Digitaltechnik Übung 6

Matteo Dietz
mdietz@student.ethz.ch

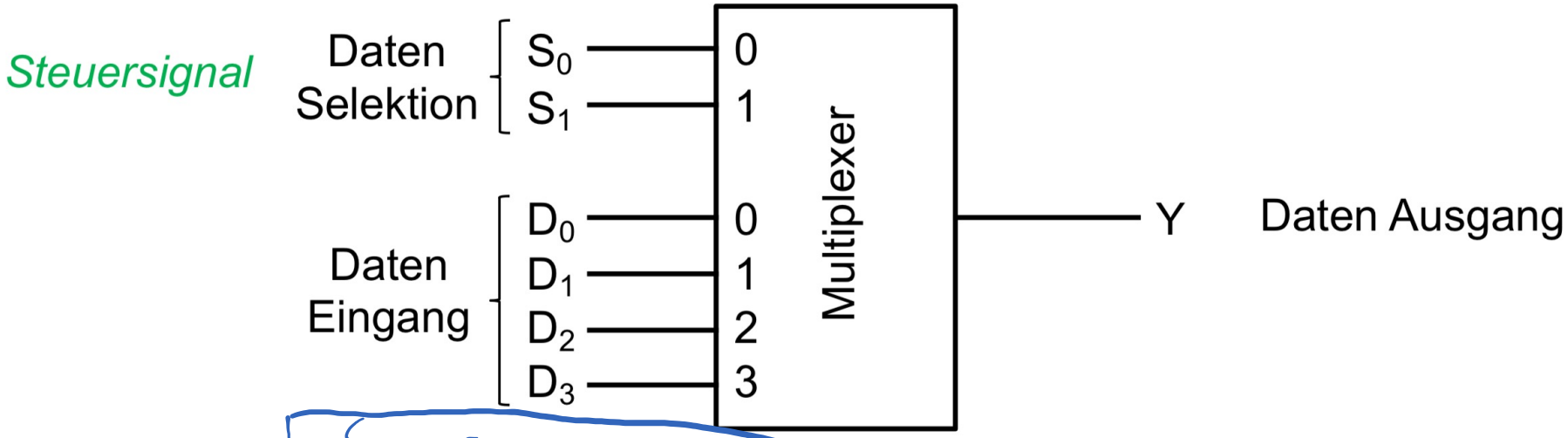
Polybox

- https://polybox.ethz.ch/index.php/s/VehRU12QqdJv98i

Theorie – Rechenschaltungen und Datenpfadkomponenten

Theorie – Multiplexer

- Funktion: Durch Auswertung eines n-Bit-Auswahlsignals einen von 2^n Eingängen auf den Ausgang schalten.
- In anderen Worten ein Datenselektor



$S_0 = S_1 = 0 \Rightarrow Y = D_0$

$S_0 = 0, S_1 = 1 \Rightarrow Y = D_1$

$S_0 = 1, S_1 = 0 \Rightarrow Y = D_2$

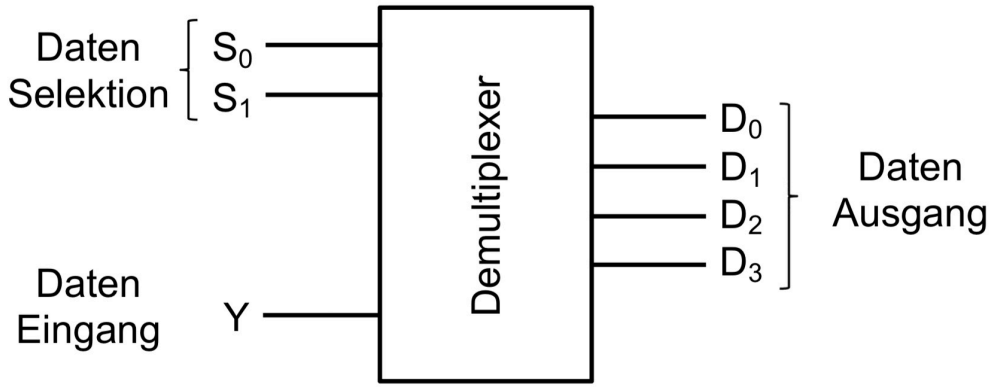
$S_0 = S_1 = 1 \Rightarrow Y = D_3$

$$Y = (\overline{S_0} \wedge \overline{S_1} \wedge D_0) \vee (\overline{S_0} \wedge S_1 \wedge D_1) \vee (S_0 \wedge \overline{S_1} \wedge D_2) \vee (S_0 \wedge S_1 \wedge D_3)$$

Theorie – Demultiplexer

- Gegenstück zum Multiplexer
- Nimmt Daten aus einem einzigen Kanal und verteilt dieses über mehrere Kanäle

  Beispiel: 1-zu-4 Demultiplexer



Die Ausgänge können zum Beispiel so programmiert werden:

$$D_0 = \overline{S_0} \wedge \overline{S_1} \wedge Y, \qquad D_1 = \overline{S_0} \wedge S_1 \wedge Y,$$

$$D_2 = S_0 \wedge \overline{S_1} \wedge Y, \qquad D_3 = S_0 \wedge S_1 \wedge Y$$

Theorie – Halbaddierer

- Ziel: Entwurf einer Schaltung, die zwei 1Bit Zahlen A und B addieren kann:

A B

$0 + 0 = 00$

$0 + 1 = 01$

$1 + 0 = 01$

$1 + 1 = 10$

CO Übertrag, Carry Out

SUM Summe

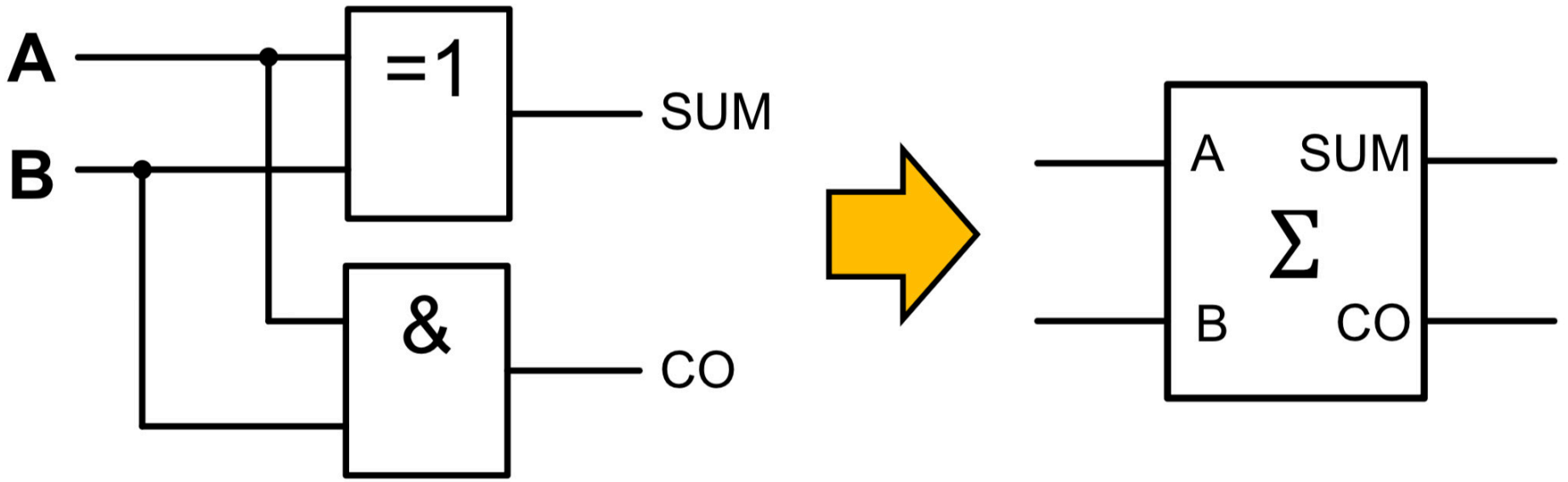
Wahrheitstabelle

A	B	SUM	CO
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

$SUM = A \oplus B$

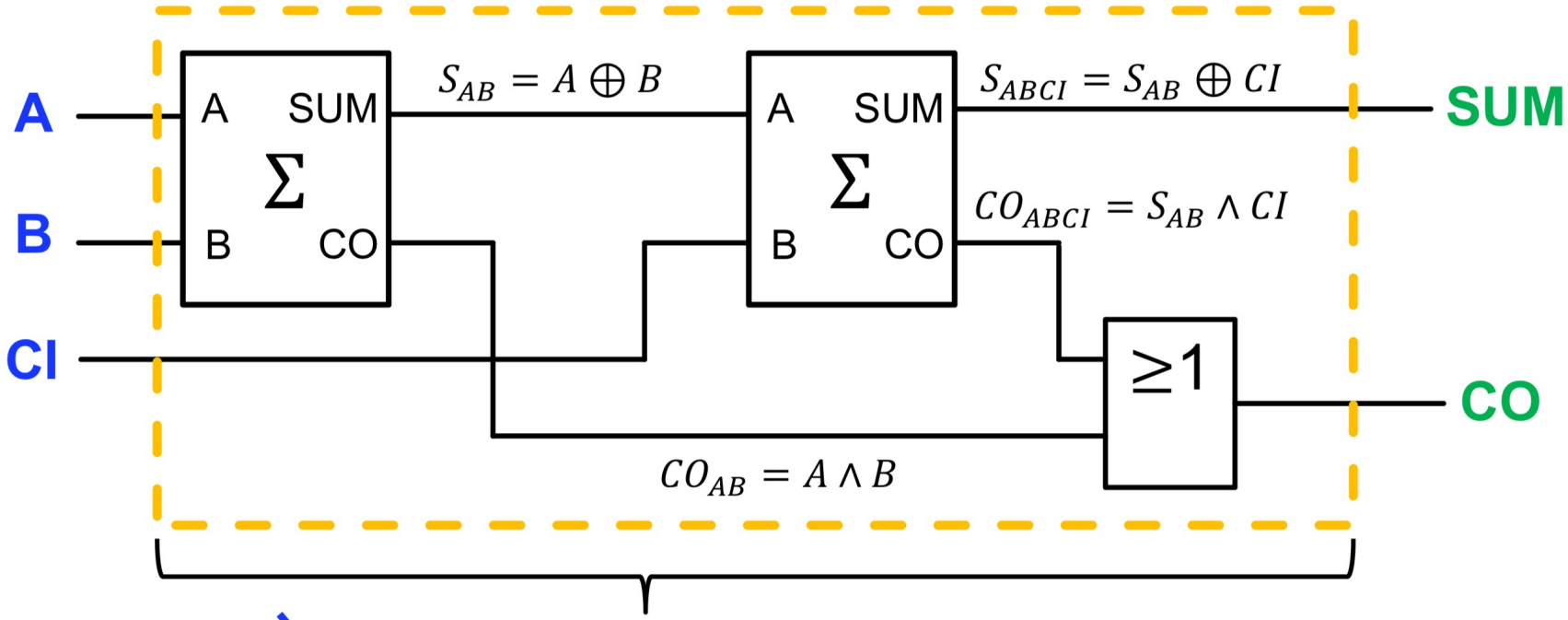
$CO = A \wedge B$

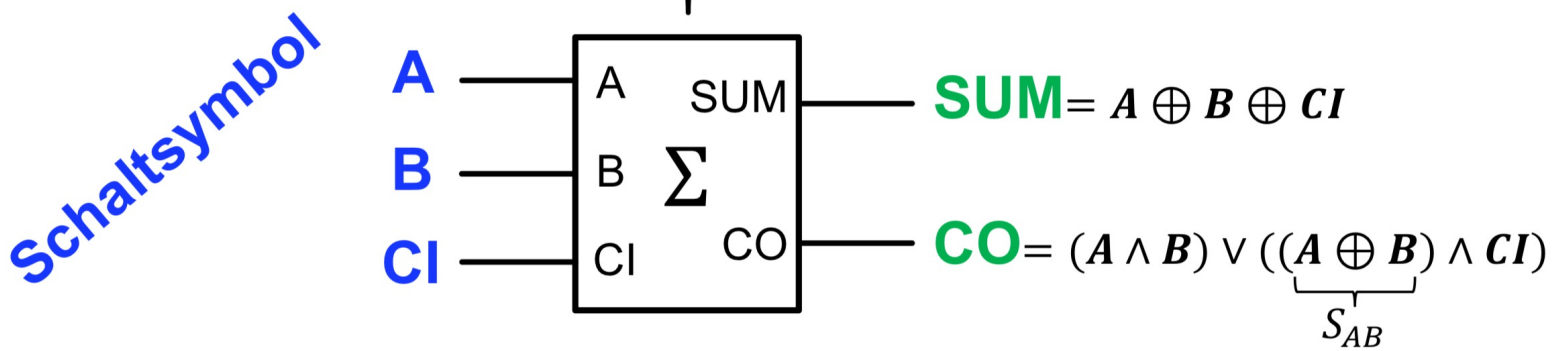
Theorie – Halbaddierer



Theorie – Volladdierer

- Ziel: Entwurf einer Schaltung, die drei 1Bit Zahlen A, B und CI addieren kann:





Theorie – Mehrbit-Addierer

- Ziel: Addition von mehrstelligen (Dual-)Zahlen
- Serienaddierer:
  - Pro Taktschritt wird eine Stelle addiert
- Paralleladdierer:
  - Pro Taktschritt werden alle Stellen addiert
  - 1. Paralleladdierer in der Normalform (Schaltnetz)
  - 2. Ripple-Carry Addierer (Kaskadierung von Volladdierern)
  - 3. Carry-Look-Ahead Addierer (Mischung aus 1.&2.)

Theorie – Paralleladdierer in der Normalform

- Ziel: Addition von zwei n-stelligen Dualzahlen mit möglichst geringer Laufzeit

- Für jedes Bit wird eine eigene Schaltgleichung generiert und in eine kombinatorische Schaltung umgesetzt. (mittels Min- oder Maxtermen)

$$S_0 = A_0 \oplus B_0 = (A_0 \wedge \overline{B_0}) \vee (\overline{A_0} \wedge B_0)$$

$$C_0 = A_0 \wedge B_0$$

$$S_1 = A_1 \oplus B_1 \oplus C_0 = ((A_1 \wedge \overline{B_1}) \vee (\overline{A_1} \wedge B_1)) \oplus (A_0 \wedge B_0)$$
$$= (A_0 \wedge B_0 \wedge A_1 \wedge B_1) \vee (A_0 \wedge B_0 \wedge \overline{A_1} \wedge \overline{B_1})$$

$$C_1 = (A_1 \wedge B_1) \vee (A_1 \wedge A_0 \wedge B_0) \vee (B_1 \wedge A_0 \wedge B_0)$$

Theorie – Paralleladdierer in der Normalform Eigenschaften

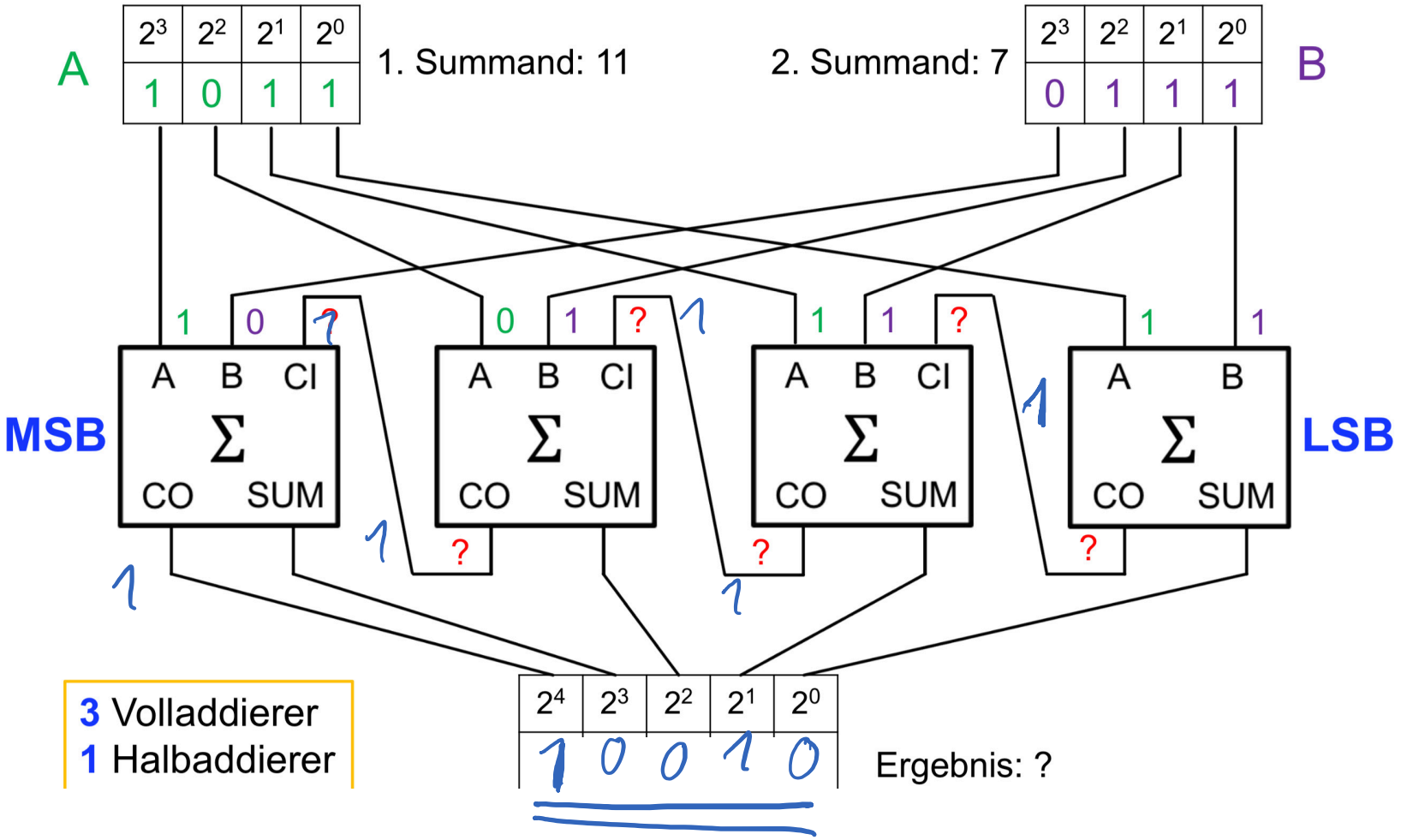
- Vorteile:
  - Zwischen Eingangsbit und Ausgangssignal gibt ex max. 3 Grundgatter
  - Laufzeit beträgt unabhängig von der Stellenanzahl der Summanden maximal 3 Gatterlaufzeiten => sehr schnell

- Nachteile:
  - Für die (n+1) Ausgänge einer Addition zweier n-stelliger Summanden müssen ~n*2^(2n-1) Min- bzw. Maxterme verknüpft warden.

- => Schneller aber schaltungsaufwendiger Addierer

Theorie – Ripple-Carry Addierer

4-Bit-Parallel-Addierschaltung (Kaskadierung von Addierern)



Theorie – Ripple-Carry Addierer Eigenschaften

- Vorteile:
  - Einfache Skalierung/Erweiterung möglich durch Kaskadierung
  - z.B. aus zwei 4-Bit-Addierern kann direct ein 8-Bit-Addierer aufgebaut warden
  - Schaltungsaufwand wächst linear mit Stellenzahl

- Nachteile:
  - Summe und Übertrag der i-ten Stelle können erst berechnet werden, wenn alle vorherigen Summen und Überträge berechnet wurden.
  - => Addierzeit wächst linear mit Stellenzahl

- => Langsamer aber einfacher Addierer

Theorie – Carry-Look-Ahead Addierer Eigenschaften

- Ziel: Kombination der Vorteile des Normalform und Ripple-Carry Addierers

- Implementation:
  - Kaskadierung der Addierer wie im Ripple-Carry Addierer
  - Berechung der Überträge erfolgt parallel zur Summenbildung mittels einer extra kombinatorischen Schaltung in einem einzigen Schritt.

- Eigenschaften:
  - Gleiche Geschwindigkeit wie Paralleladdierer in der Normalform, braucht aber weniger Grundgatter
  - Aufwand ist grösser als beim Ripple-Carry

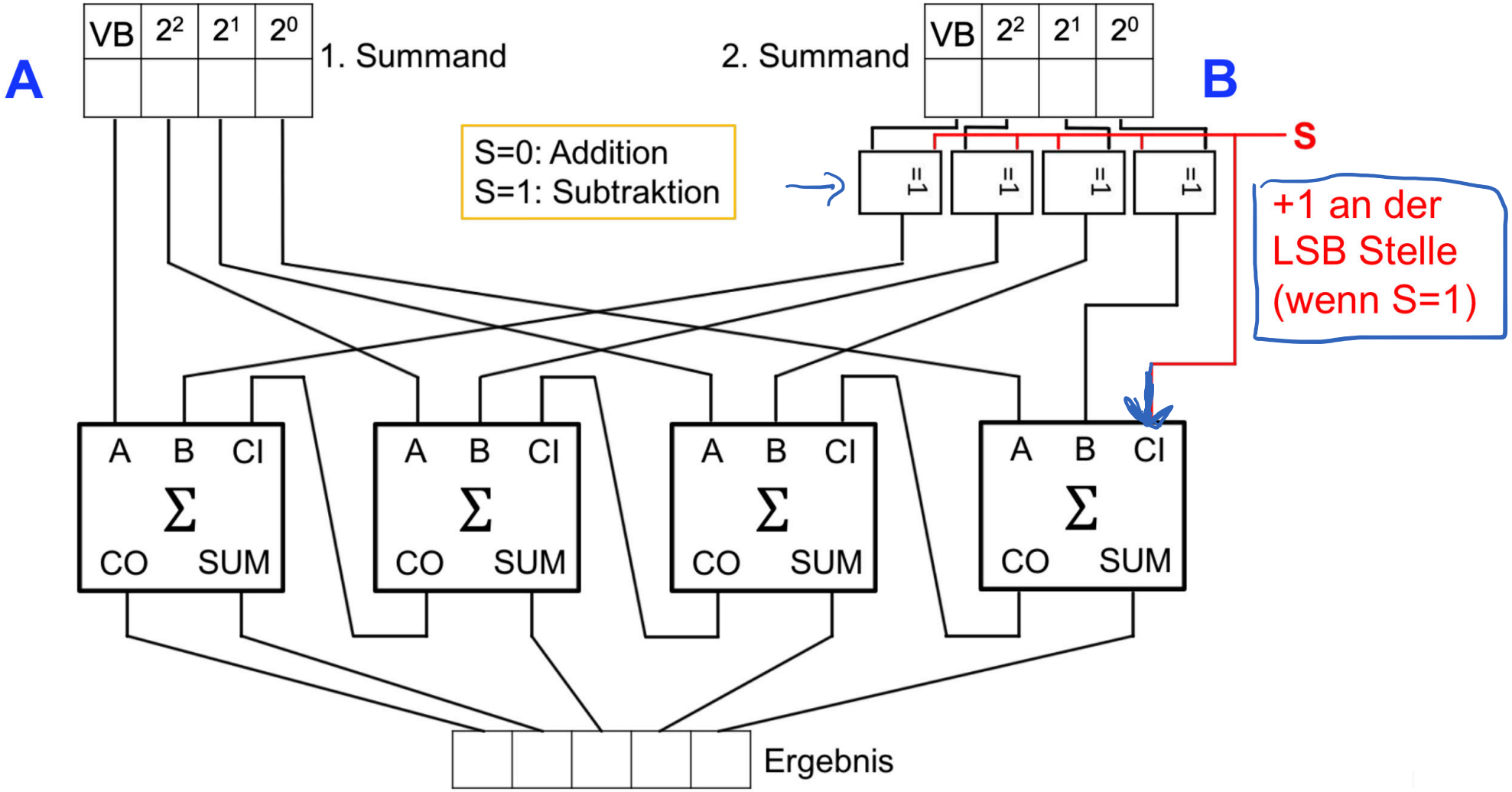
Theorie – Subtrahierer

- Ziel: Entwurf einer Schaltung, die Mehrbit-Addition und Subtraktion durchführt.
- Subtraktion: Zweierkomplement des Subtrahenden bilden und anschliessend normale Addition durchführen.
- Zweierkomplement bilden:
  - Bitinversion mit XOR-Gattern:

$B_{alt,i}$	S	$B_{neu,i} = B_{alt,i} \oplus S$	
0	0	0	Passiert nichts
1	0	1	
0	1	1	Bit Inversion
1	1	0	

  - +1 im LSB mit CI (siehe nächste Folie)

Theorie – Subtrahierer



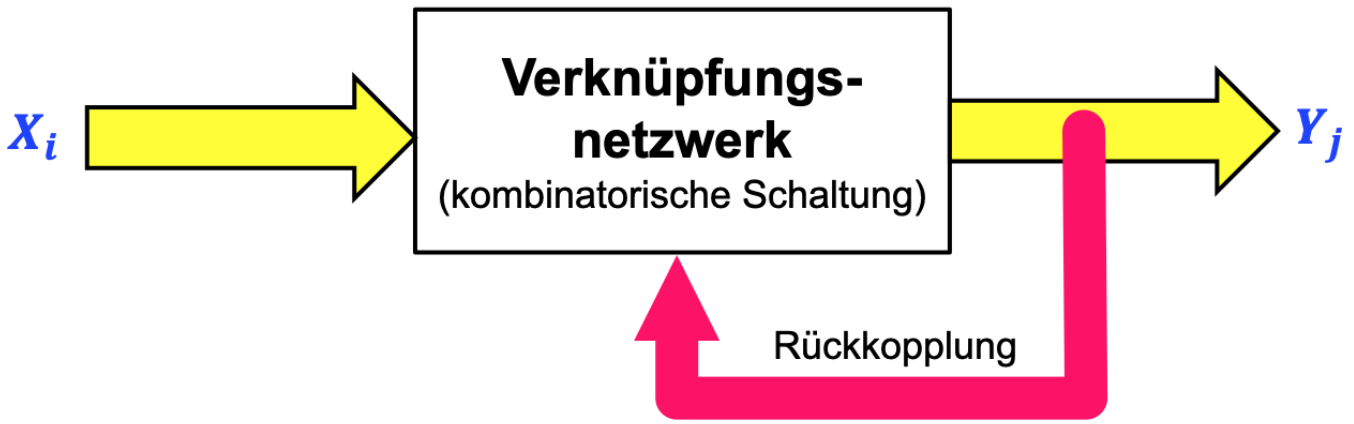
Theorie – Latches und Flipflops

- Ziel: Entwurf einer Schaltung, die Informationen speichern kann

Theorie – Sequentielle Schaltung

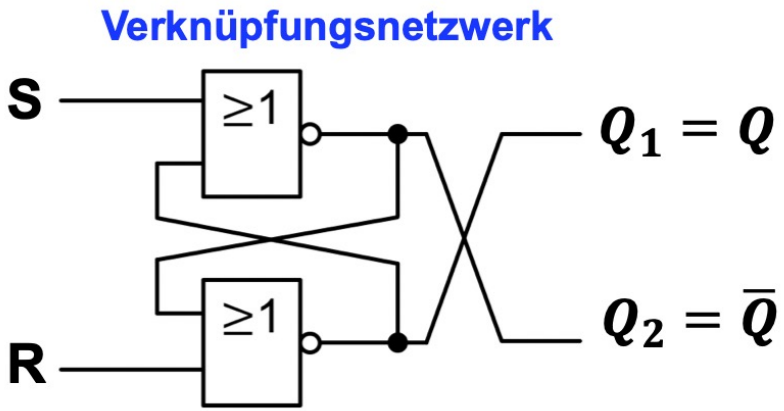
- Mindestens eine Rückkopplung
- Wert am Ausgang hängt auch von vergangenen Werten ab

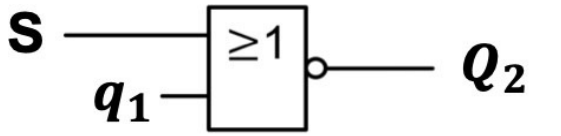
$$Y_j(t_n) = f(X_i(t_n), Y_j(t_{n-1}))$$

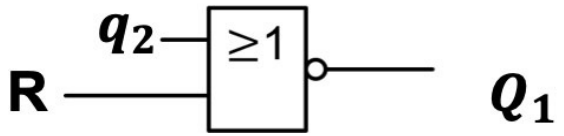


Theorie – SR-Latch

- 2 Eingänge: Set und Reset
- 2 Ausgänge: Q und ¬Q







Es gelten die folgenden Gleichungen

$$Q_1 = \overline{R \vee q_2} = \overline{R} \wedge \overline{q_2}$$

Und

$$Q_2 = \overline{S \vee q_1} = \overline{S} \wedge \overline{q_1}$$

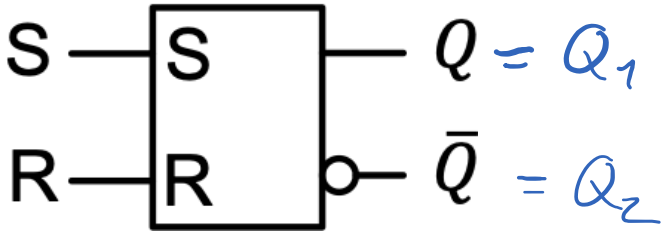
Theorie – SR-Latch

- Gleichung: $Q_{1n+1} = S \vee (Q_{1n} \wedge \overline{R})$
- $S \wedge R = 0$, da die Ausgänge nicht gleich sein dürfen

- Folgezustandstabelle des SR-Latches:

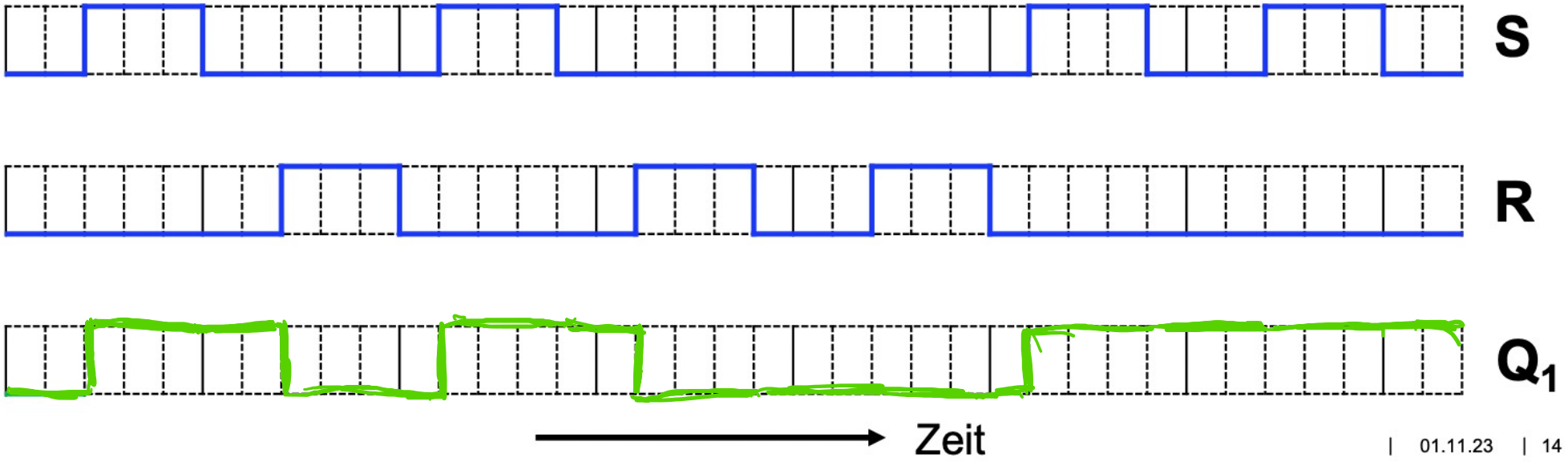
Fall	S	R	Q_{1n+1}	Q_{2n+1}	
1	0	0	Q_{1n}	Q_{2n}	speichern
2	0	1	0	1	rücksetzen
3	1	0	1	0	setzen
4	1	1	-	-	unzulässig

Schaltzeichen



Aufgabe – SR-Latch

Beispiel: Zeitverhalten eines SR-Latches



| 01.11.23 | 14

Theorie – SRT-Latch

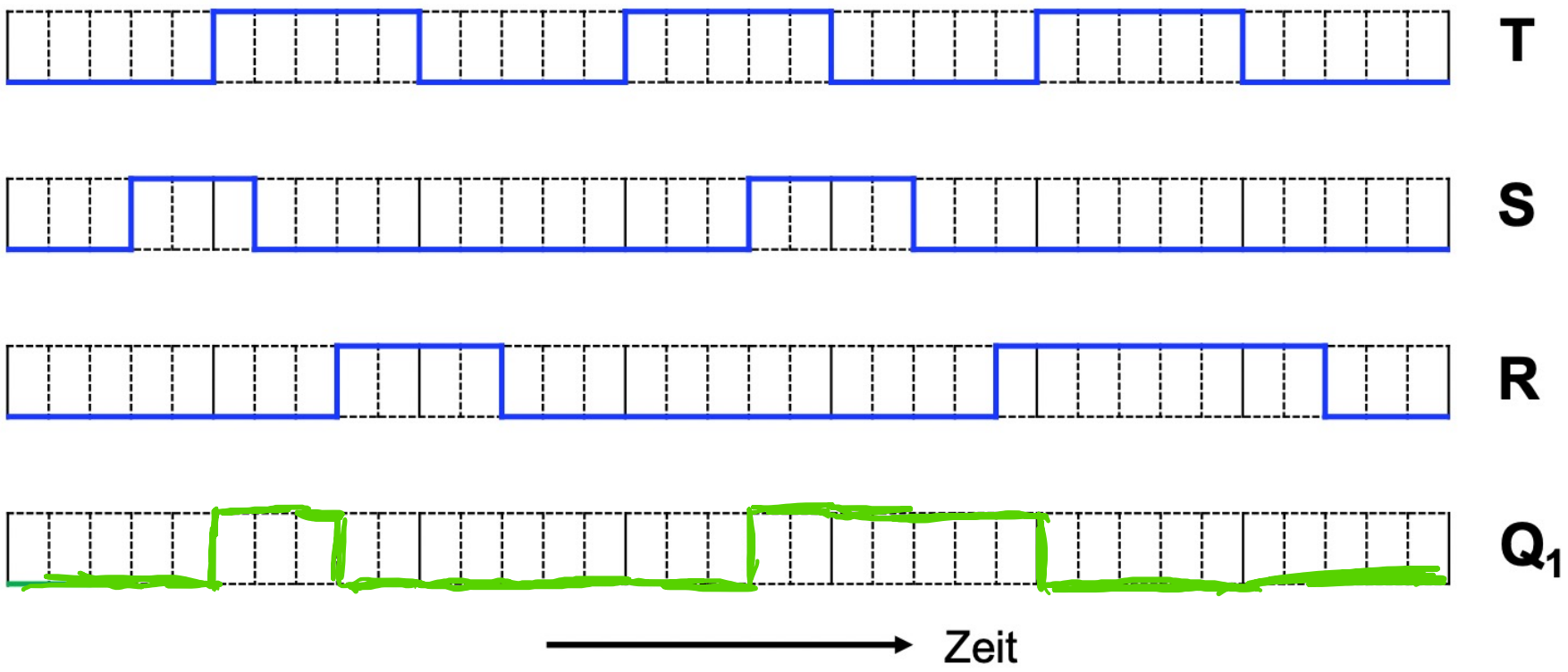
- Taktzustandgesteuertes SR-Latch
- Set/Reset nur, wenn T = 1

- Folgezustandstabelle des SRT-Latches:

Fall	T	S	R	Q_{1n+1}	Q_{2n+1}	
1	0	X	X	Q_{1n}	Q_{2n}	keine Änd.
2	1	0	0	Q_{1n}	Q_{2n}	speichern
3	1	0	1	0	1	rücksetzen
4	1	1	0	1	0	setzen
5	1	1	1	-	-	unzulässig

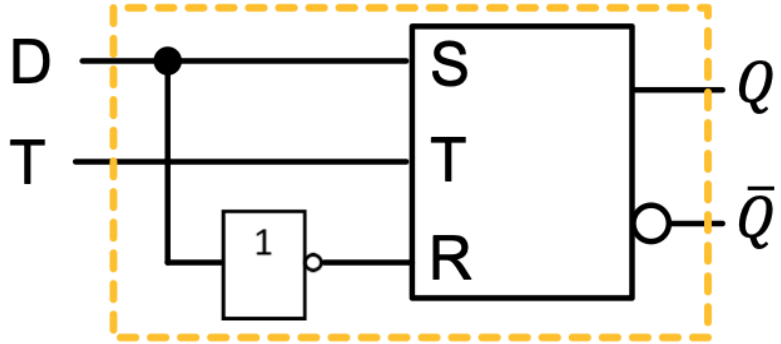
Aufgabe – SRT-Latch

Zeitverhalten eines taktzustandsgesteuerten SRT-Latches:

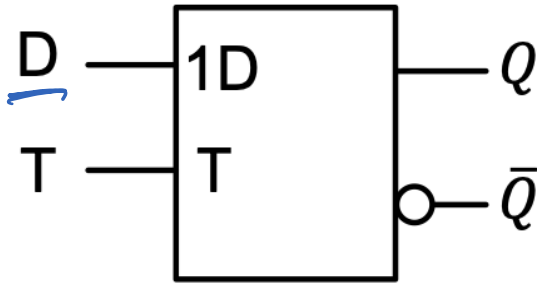


Theorie – D-Latch

- T = 1: transparent, $Q_{1n+1} = D$
- T = 0: speichert zuletzt anliegenden Wert, $Q_{1n+1} = Q_{1n}$
- Gleichung: $Q_{n+1} = (Q_n \wedge \overline{T}) \vee (D \wedge T)$



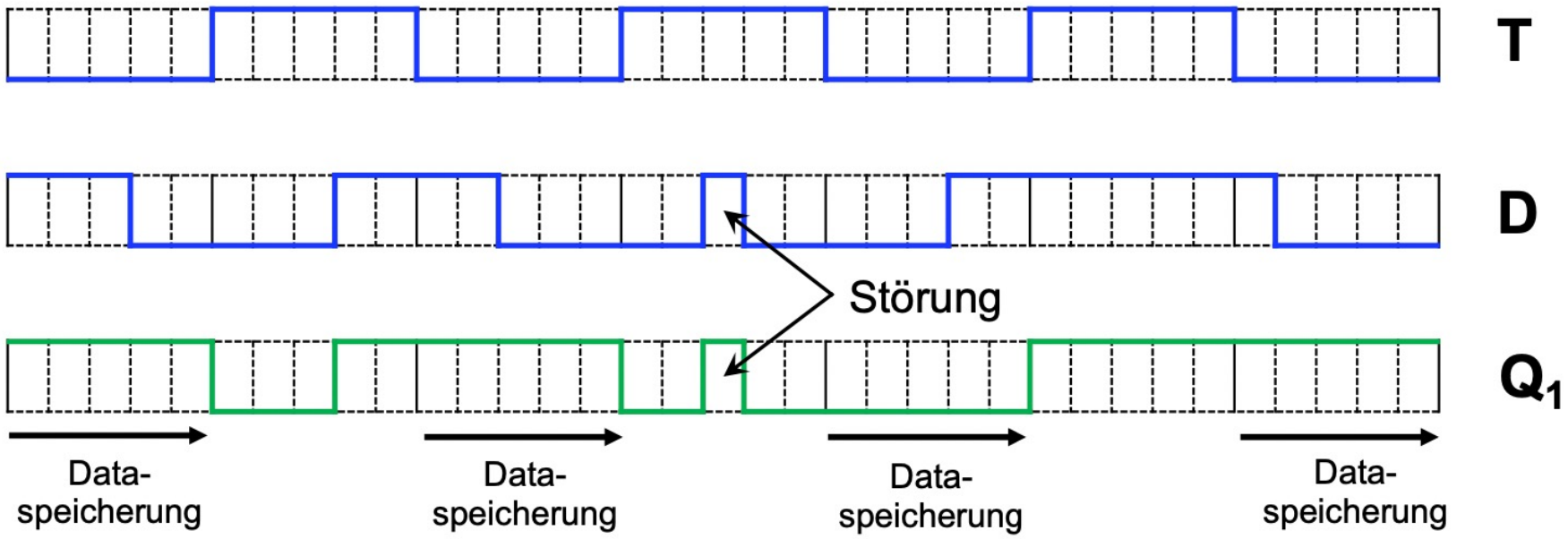
Basisschaltung D-Latch



Schaltzeichen D-Latch

Theorie – D-Latch

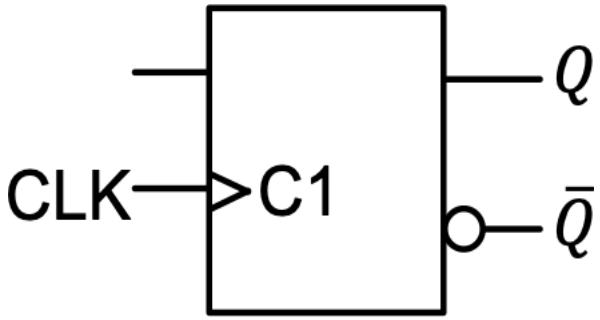
- Probleme von Latches:
  - Latch ist transparent gegenüber Störimpulsen während T=1:



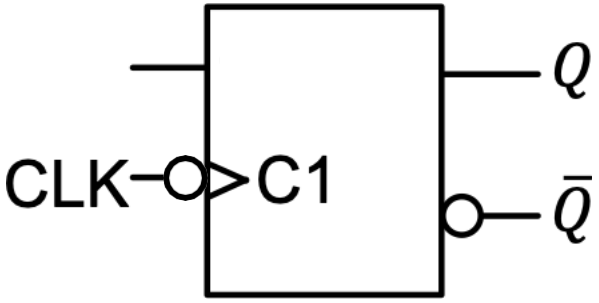
- Lösung dieser Probleme:
  - Flipflops

Theorie – Flipflops

- Taktflankengesteuert
- Flipflops sind nur bei Taktänderungen (Taktflanken) aktiv
  - Steigende Taktflanke (links)
  - Fallende Taktflanke (rechts)



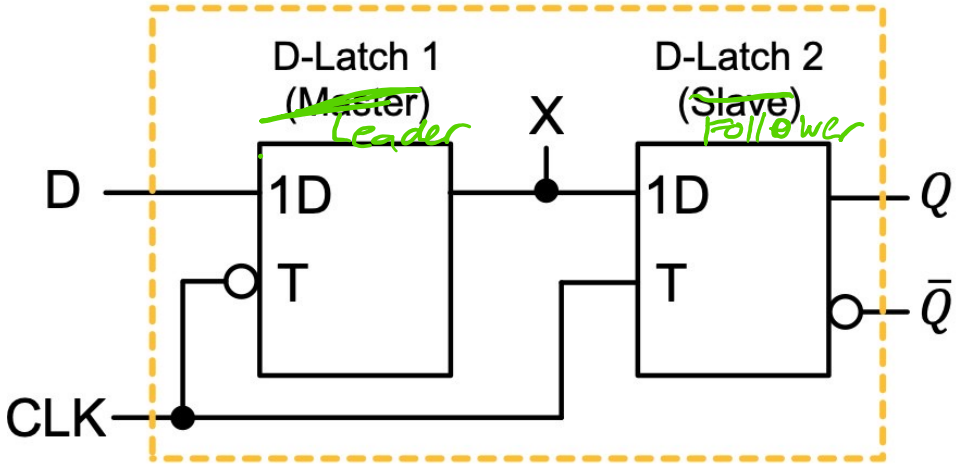
T = 0→1



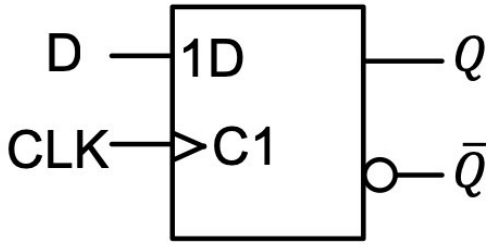
T = 1→0

Theorie – D-Flipflop

- Wert bei steigender Taktflanke wird übernommen und gespeichert
- Gleichung: $Q_{n+1} = D$, wenn CLK 0→1, sonst keine Änderung

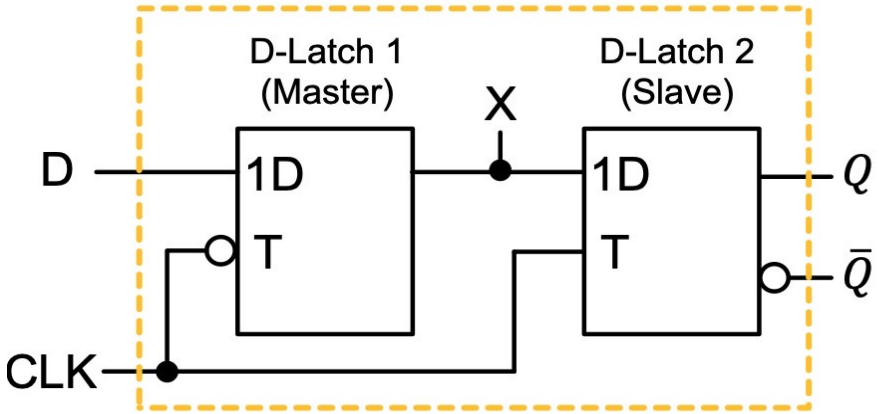


Basisschaltung D-Flipflop

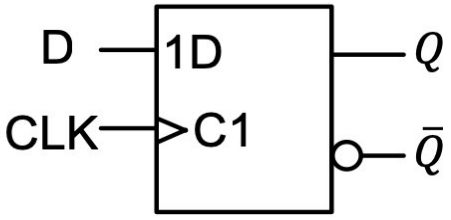


Schaltzeichen D-Flipflop

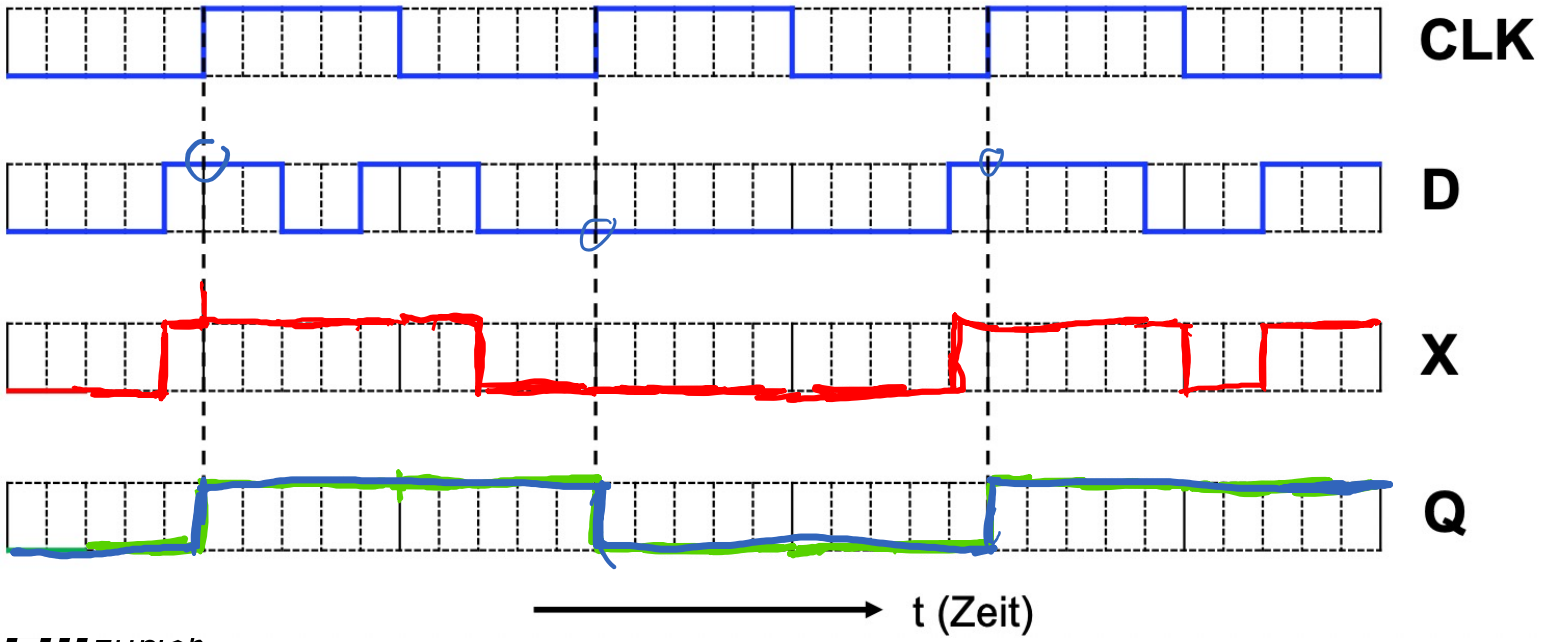
Aufgabe – D-Flipflop



Basisschaltung D-Flipflop

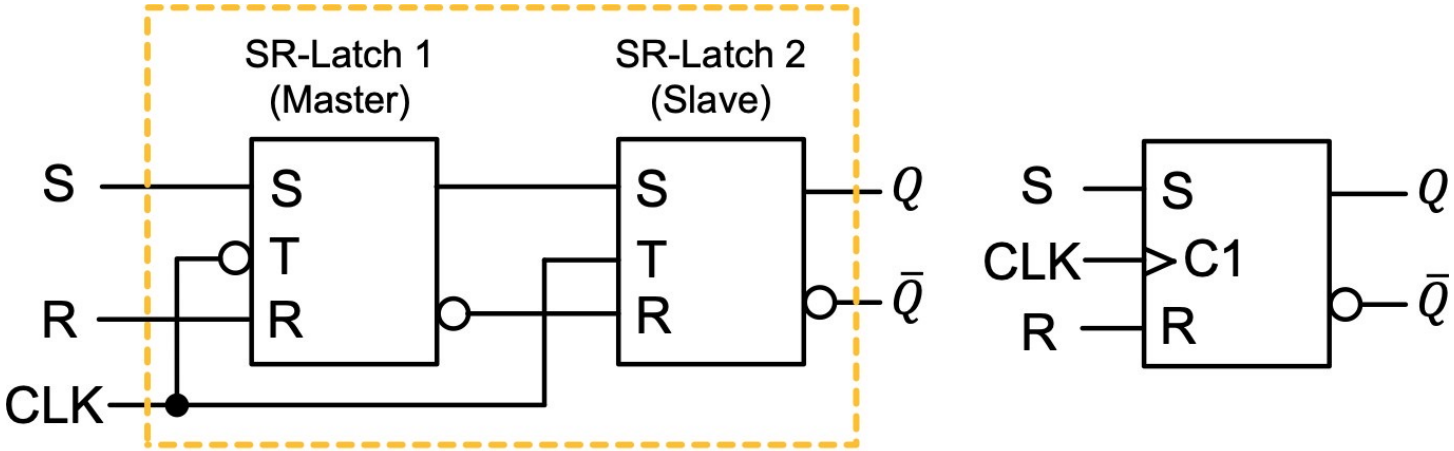


Schaltzeichen D-Flipflop



Theorie – SR-Flipflop

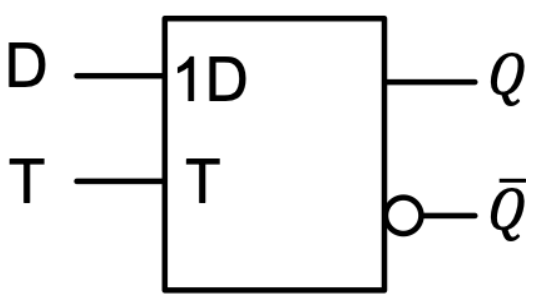
- Set/Reset nur bei steigender Taktflanke
- Weniger störungsanfällig als Latch
- Gleichung: $Q_{1n+1} = (S \vee (\bar{R} \wedge Q_1))_n$ unter der Bedingung $R \wedge S = 0$

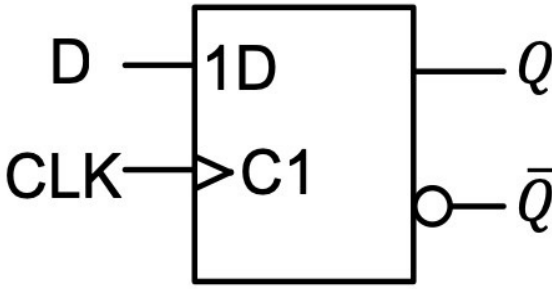


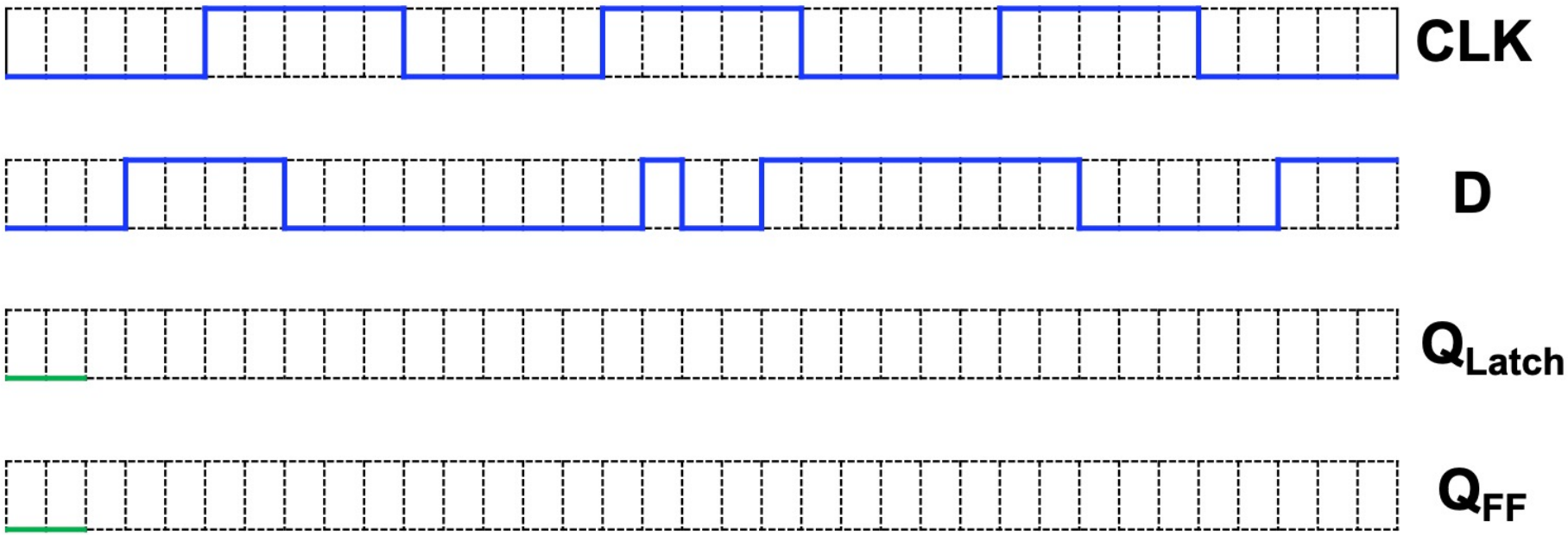
Basisschaltung SR-Flipflop

Schaltzeichen SR-Flipflop

Aufgabe – Unterschied D-Latch und D-Flipflop

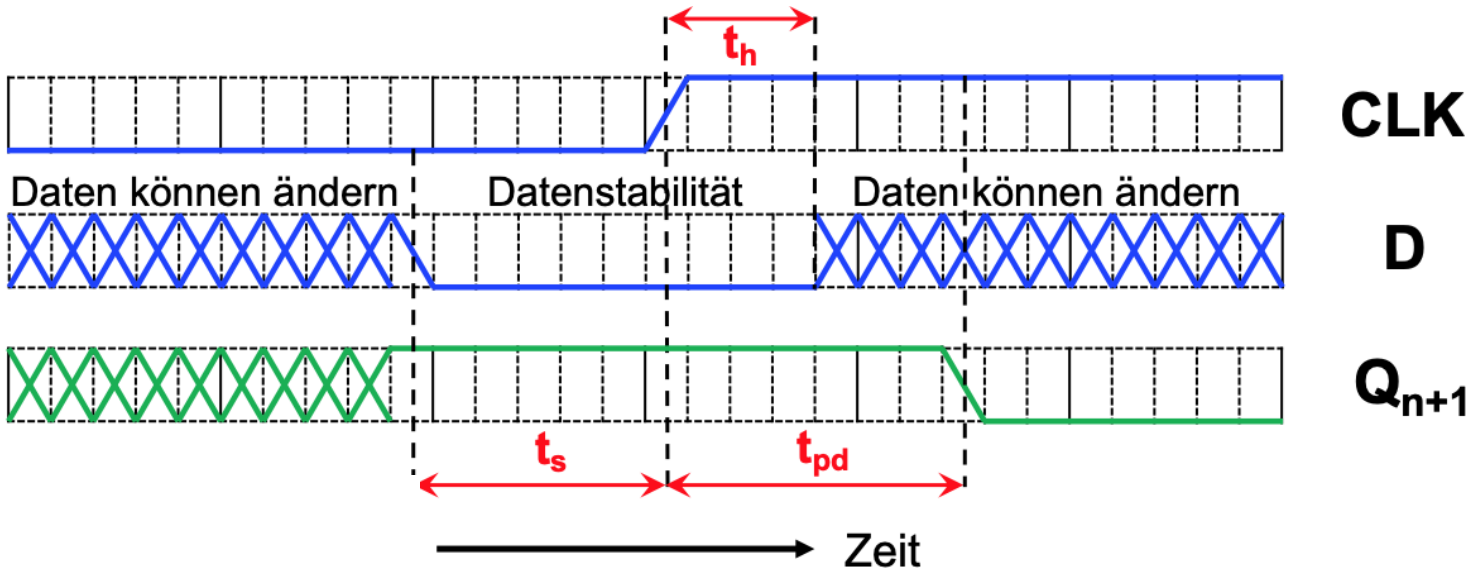






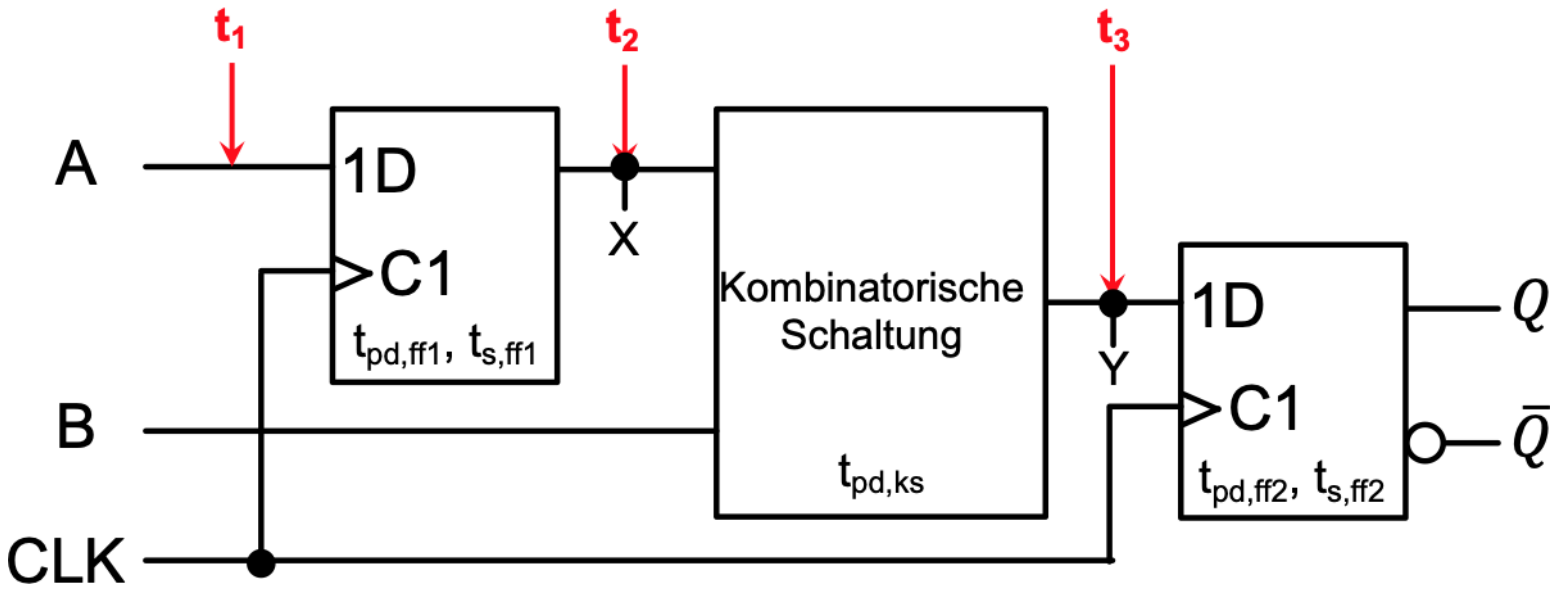
Theorie – Dynamik von Flipflop

- Setup-zeit t_s: So lange muss Signal stabil vor Taktflanke anliegen
- Haltezeit t_h: So lange muss Signal stabil nach Taktflanke anliegen
- Verzögerungszeit t_{pd}: Zwischen Taktflanke und Reaktion am Ausgang



Theorie – Maximale Taktfrequenz

- Ab 2 kaskadierten Flipflops müssen wir die Verzögerungszeiten berücksichtigen
- Minimale Taktperiode: $T_{min} = t_{pd,ff1} + t_{pd,ks} + t_{s,ff2}$
- Maximale Taktfrequenz: $f_{max} = 1/T_{min}$



Kahoot!