minimale Taktperiode bestimmt. Verwenden Sie dafür die Variablen-namen aus Tabelle D1. (2 Punkte)
- Aus der gemessenen maximalen Taktfrequenz von z_1 und z_0 können die Durchlaufzeiten $t_{d,or}$ und $t_{d,and}$ des ODER und UND Gatters gerechnet werden. Wie gross sind sie? (3 Punkte)

D-Flipflop Setup-Zeit	$t_{setup, dff}$	2 ns
D-Flipflop Durchlaufzeit	$t_{pd, dff}$	4 ns
D-Flipflop Hold-Zeit	$t_{hold, dff}$	2 ns
NOT Durchlaufzeit	$t_{d, not}$	1 ns
AND Durchlaufzeit	$t_{d, and}$?? ns
OR Durchlaufzeit	$t_{d, or}$?? ns

Tabelle D1: Zeitparameter

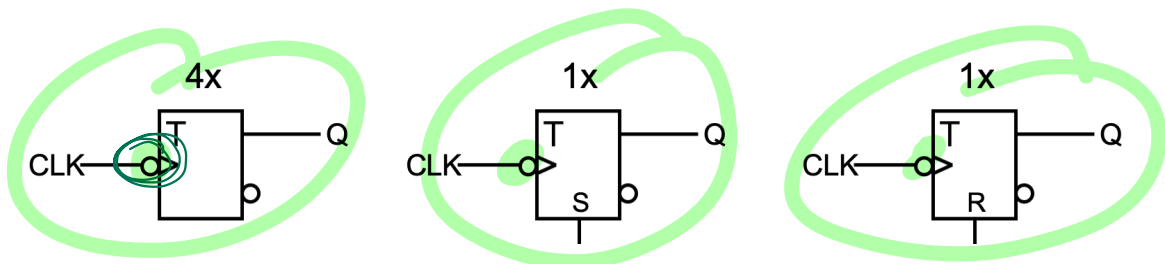


Abbildung D3: (links) Normale T-Flipflops. (mitte) T-Flipflop mit einem asynchronen Setz (S) Eingang. (rechts) T-Flipflop mit einem asynchronen Rücksetz (R) Eingang.

Teilaufgabe 3: Asynchronzähler mit T-Flipflops

Das Ziel dieser Teilaufgabe ist, einen asynchronen Binärzähler zu entwerfen, der in der Regel von 0 bis 15 zählt ausser wenn ein Eingangssignal $IN=1$ ist. In diesem Fall muss er bei 5 neu starten, sobald er 12 erreicht. Alle Zahlen sind als Dualzahlen kodiert. Ihnen stehen 1 T-Flipflop mit einem asynchronen Setz (S) Eingang, 1 T-Flipflop mit einem asynchronen Rücksetz (R) Eingang sowie 4 normale T-Flipflops ohne zusätzliche asynchrone Eingänge zur Verfügung. Diese Bauelemente sind in Abbildung D3 gezeigt.

Beim T-Flipflop mit dem asynchronen Setz Eingang wird der Ausgang Q sofort 1 wenn $S=1$, unabhängig vom Taktsignal. Beim T-Flipflop mit dem asynchronen Rücksetz Eingang wird der Ausgang Q sofort 0 wenn $R=1$, auch unabhängig vom Taktsignal. Wenn der zu entwerfende Zähler 12 erreicht und $IN=1$, dann muss er sofort bei 5 neu anfangen. Sonst ($IN=0$) zählt er von 0 bis 15 und fängt dann wieder bei 0 an.

Wenn Sie N Flipflops benötigen, um diesen Zähler aufzubauen, dann sollten die Flipflop Ausgänge Q_0 bis Q_{N-1} benannt werden, wobei Q_0 dem 'least significant bit' entspricht und Q_{N-1} dem 'most significant bit'.

6. Wie viele T Flipflops sind nötig, um diesen Zähler zu realisieren? (1 Punkt)
7. Wie viele nicht verwendete Zustände ergeben sich daraus? Begründen Sie Ihre Aussage. (1 Punkt)
8. Was ist die logische Bedingung, damit der Zähler in einen anderen Zustand wechselt, wenn er 12 erreicht? Drücken Sie diese Bedingung als Funktion von den Q -Ausgängen und vom IN Signal aus. (1 Punkt)
9. Zeichnen Sie diesen Zähler auf dem Lösungsblatt. Benützen Sie die T-Flipflops aus Abbildung D3 sowie NICHT, UND und ODER Gatter mit einer beliebigen Anzahl an Eingängen. (4 Punkte)

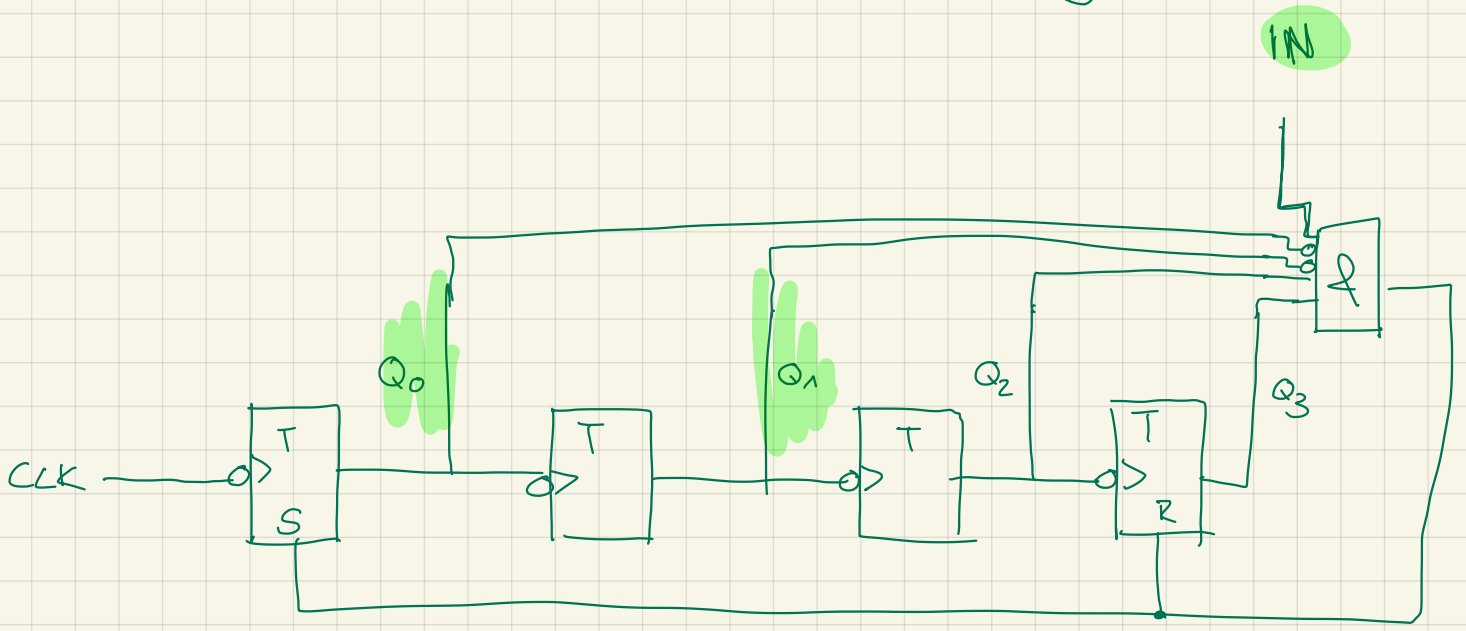
6. $16 \rightarrow 4$

7. 0 $4 \text{ FF} \rightarrow 2^n = 16 \text{ Zustände}$

8. $12 = 1100$ $IN \wedge \overline{Q_0} \wedge \overline{Q_1} \wedge Q_2 \wedge Q_3 = 1$
 $Q_3 \quad Q_0$

$$2^n - 1 \Big|_{n=4} = 15$$

$Q_3 \quad Q_1$
 $S = 0101$
 $R = 1100$
 $12 = 1100$
 $(1) || 1$



$12 = 1100$
 $S = 0101$
 $Q_3 \quad Q_0$

0000
 0001
 0010
 ...
 1111 → 0000
 15

Basisprüfung D-ITET

Digitaltechnik

27. Januar 2020

Hinweise zu den Aufgaben:

- Punktezahl pro Aufgabe:

Bitte leer lassen!

	Punkte	Visum
A:	30	
B:	16	
C:	20	
D:	20	
Total:	86	

- Schreiben Sie auf jedes Blatt Ihren Kode, der Ihnen auf dem Deckblatt zugeteilt worden ist.
- Beantworten Sie nur gestellte Fragen.
- Sollten Unklarheiten auftreten, wenden Sie sich an die Betreuer.
- Die Aufgaben können in beliebiger Reihenfolge gelöst werden.
- Resultate können nur bewertet werden, wenn ihre Herleitung bzw. Begründung in der angegebenen Lösung enthalten ist.
- Folgefehler werden berücksichtigt.
- **Benutzen Sie für die Lösungen ausschliesslich die vorgedruckten Lösungsblätter.** Sie können jederzeit neue Blätter bei einem Betreuer verlangen. Beachten Sie, dass auf jedem Lösungsblatt steht, für welche Aufgabe es gedacht ist.
- Benutzen Sie einen dokumentenechten Schreiber, d.h. Kugelschreiber, Füllfederhalter o.ä. Bitte fertigen Sie die Prüfung **nicht** mit Bleistift an, **keine rote Farbe!**
- Es sind **keine Kommunikationsmittel** und **keine Rechner** erlaubt.
- Beschriebene Blätter sind verdeckt abzulegen.
- Nach Prüfungsende können keine weiteren Blätter abgegeben werden.
- Legen Sie bitte Ihre Legi auf den Tisch. Es wird eine Kontrolle durchgeführt.

Aufgabe A

Automaten (30 Punkte)

Teilaufgabe 1: Entwurf eines Automaten

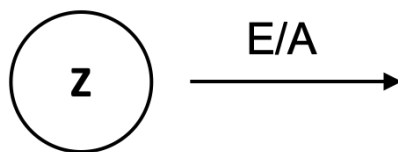
1. Eingänge:

Ausgänge:

2. Anzahl an Kodierungsbits für die Eingänge:

Anzahl an Kodierungsbits für die Ausgänge:

3. Zustandsdiagramm für die zu entwerfen Anlage als **Mealy**-Automat und Beschreibung der Zustände:



4. Minimale Anzahl an Flipflops für die Realisierung dieses Automaten:

5. Anzahl an Zeilen der kompletten Zustandsfolgetabelle:
Anzahl an Spalten der kompletten Zustandsfolgetabelle:
Begründung:

6. Kann dieser Automat auch als Moore-Automat realisiert werden (mit Begründung)?:

Teilaufgabe 2: Analyse des Zustandsdiagramms eines Automaten

7. Automatentyp und Begründung:

zu 8)

e_1	e_0	z_{1n}	z_{0n}	z_{1n+1}	z_{0n+1}	Y_n
0	0	0	0			
0	0	0	1			
0	0	1	0			
0	0	1	1			
0	1	0	0			
0	1	0	1			
0	1	1	0			
0	1	1	1			
1	0	0	0			
1	0	0	1			
1	0	1	0			
1	0	1	1			
1	1	0	0			
1	1	0	1			
1	1	1	0			
1	1	1	1			

zu 9)

z_{1n+1}

$\begin{matrix} e_1e_0 \\ z_1z_0 \end{matrix}$	00	01	11	10
00				
01				
11				
10				

z_{0n+1}

$\begin{matrix} e_1e_0 \\ z_1z_0 \end{matrix}$	00	01	11	10
00				
01				
11				
10				

y_n

$\begin{matrix} e_1e_0 \\ z_1z_0 \end{matrix}$	00	01	11	10
00				
01				
11				
10				

$z_{1(n+1)} =$

$z_{0(n+1)} =$

$y_n =$

11. Unzulässige Eingangskombination:

zu 12)

Z_{1n+1}

e_1e_0 z_1z_0	00	01	11	10
00				
01				
11				
10				

Z_{0n+1}

e_1e_0 z_1z_0	00	01	11	10
00				
01				
11				
10				

$z_{1(n+1)} =$

$z_{0(n+1)} =$

Erreichte Punkte:

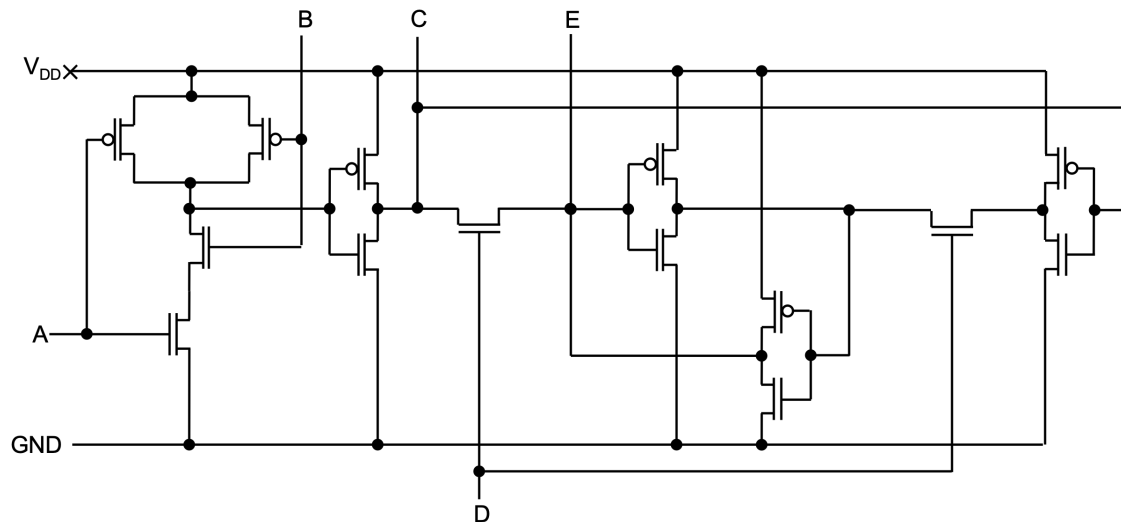
1)	/2
2)	/1
3)	/6
4)	/1
5)	/2
6)	/1
7)	/1
8)	/2
9)	/6
10)	/3
11)	/1
12)	/4
Total	/30

Aufgabe B

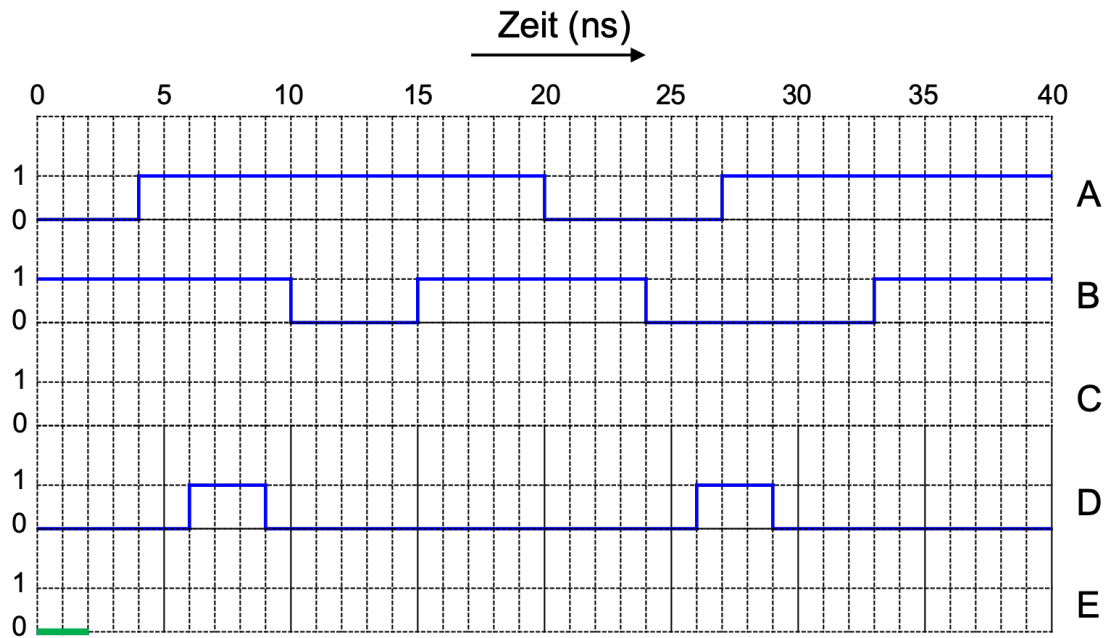
CMOS (16 Punkte)

Teilaufgabe 1: Analyse einer CMOS Schaltung

1. Kennzeichnen der CMOS Blöcke:



2. Zeitdiagramm



Teilaufgabe 2: Pull-up/Pull-down Umwandlung

3. Pull-up Pfad:

4. Funktionsgleichungen:

$$Z_{\text{Pull-up}} =$$

$$Z_{\text{Pull-down}} =$$

Erreichte Punkte:

1)	/6
2)	/4
3)	/4
4)	/2
Total	/16

Aufgabe C

Zahlensysteme, Rechnen und Minimieren (20 Punkte)

Teilaufgabe 1: Umrechnung zwischen Zahlensystemen

1. $-33_{10} =$

2. $10001000_2 =$

3. $-7.3_{10} =$

Fehler:

4. $11111001111.101_2 =$

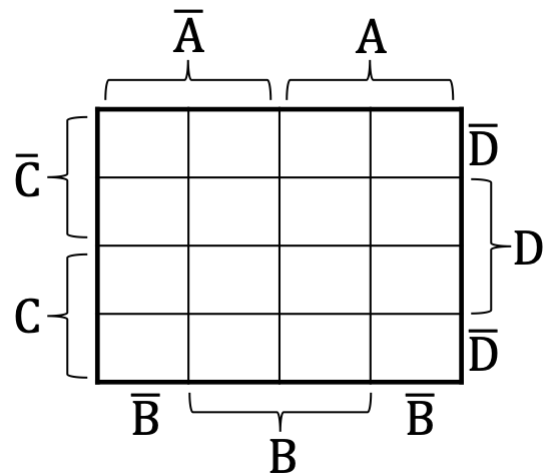
5. A als Zweierkomplementzahl:

-B als Zweierkomplementzahl:

C als Zweierkomplementzahl:

C als Dezimalzahl:

Teilaufgabe 2: Logik Minimierung mit Karnaugh-Diagramm
zu 6)



7. $Z =$

8. Hazards?

Bei welchem Übergang (welchen Übergängen)?

Teilaufgabe 3: Logik Minimierung mit Bool'scher Algebra

9. $\overline{(A \wedge \overline{B}) \vee (\overline{A} \wedge B)} \wedge (A \vee B) =$

10. $((A \wedge \overline{B} \wedge (C \vee (B \wedge D))) \vee \overline{A} \vee B) \wedge \overline{C} =$

Erreichte Punkte:

1)	/1
2)	/1
3)	/2
4)	/1
5)	/3
6)	/4
7)	/2
8)	/1
9)	/2
10)	/3
Total	/20

Aufgabe D

Sequenzielle Schaltungen (20 Punkte)

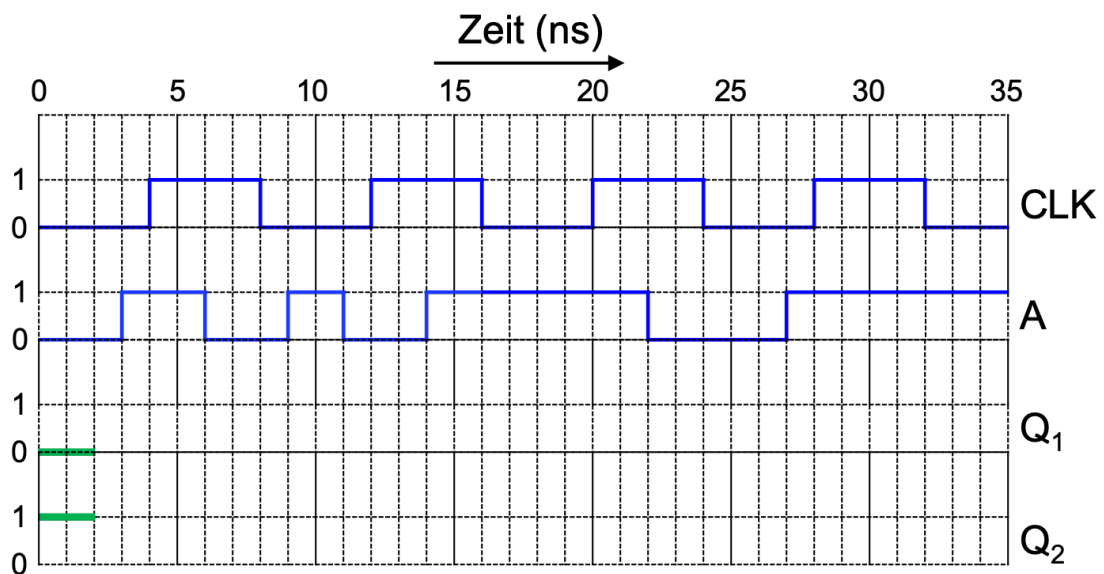
Teilaufgabe 1: Latches und Flipflops

1. Benennung der Bauteile

Links:

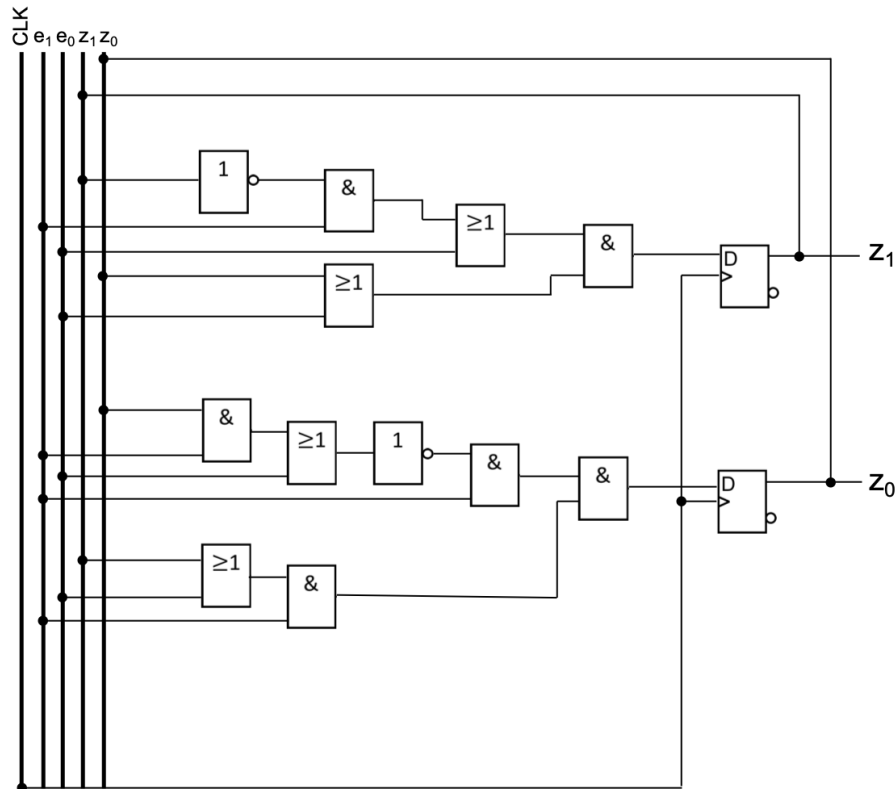
Rechts:

2. Zeitdiagramm für die Bauelemente aus Abbildung D1:



Teilaufgabe 2: Taktfrequenzen von Schaltungen

3. Zeichnen Sie das längste Signalpfad für z_1 und z_0 .



4. Gleichung (Bedingung) für die minimale Taktperiode, die jedem Signalpfad entspricht

$$T_{min,z_1} \geq$$

$$T_{min,z_0} \geq$$

5. $t_{d, and} =$

$$t_{d, or} =$$

Teilaufgabe 3: Asynchronzähler mit T-Flipflops

6. Anzahl an T-Flipflops:

7. Anzahl nicht verwendeter Zustände:

8. Bedingung, damit der Zähler in einen anderen Zustand wechselt, wenn er 12 erreicht:

9. Zähler Schaltung mit T-Flipflops:

Erreichte Punkte:

1)	/2
2)	/4
3)	/2
4)	/2
5)	/3
6)	/1
7)	/1
8)	/1
9)	/4
Total	/20

Basisprüfung D-ITET

Digitaltechnik

27. Januar 2020

Hinweise zu den Aufgaben:

- Punktezahl pro Aufgabe:

Bitte leer lassen!

	Punkte	Visum
A:	30	
B:	16	
C:	20	
D:	20	
Total:	86	

- Schreiben Sie auf jedes Blatt Ihren Kode, der Ihnen auf dem Deckblatt zugeteilt worden ist.
- Beantworten Sie nur gestellte Fragen.
- Sollten Unklarheiten auftreten, wenden Sie sich an die Betreuer.
- Die Aufgaben können in beliebiger Reihenfolge gelöst werden.
- Resultate können nur bewertet werden, wenn ihre Herleitung bzw. Begründung in der angegebenen Lösung enthalten ist.
- Folgefehler werden berücksichtigt.
- **Benutzen Sie für die Lösungen ausschliesslich die vorgedruckten Lösungsblätter.** Sie können jederzeit neue Blätter bei einem Betreuer verlangen. Beachten Sie, dass auf jedem Lösungsblatt steht, für welche Aufgabe es gedacht ist.
- Benutzen Sie einen dokumentenechten Schreiber, d.h. Kugelschreiber, Füllfederhalter o.ä. Bitte fertigen Sie die Prüfung **nicht** mit Bleistift an, **keine rote Farbe!**
- Es sind **keine Kommunikationsmittel** und **keine Rechner** erlaubt.
- Beschriebene Blätter sind verdeckt abzulegen.
- Nach Prüfungsende können keine weiteren Blätter abgegeben werden.
- Legen Sie bitte Ihre Legi auf den Tisch. Es wird eine Kontrolle durchgeführt.

Aufgabe A

Automaten (30 Punkte)

Teilaufgabe 1: Entwurf eines Automaten

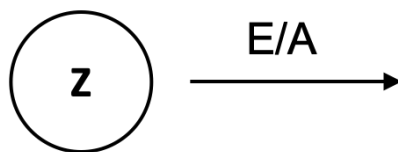
1. Eingänge:

Ausgänge:

2. Anzahl an Kodierungsbits für die Eingänge:

Anzahl an Kodierungsbits für die Ausgänge:

3. Zustandsdiagramm für die zu entwerfen Anlage als **Mealy**-Automat und Beschreibung der Zustände:



4. Minimale Anzahl an Flipflops für die Realisierung dieses Automaten:

5. Anzahl an Zeilen der kompletten Zustandsfolgetabelle:
Anzahl an Spalten der kompletten Zustandsfolgetabelle:
Begründung:

6. Kann dieser Automat auch als Moore-Automat realisiert werden (mit Begründung)?:

Teilaufgabe 2: Analyse des Zustandsdiagramms eines Automaten

7. Automatentyp und Begründung:

zu 8)

e_1	e_0	z_{1n}	z_{0n}	z_{1n+1}	z_{0n+1}	Y_n
0	0	0	0			
0	0	0	1			
0	0	1	0			
0	0	1	1			
0	1	0	0			
0	1	0	1			
0	1	1	0			
0	1	1	1			
1	0	0	0			
1	0	0	1			
1	0	1	0			
1	0	1	1			
1	1	0	0			
1	1	0	1			
1	1	1	0			
1	1	1	1			

zu 9)

z_{1n+1}

$\begin{matrix} e_1e_0 \\ z_1z_0 \end{matrix}$	00	01	11	10
00				
01				
11				
10				

z_{0n+1}

$\begin{matrix} e_1e_0 \\ z_1z_0 \end{matrix}$	00	01	11	10
00				
01				
11				
10				

y_n

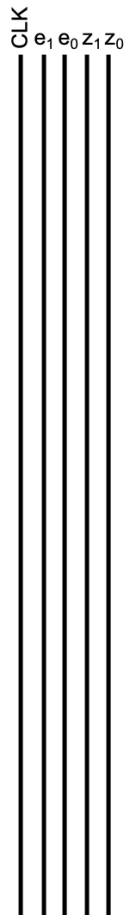
$\begin{matrix} e_1e_0 \\ z_1z_0 \end{matrix}$	00	01	11	10
00				
01				
11				
10				

$z_{1(n+1)} =$

$z_{0(n+1)} =$

$y_n =$

zu 10)



11. Unzulässige Eingangskombination:

zu 12)

Z_{1n+1}

e_1e_0 z_1z_0	00	01	11	10
00				
01				
11				
10				

Z_{0n+1}

e_1e_0 z_1z_0	00	01	11	10
00				
01				
11				
10				

$z_{1(n+1)} =$

$z_{0(n+1)} =$

Erreichte Punkte:

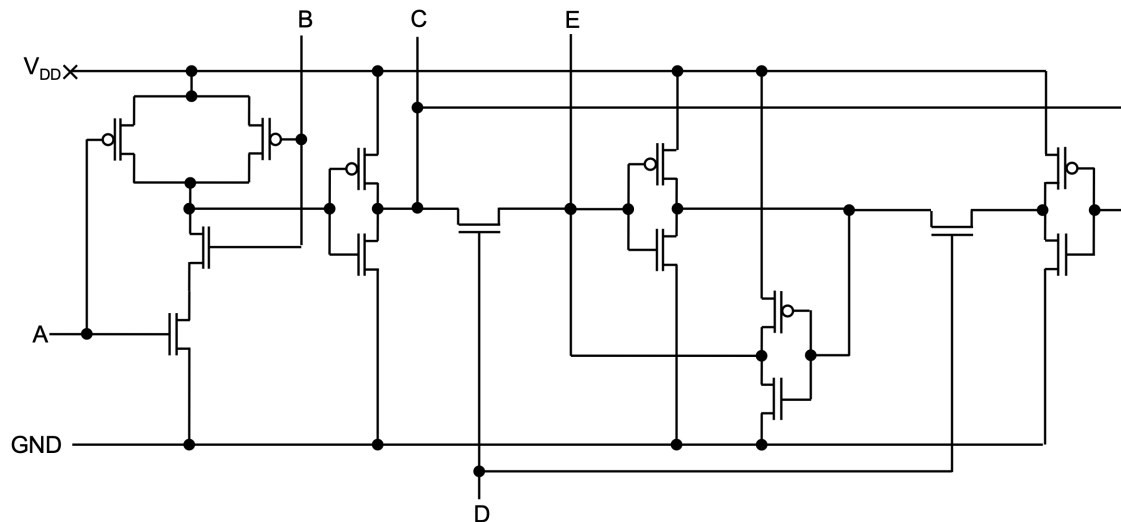
1)	/2
2)	/1
3)	/6
4)	/1
5)	/2
6)	/1
7)	/1
8)	/2
9)	/6
10)	/3
11)	/1
12)	/4
Total	/30

Aufgabe B

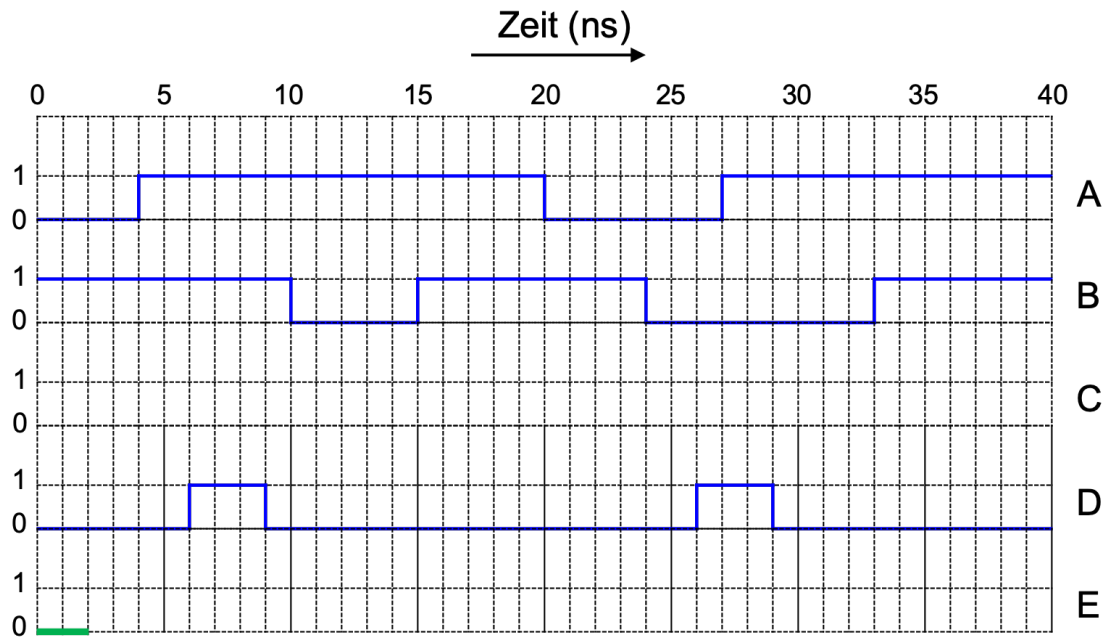
CMOS (16 Punkte)

Teilaufgabe 1: Analyse einer CMOS Schaltung

1. Kennzeichnen der CMOS Blöcke:



2. Zeitdiagramm



Teilaufgabe 2: Pull-up/Pull-down Umwandlung

3. Pull-up Pfad:

4. Funktionsgleichungen:

$$Z_{\text{Pull-up}} =$$

$$Z_{\text{Pull-down}} =$$

Erreichte Punkte:

1)	/6
2)	/4
3)	/4
4)	/2
Total	/16

Aufgabe C

Zahlensysteme, Rechnen und Minimieren (20 Punkte)

Teilaufgabe 1: Umrechnung zwischen Zahlensystemen

1. $-33_{10} =$

2. $10001000_2 =$

3. $-7.3_{10} =$

Fehler:

4. $11111001111.101_2 =$

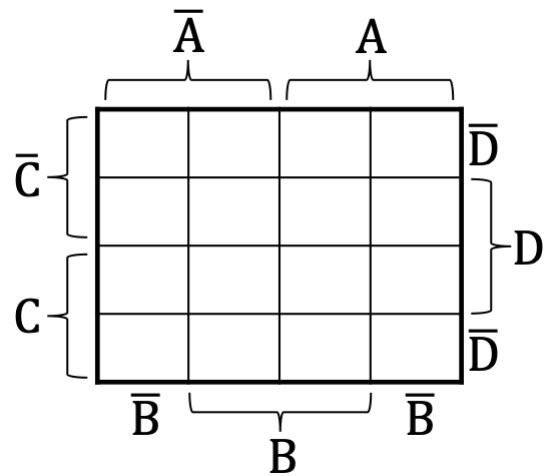
5. A als Zweierkomplementzahl:

-B als Zweierkomplementzahl:

C als Zweierkomplementzahl:

C als Dezimalzahl:

Teilaufgabe 2: Logik Minimierung mit Karnaugh-Diagramm
zu 6)



7. $Z =$

8. Hazards?

Bei welchem Übergang (welchen Übergängen)?

Teilaufgabe 3: Logik Minimierung mit Bool'scher Algebra

9. $\overline{(A \wedge \overline{B}) \vee (\overline{A} \wedge B)} \wedge (A \vee B) =$

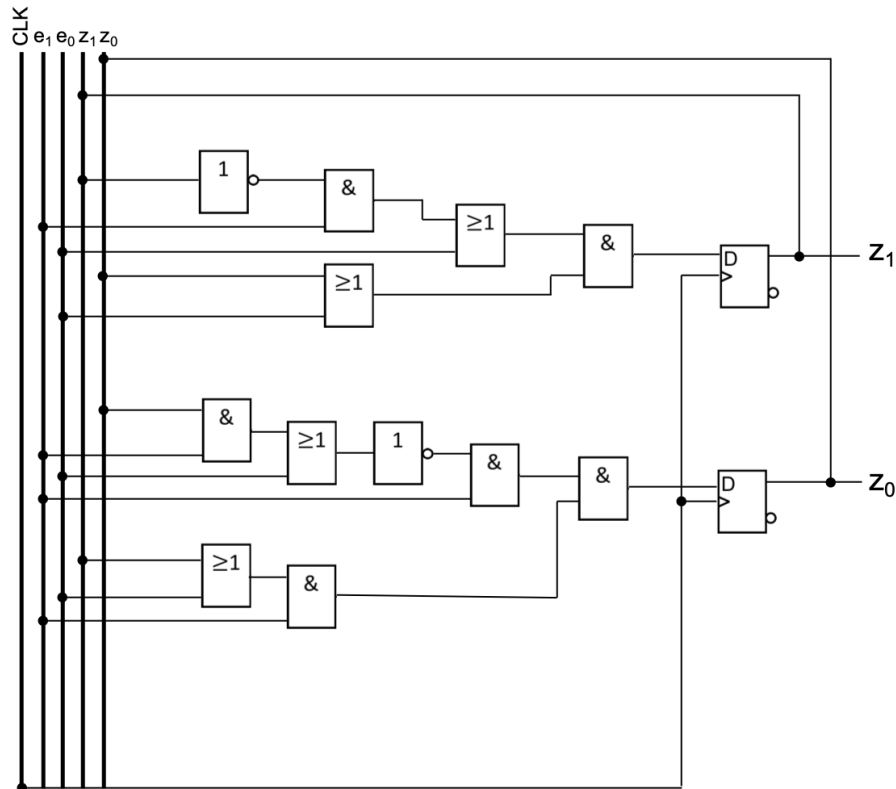
10. $((A \wedge \overline{B} \wedge (C \vee (B \wedge D))) \vee \overline{A} \vee B) \wedge \overline{C} =$

Erreichte Punkte:

1)	/1
2)	/1
3)	/2
4)	/1
5)	/3
6)	/4
7)	/2
8)	/1
9)	/2
10)	/3
Total	/20

Teilaufgabe 2: Taktfrequenzen von Schaltungen

3. Zeichnen Sie das längste Signalpfad für z_1 und z_0 .



4. Gleichung (Bedingung) für die minimale Taktperiode, die jedem Signalpfad entspricht

$$T_{min,z_1} \geq$$

$$T_{min,z_0} \geq$$

5. $t_{d, and} =$

$$t_{d, or} =$$

Teilaufgabe 3: Asynchronzähler mit T-Flipflops

6. Anzahl an T-Flipflops:

7. Anzahl nicht verwendeter Zustände:

8. Bedingung, damit der Zähler in einen anderen Zustand wechselt, wenn er 12 erreicht:

9. Zähler Schaltung mit T-Flipflops:

Erreichte Punkte:

1)	/2
2)	/4
3)	/2
4)	/2
5)	/3
6)	/1
7)	/1
8)	/1
9)	/4
Total	/20

Tipps

Tipps

- Macht euch klar, was eure Stärken sind.
- Ihr müsst nicht alles lösen.

Fragen?

Viel Erfolg!