Digitaltechnik
Wintersemester 2017/2018
9. Vorlesung

TECHNISCHE UNIVERSITÄT DARMSTADT

Inhalt

TECHNISCHE UNIVERSITÄT DARMSTADT

Einleitung

TECHNISCHE UNIVERSITÄT DARMSTADT

Organisatorisches

TECHNISCHE UNIVERSITÄT DARMSTADT

- Zusammenlegung von Übungsgruppen ab KW 51 geplant
  - Tutoren-Zuordnung bleibt erhalten
  - genaue Infos bis Ende KE50 im Moodle

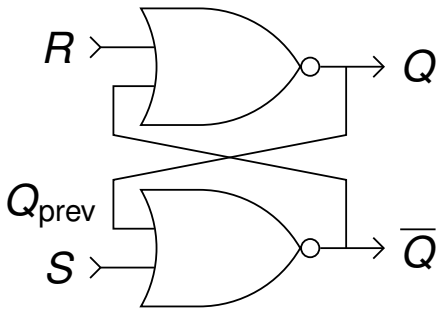
Rückblick auf die letzten Vorlesungen

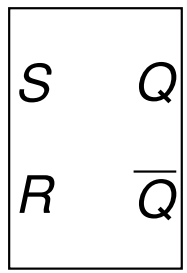
TECHNISCHE UNIVERSITÄT DARMSTADT

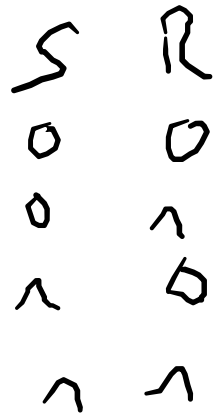
- Sequentielle Logik
  - Sequentielle Schaltungen
  - Speicherelemente
  - Synchrone sequentielle Logik
- Endliche Zustandsautomaten
  - Konzept, Notationen und Anwendungsbeispiele
  - Moore vs. Mealy
  - Zerlegen von Zustandsautomaten

Harris 2013
Kap. 3.1 - 3.4

Wiederholung: Speicherelemente

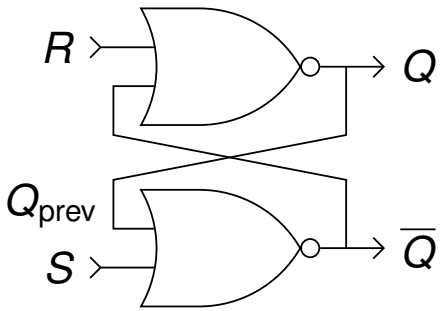


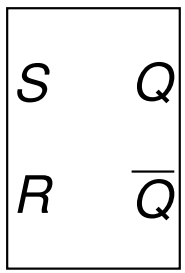


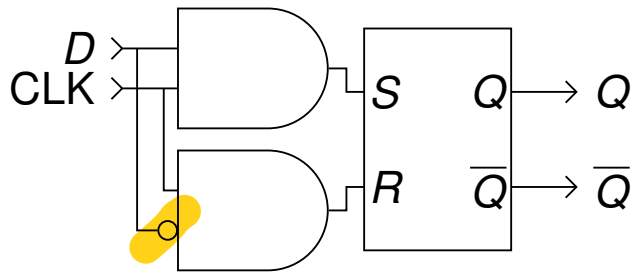


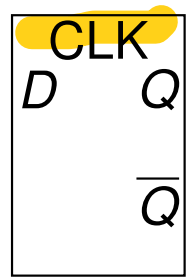
Wiederholung: Speicherelemente

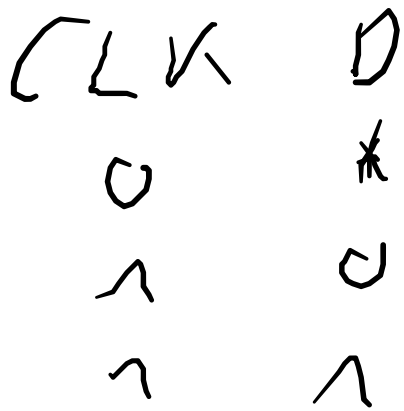
TECHNISCHE UNIVERSITÄT DARMSTADT



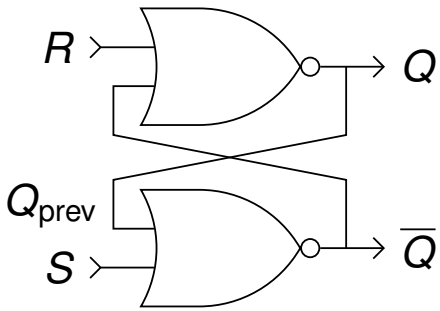


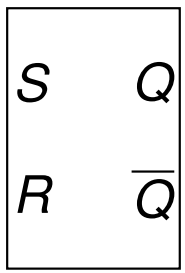


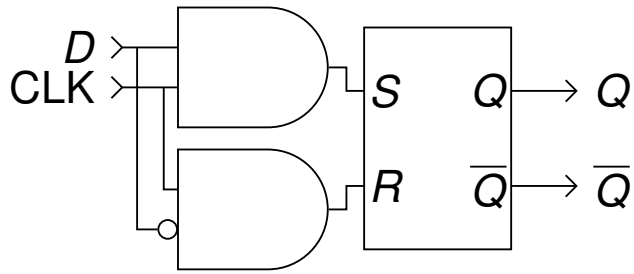


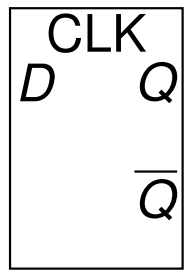


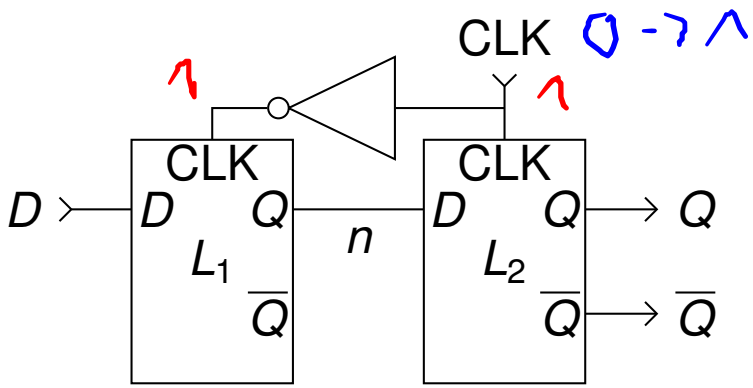
Wiederholung: Speicherelemente

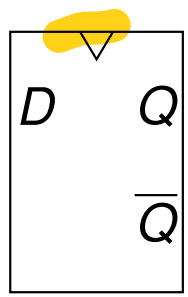






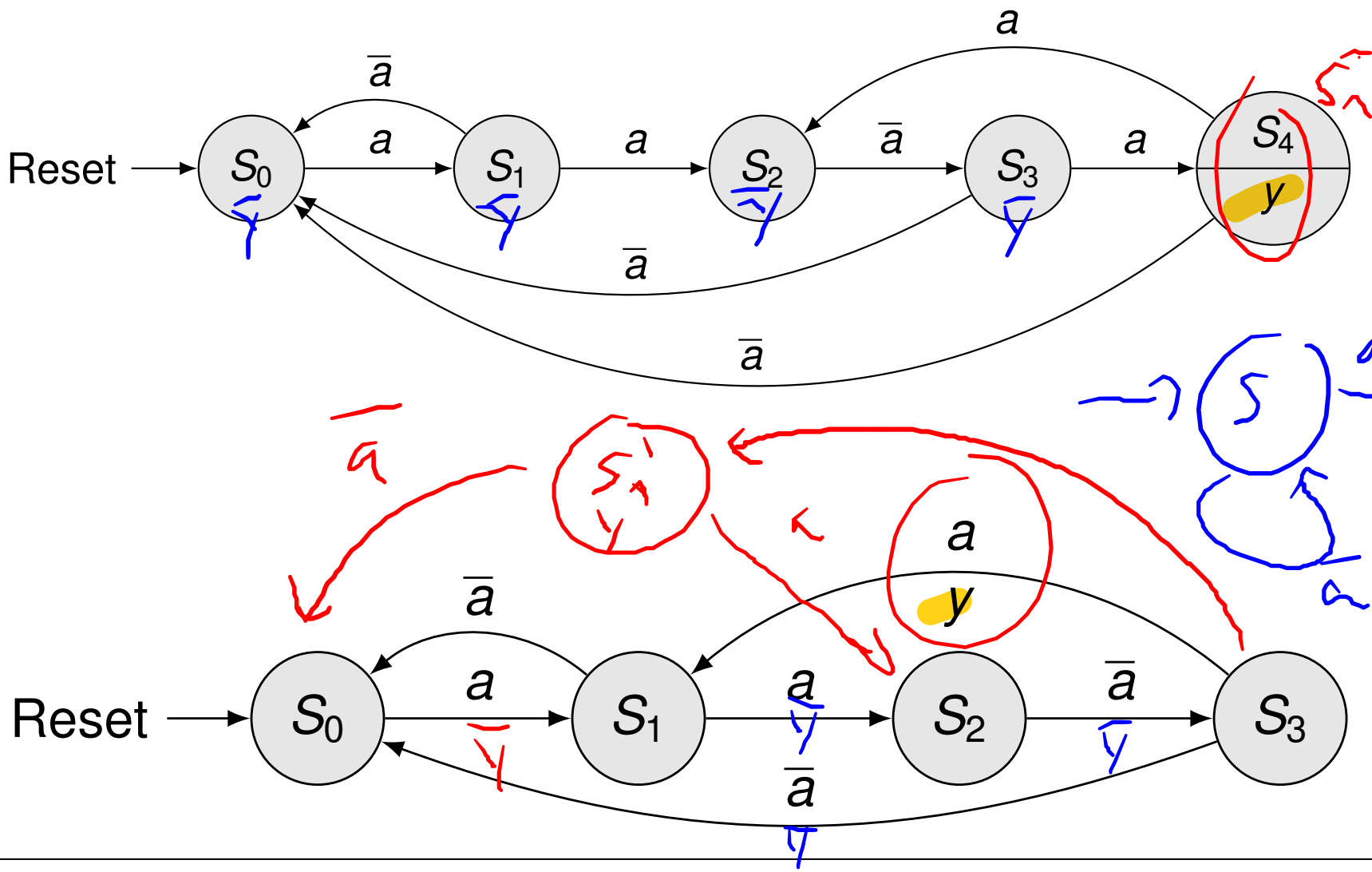






Wiederholung: Endliche Automaten
Notation, Überführung

TECHNISCHE UNIVERSITÄT DARMSTADT

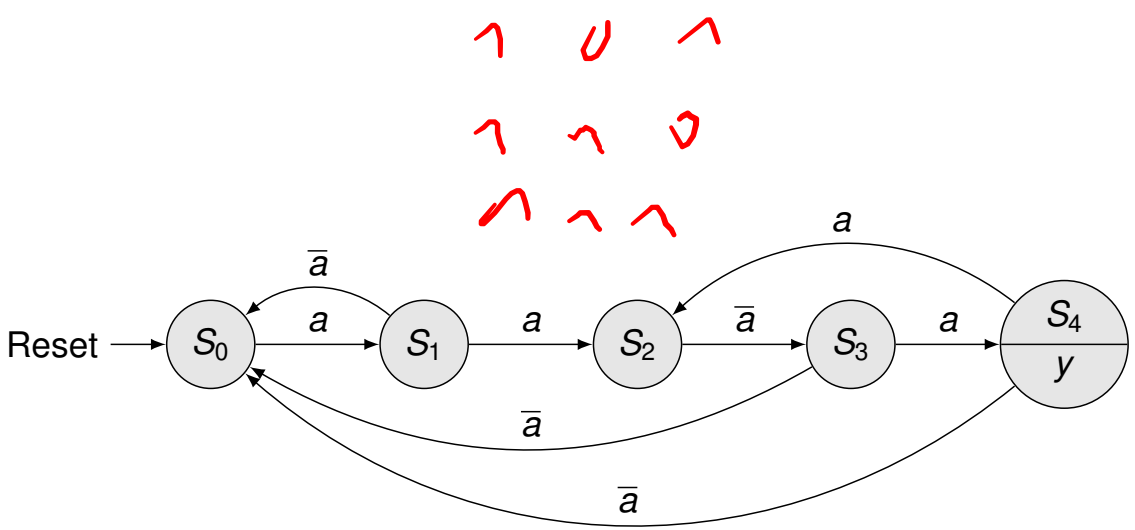


Wiederholung: Endliche Automaten
Don't Cares

TECHNISCHE UNIVERSITÄT DARMSTADT

S	a	S'
S_0	0	S_0
S_0	1	S_1
S_1	0	S_0
S_1	1	S_2
S_2	0	S_3
S_2	1	S_2
S_3	0	S_0
S_3	1	S_4
S_4	0	S_0
S_4	1	S_2

S	s_2	s_1	s_0
S_0	0	0	0
S_1	0	0	1
S_2	0	1	0
S_3	0	1	1
S_4	1	0	0

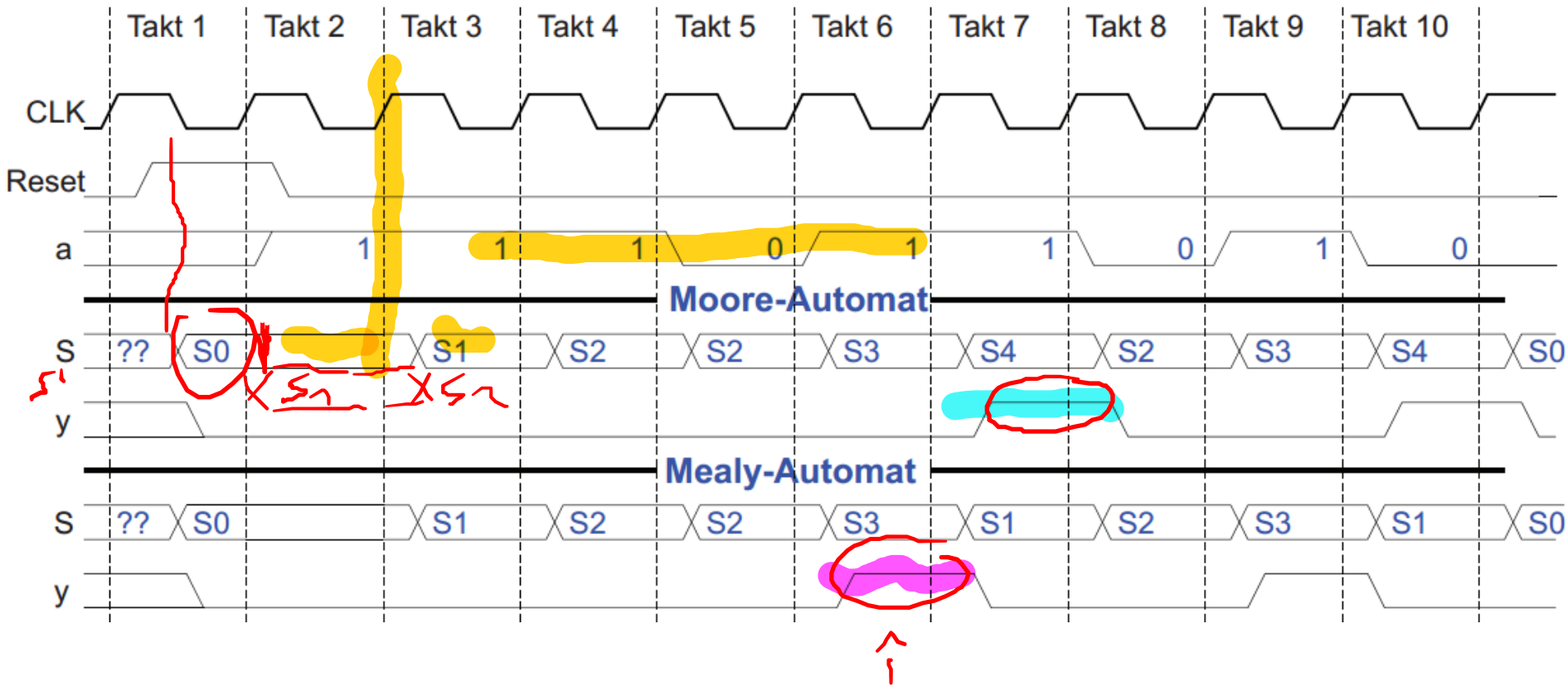


pattern/moore/state.esp

```
1  .i 4
2  .o 3
3  0000 000
4  0001 001
5  0010 000
6  0011 010
7  0100 011
8  0101 010
9  0110 000
10 0111 100
11 1000 000
12 1001 010
13 1010 ---
14 1011 ---
15 1100 ---
16 1101 ---
17 1110 ---
18 1111 ---
```

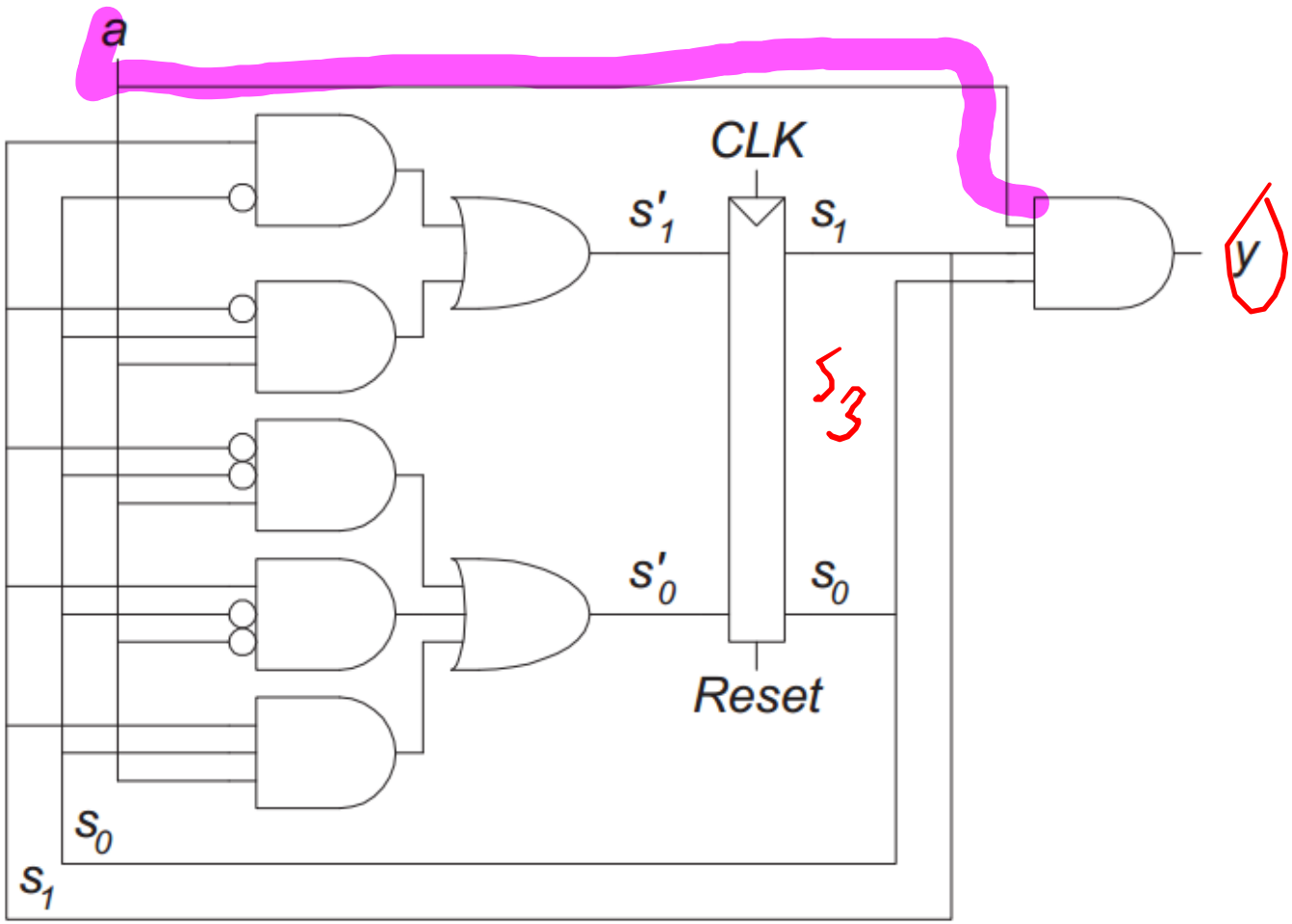
Wiederholung: Endliche Automaten
Mealy vs. Moore

TECHNISCHE UNIVERSITÄT DARMSTADT



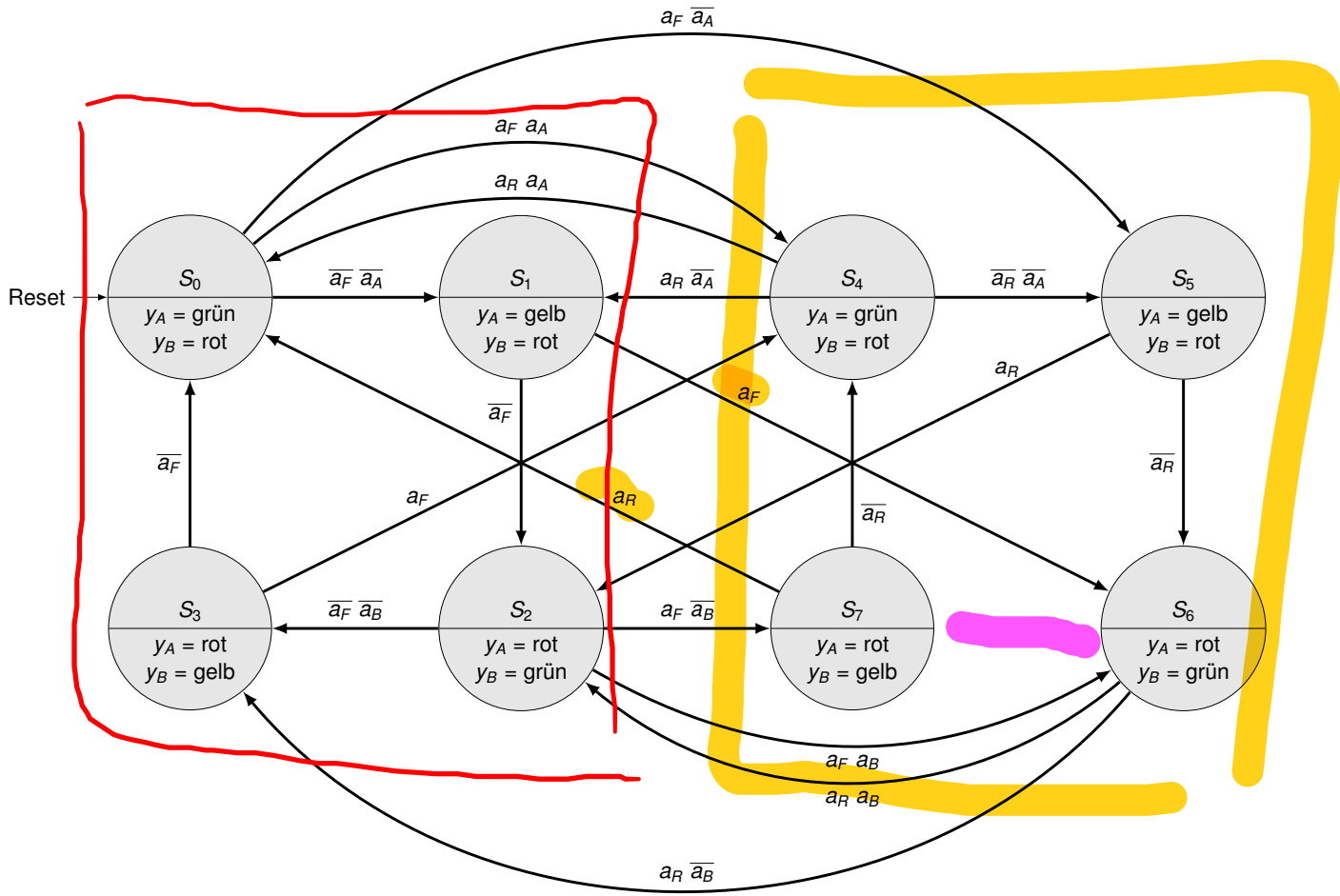
Wiederholung: Endliche Automaten
Mealy vs. Moore

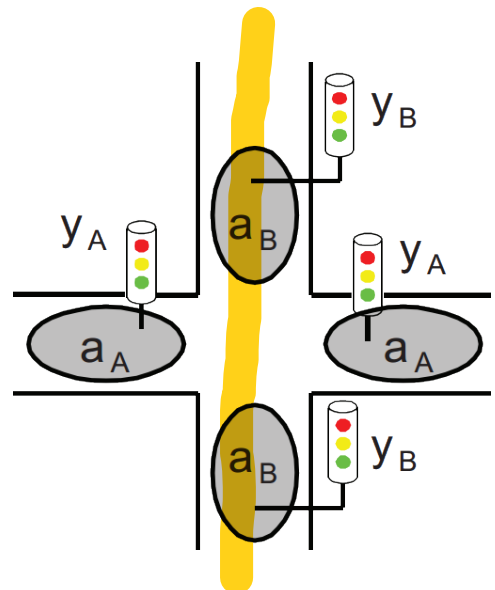
TECHNISCHE UNIVERSITÄT DARMSTADT



Wiederholung: Endliche Automaten
Zerlegung

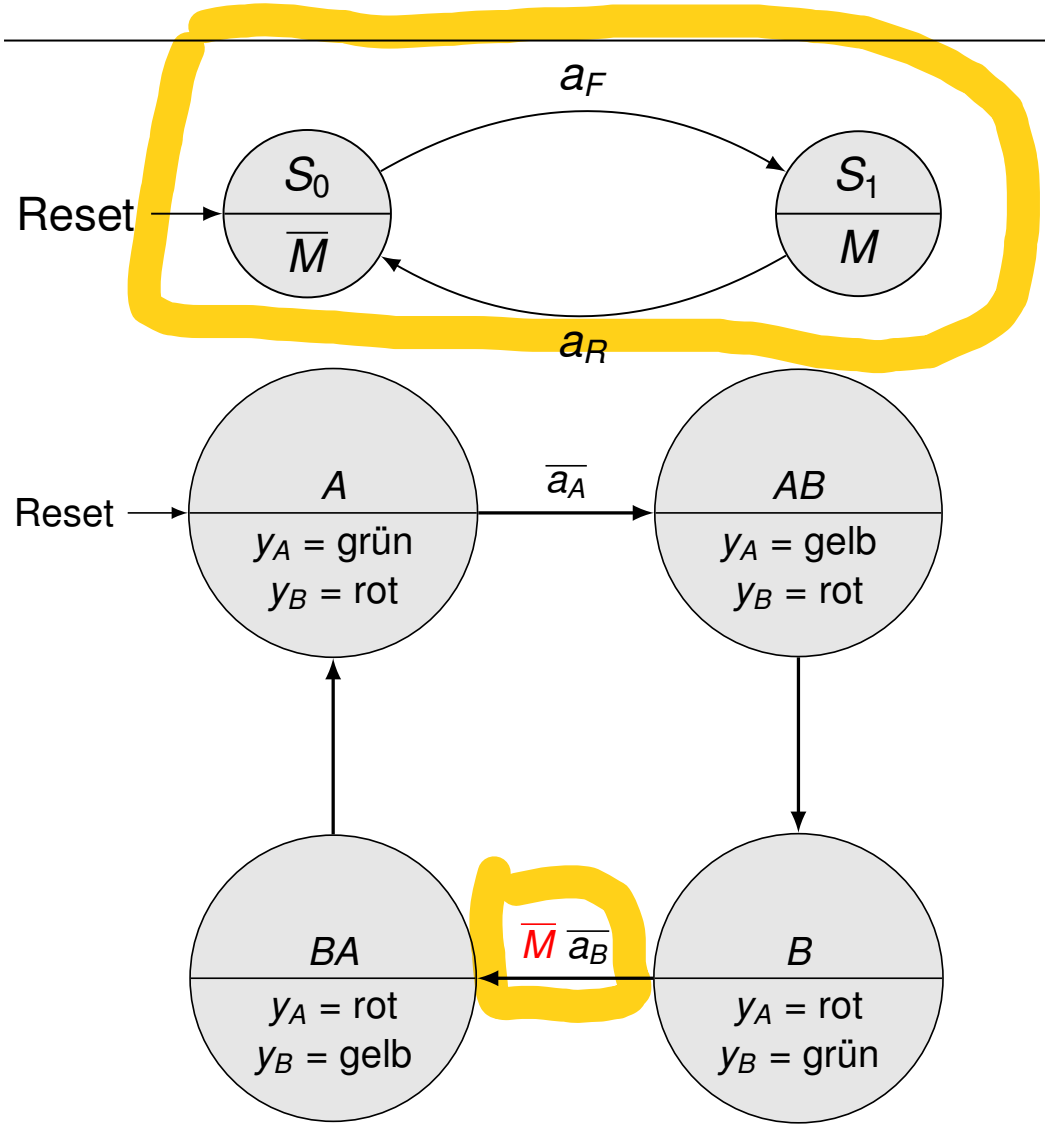
TECHNISCHE UNIVERSITÄT DARMSTADT

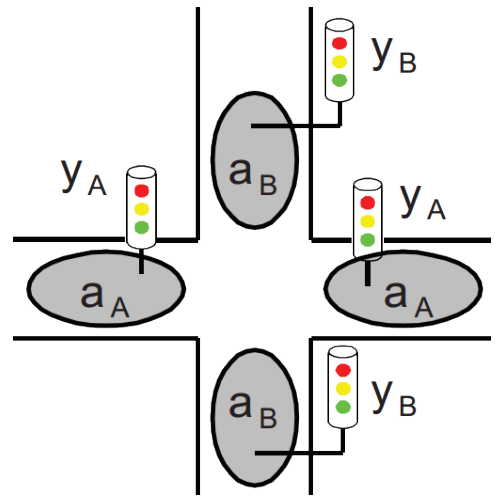




Wiederholung: Endliche Automaten
Zerlegung

TECHNISCHE UNIVERSITÄT DARMSTADT





Überblick der heutigen Vorlesung

TECHNISCHE UNIVERSITÄT DARMSTADT

- Zeitverhalten synchroner sequentieller Schaltungen
- Parallelität

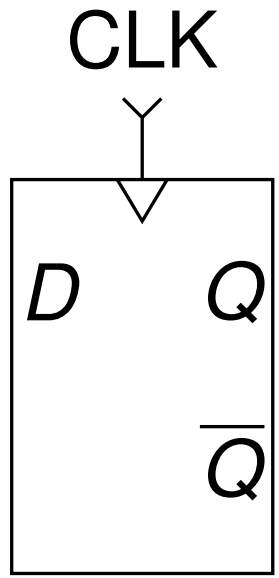
Harris 2013
Kap. 3.5-3.6
Seite 133 - 153

Zeitverhalten synchroner sequentieller Logik

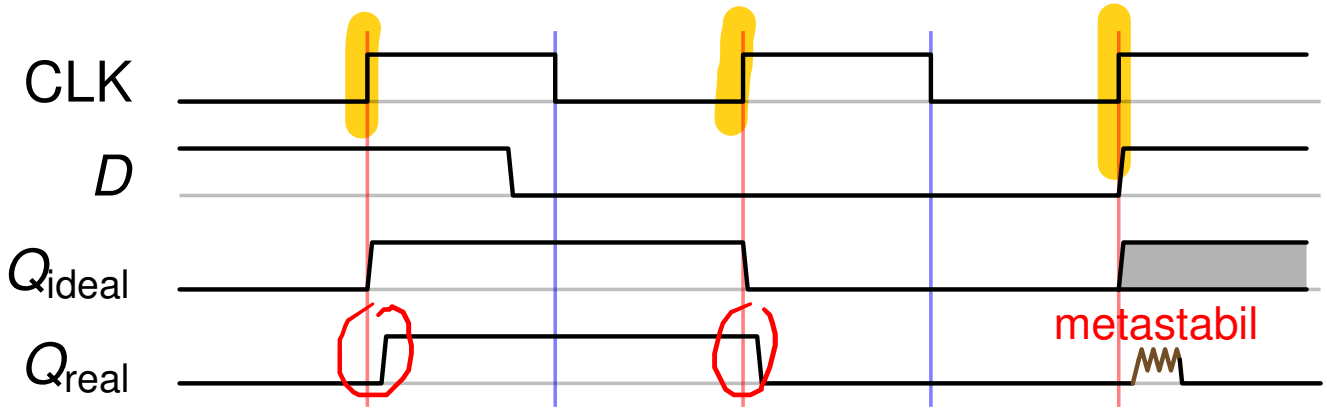
TECHNISCHE UNIVERSITÄT DARMSTADT

Zeitverhalten synchroner sequentieller Logik

TECHNISCHE UNIVERSITÄT DARMSTADT



- Flip-Flop übernimmt *D* zur steigenden Taktflanke
- Was passiert bei zeitgleicher Änderung von *D* und CLK?
- bisher vereinfachte Annahme:
  - Wert unmittelbar vor der Taktflanke wird übernommen
- Aber:
  - Was heißt „unmittelbar“?
  - Wie schnell wird neuer Zustand am Ausgang sichtbar?
  - Was muss daher bei synchronen sequentiellen Schaltungen beachtet werden?



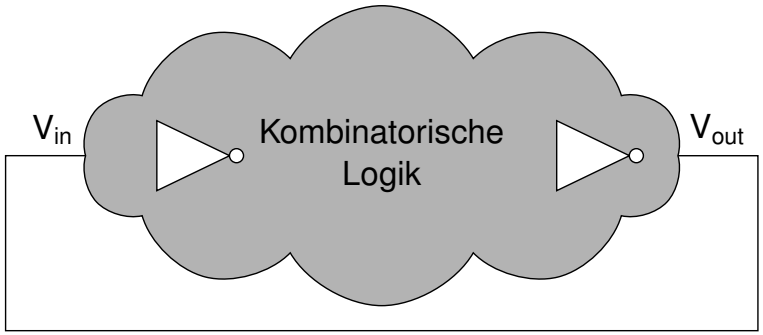
Analogie zur Fotografie: Bewegungsunschärfe (Bewegung während Belichtung)

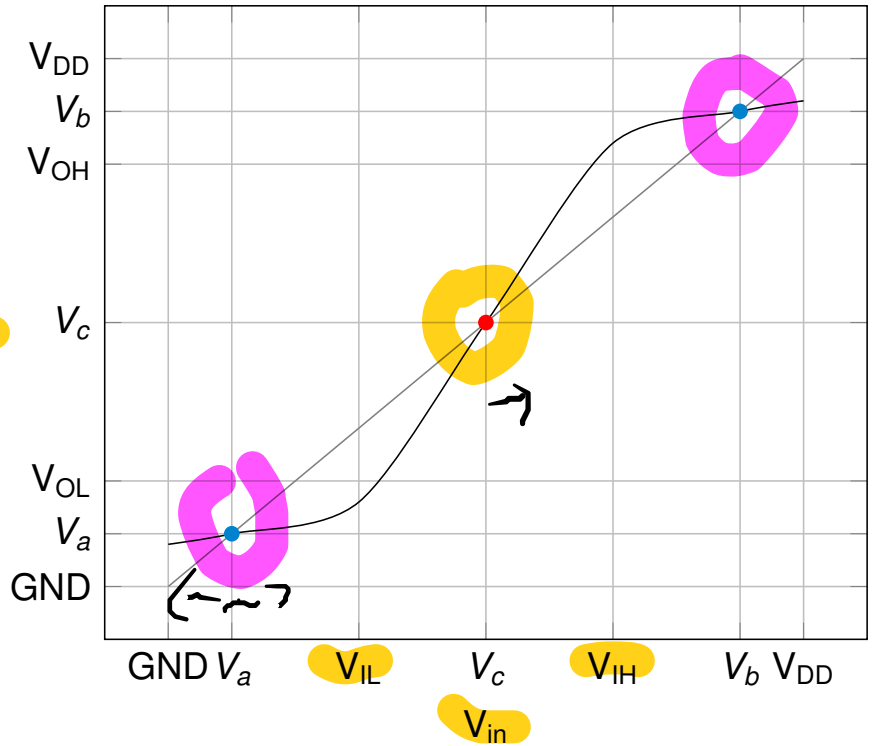
TECHNISCHE UNIVERSITÄT DARMSTADT

Quelle: `http://lightwatching.de`

Metastabilität durch Rückkopplung kombinatorischer Logik

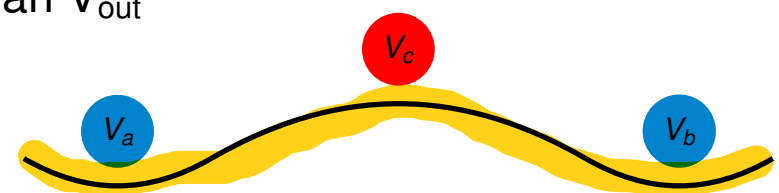
TECHNISCHE UNIVERSITÄT DARMSTADT





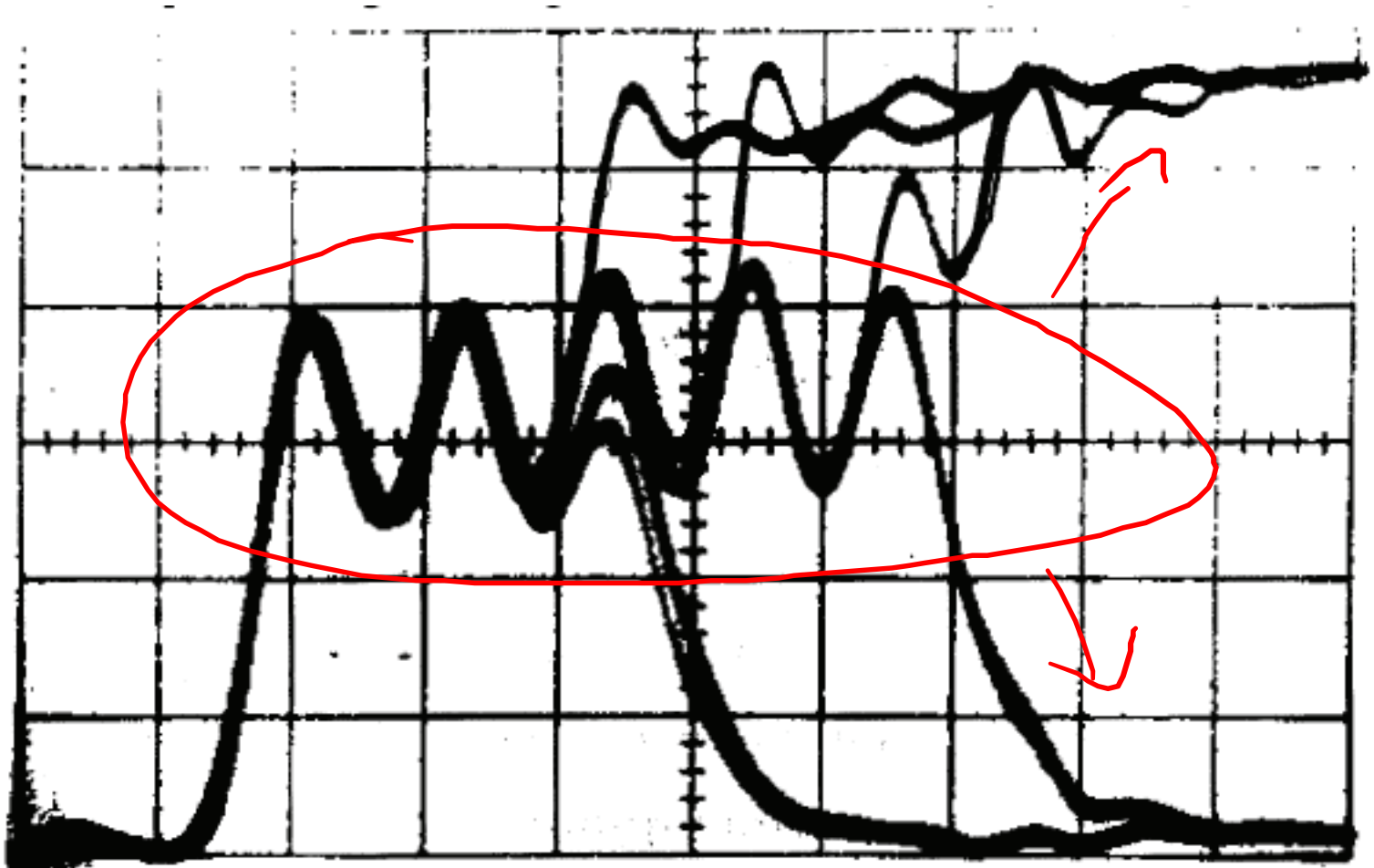
- Rückkoppling stabil für $V_{out} = V_{in}$
- Transferfunktion schneidet Hauptdiagonale in
  - V_a repräsentiert 0
  - V_b repräsentiert 1
  - V_c im „verbotenen“ Spannungsbereich
- V_c ist „metastabil“, da
  - kleine Änderung an V_{in} $\rightarrow$ große Änderung an V_{out}

$\Rightarrow$ geht nach zufälliger Verzögerung in einen stabilen Zustand über



Metastabilität an Flip-Flops

TECHNISCHE UNIVERSITÄT DARMSTADT



Quelle: Thomas. J. Chaney, Washington University

Zeitanforderungen an DFF Eingangssignale

TECHNISCHE UNIVERSITÄT DARMSTADT

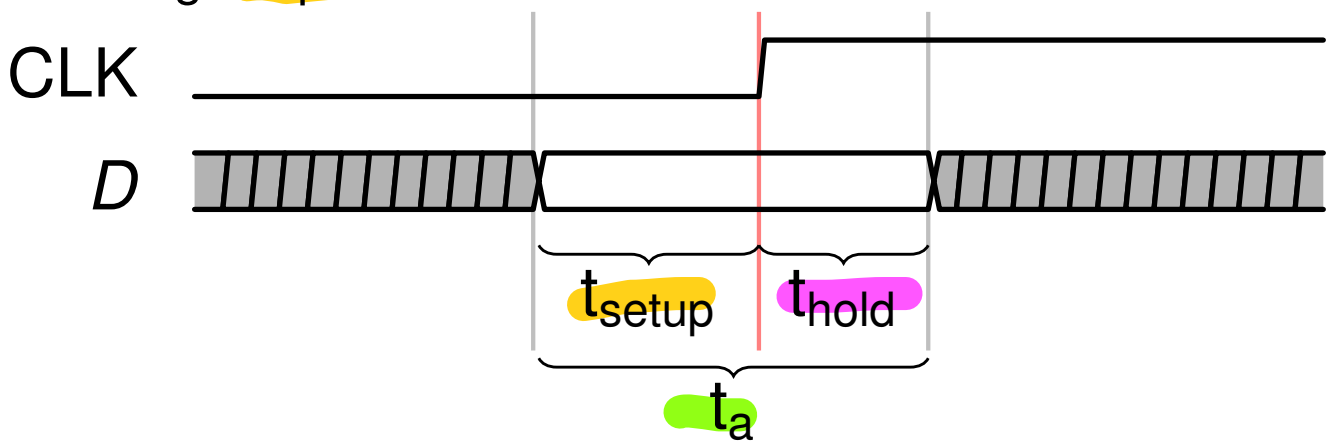
- Dateneingang *D* muss in Abtast-Zeitfenster um Taktflanke stabil sein, um Metastabilität zu vermeiden

t_{setup} Zeitintervall vor Taktflanke, in dem *D* stabil sein muss

t_{hold} Zeitintervall nach Taktflanke, in dem *D* stabil sein muss

t_a Abtastzeitfenster: $t_a = t_{setup} + t_{hold}$

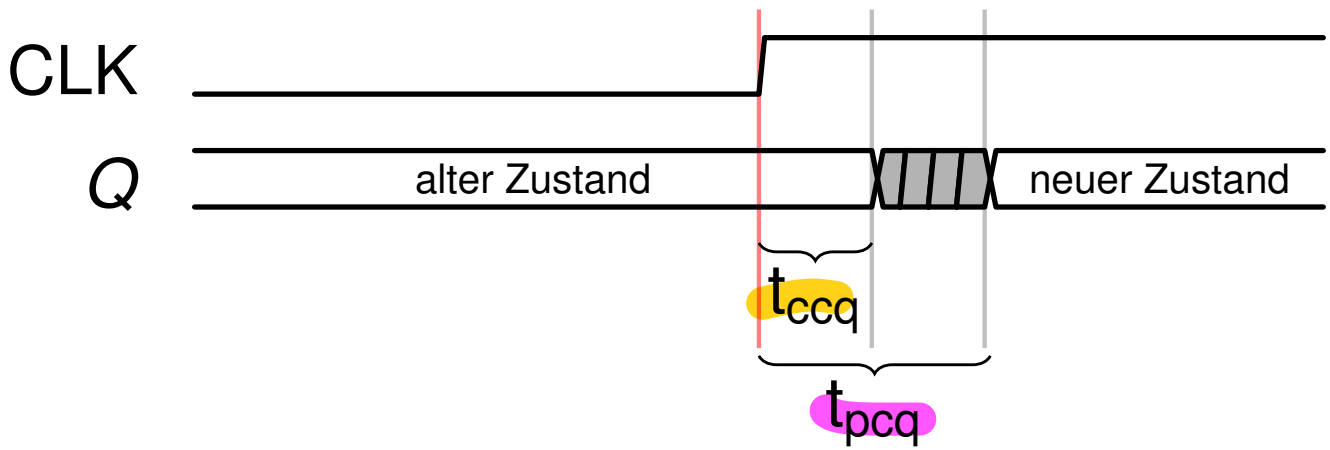
- Größenordnung: 10 ps



Zeitcharakteristik der DFF Ausgangssignale

TECHNISCHE UNIVERSITÄT DARMSTADT

- Verzögerung des Registerausgangs relativ zur steigenden Taktflanke
  - t_{ccq} Kontaminationsverzögerung, bis Q (frühestens) umschaltet
  - t_{pcq} Laufzeitverzögerung, bis Q (spätestens) stabil
- Größenordnung: 10 ps



Dynamische Entwurfsdisziplin

TECHNISCHE UNIVERSITÄT DARMSTADT

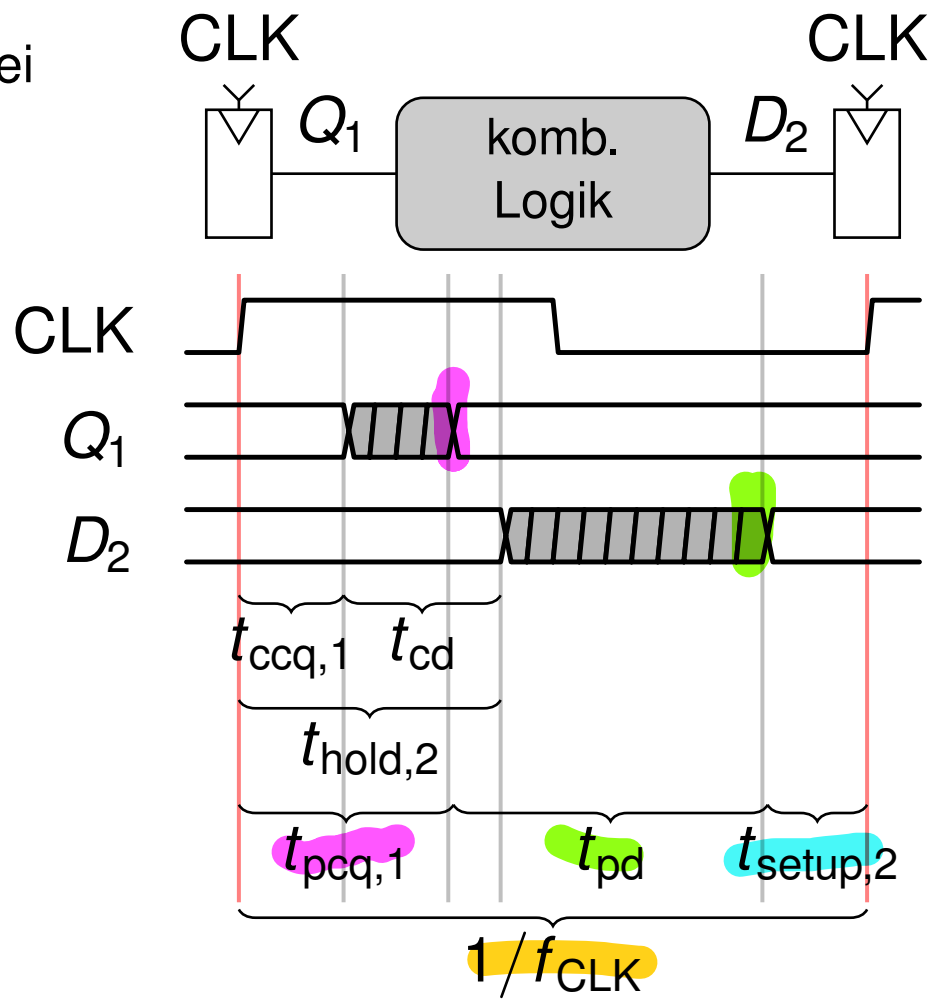
- kombinatorische Logik zwischen zwei Registern hat max/min Verzögerung
- abhängig von Verzögerungen der Gatter und des *ersten* Registers

$\Rightarrow$ Timing-Bedingungen des *zweiten* Registers müssen erfüllt werden

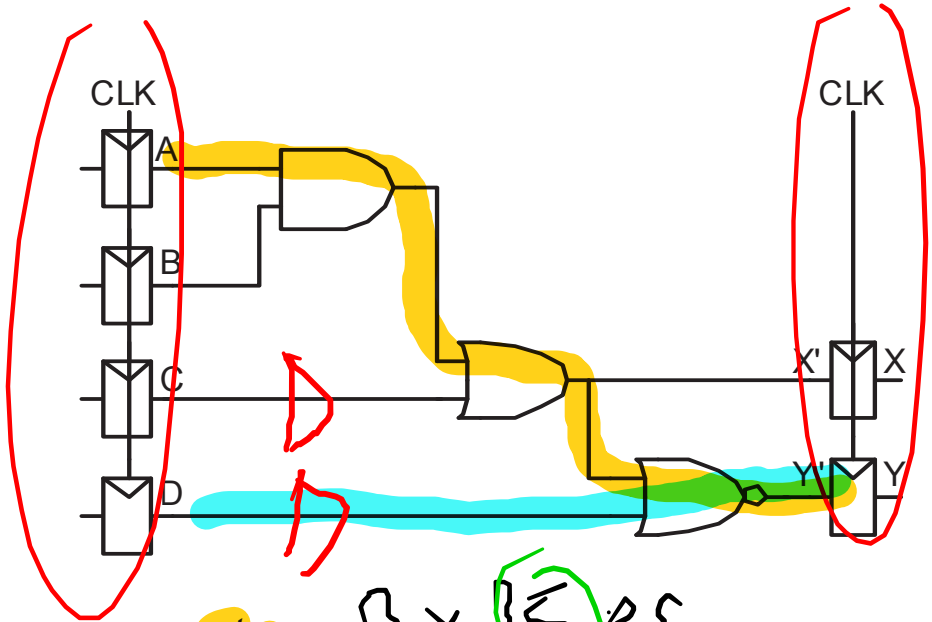
- $t_{ccq,1} + t_{cd} > t_{hold,2}$
- $t_{pcq,1} + t_{pd} + t_{setup,2} \leq \frac{1}{f_{CLK}}$

$\Rightarrow$ maximale Taktrate wird durch *kritischen Pfad* bestimmt

- $f_{CLK} \leq \frac{1}{t_{pcq}+t_{pd}+t_{setup}}$



Analyse des Zeitverhaltens



Verzögerungsangaben

t_{ccq} = 30 ps

t_{pcq} = 50 ps

t_{setup} = 60 ps

t_{hold} = 70 ps

Pro Gatter

t_{pd} = 35 ps

t_{cd} = 25 ps

t_{pd} = 3 × 35 ps

t_{cd} = 1 × 25 ps

Einhalten von Setup-Zeitanforderung:

$T_c \geq$ 50 ps + 35 ps + 60 ps

$f_c = 1/T_c =$ 145

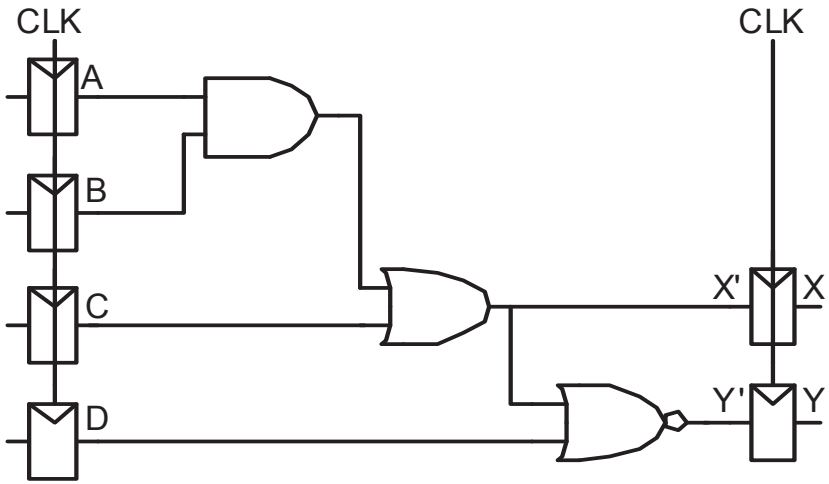
Einhalten von Hold-Zeitanforderung:

$t_{ccq} + t_{cd} > t_{hold}$?

30 ps + 25 ps = 55 ps

Analyse des Zeitverhaltens

TECHNISCHE UNIVERSITÄT DARMSTADT



Verzögerungsangaben

t_{ccq} = 30 ps

t_{pcq} = 50 ps

t_{setup} = 60 ps

t_{hold} = 70 ps

Pro Gatter:

t_{pd} = 35 ps

t_{cd} = 25 ps

t_{pd} = 3 x 35 ps = 105 ps

t_{cd} = 25 ps

Einhalten der Setup-Zeitanforderung:

$T_c \geq (50 + 105 + 60)$ ps = 215 ps

$f_c = 1/T_c$ = 4,65 GHz

Einhalten der Hold-Zeitanforderung:

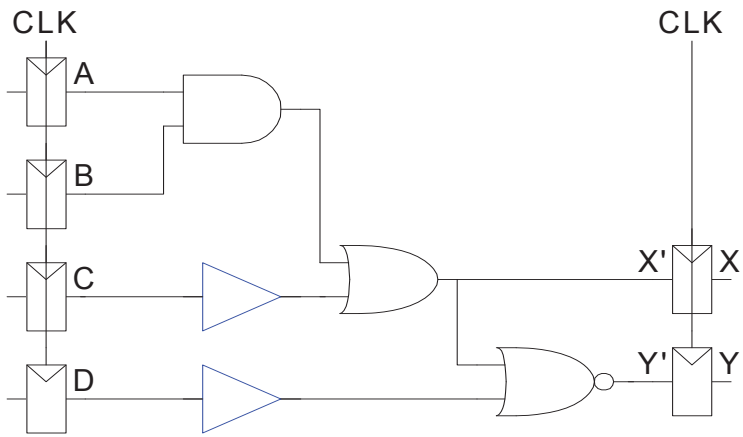
$t_{ccq} + t_{cd} > t_{hold}$?

(30 + 25) ps > 70 ps ? Nein, verletzt!

Beheben der verletzten Hold-Zeitanforderung

TECHNISCHE UNIVERSITÄT DARMSTADT

Füge Puffer in zu kurze Pfade ein!



Verzögerungsangaben

t_{ccq} = 30 ps

t_{pcq} = 50 ps

t_{setup} = 60 ps

t_{hold} = 70 ps

Pro Gatter:

t_{pd} = 35 ps

t_{cd} = 25 ps

t_{pd} =

t_{cd} =

Einhalten der Setup-Zeitanforderung:

$T_c \geq$

f_c =

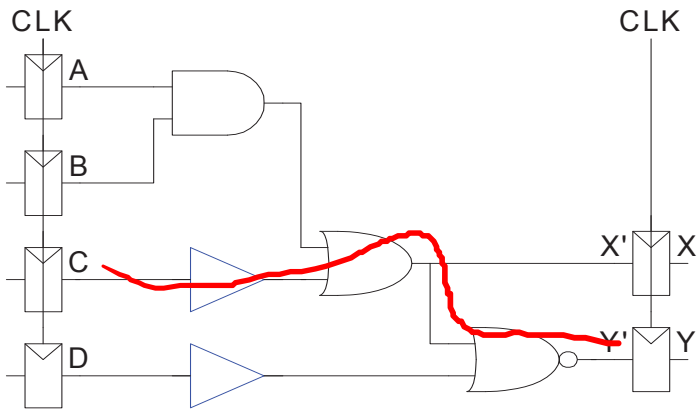
Einhalten der Hold-Zeitanforderung:

$t_{ccq} + t_{cd} > t_{hold}$?

Beheben der verletzten Hold-Zeitanforderung

TECHNISCHE UNIVERSITÄT DARMSTADT

Füge Puffer in zu kurze Pfade ein!



Verzögerungsangaben

t_{ccq} = 30 ps

t_{pcq} = 50 ps

t_{setup} = 60 ps

t_{hold} = 70 ps

Pro Gatter:

t_{pd} = 35 ps

t_{cd} = 25 ps

t_{pd} = 3 x 35 ps = 105 ps

t_{cd} = 2 x 25 ps = 50 ps

Einhalten der Setup-Zeitanforderung:

$T_c \geq (50 + 105 + 60)$ ps = 215 ps

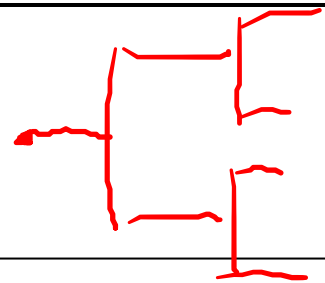
$f_c = 1/T_c$ = 4.65 GHz

Einhalten der Hold-Zeitanforderung:

$t_{ccq} + t_{cd} > t_{hold}$?

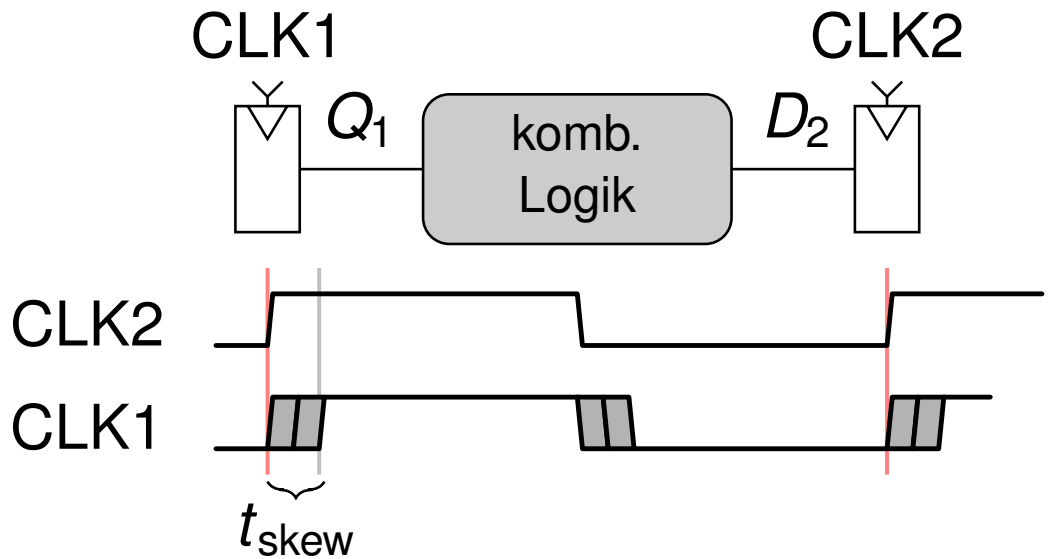
(30 + 50) ps > 70 ps ? Ja, eingehalten!

Taktverschiebung (clock skew)



TECHNISCHE UNIVERSITÄT DARMSTADT

- Takt kommt nicht bei allen Registern gleichzeitig an
  - unterschiedliche Verdrahtungswege auf dem Chip (clock tree),
  - Logik in Taktsignal (bspw. gated clock)
- t_{skew} ist max. Differenz der Taktankunftszeit zwischen zwei Registern



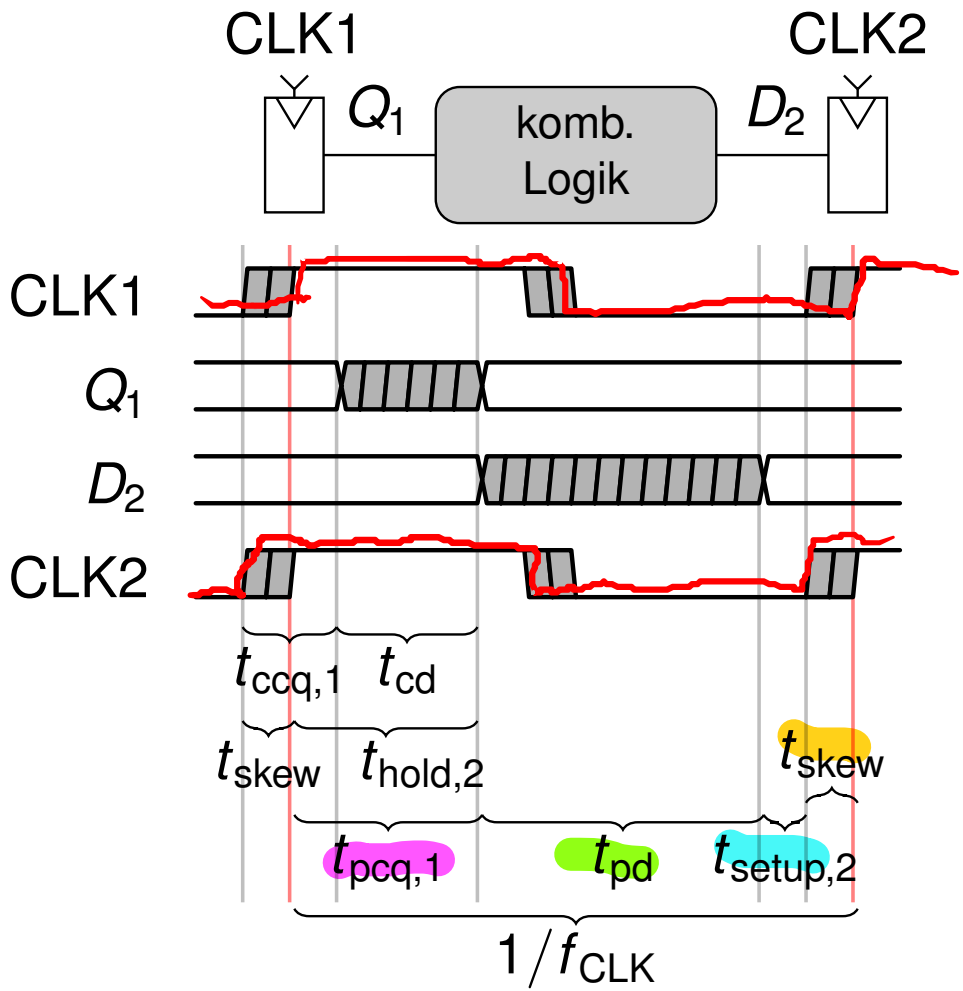
Timing-Bedingungen mit Taktverschiebung

TECHNISCHE UNIVERSITÄT DARMSTADT

- Timing-Bedingungen müssen auch im worst-case eingehalten werden:
  - $t_{ccq,1} + t_{cd} > t_{skew} + t_{hold,2}$
  - $t_{pcq,1} + t_{pd} + t_{setup,2} + t_{skew} \leq \frac{1}{f_{CLK}}$

$\Rightarrow$ idR. wird Timing durch t_{skew} enger

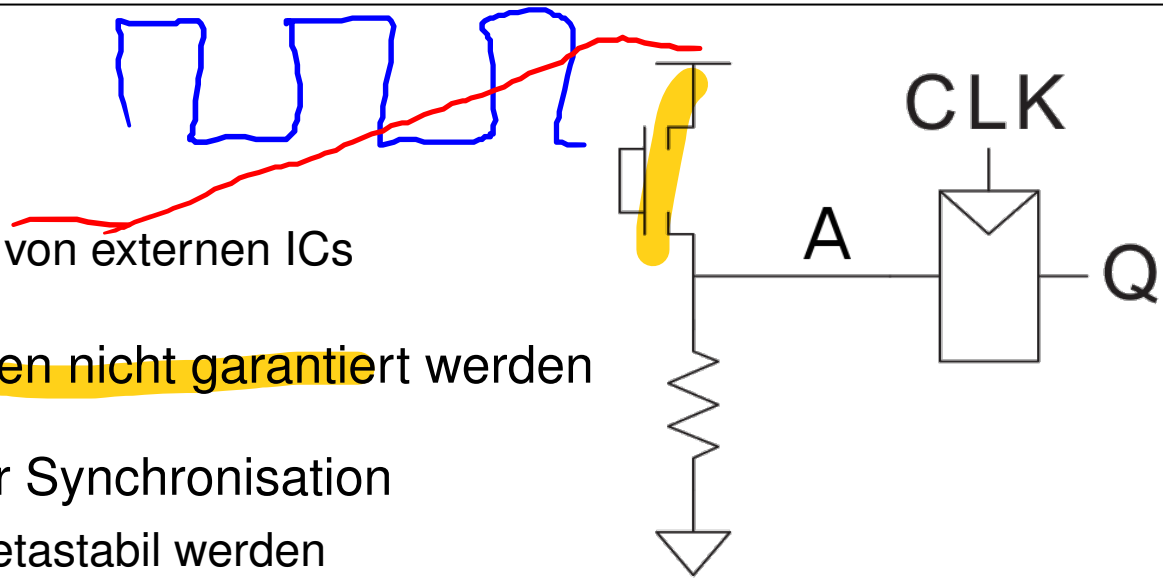
- Taktfrequenz kann durch t_{skew} auch steigen, wenn CLK2 sicher nach CLK1 schaltet



Verletzung der dynamischen Entwurfsdisziplin

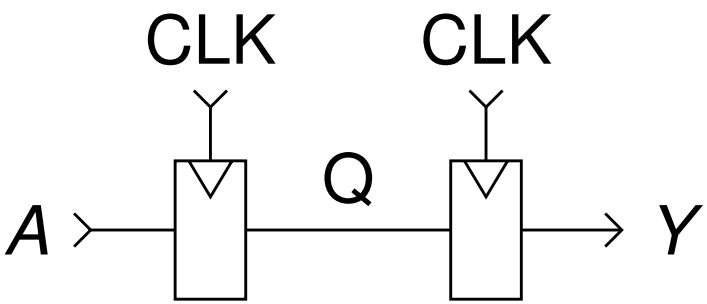
TECHNISCHE UNIVERSITÄT DARMSTADT

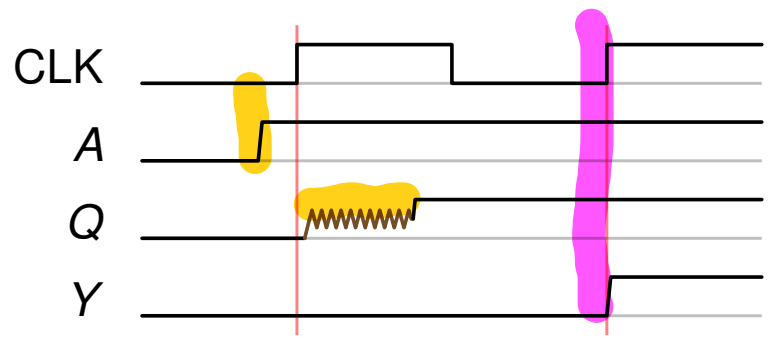
- asynchronen Eingänge:
  - Benutzereingaben
  - Kommunikationssignale von externen ICs



$\Rightarrow$ Timing-Bedingungen können nicht garantiert werden

- zweifaches Shiftregister für Synchronisation
  - erstes Flip-Flop kann metastabil werden
  - kippt idR vor nächster Taktflanke in stabilen Zustand
  - zweites Flip-Flop wird nicht metastabil





Parallelität

TECHNISCHE UNIVERSITÄT DARMSTADT

Arten der Parallelität

TECHNISCHE UNIVERSITÄT DARMSTADT

- räumliche Parallelität
  - mehrere Aufgaben durch vervielfachte Hardware gleichzeitig bearbeiten
- zeitliche Parallelität
  - Aufgabe in mehrere Unteraufgaben aufteilen
  - Unteraufgaben parallel ausgeführen
- Beispiel: Fließbandprinzip bei Autofertigung („Pipelining“)
  - nur eine Station für pro Arbeitsschritt
  - alle unterschiedlichen Arbeitsschritte für mehrere Autos werden parallel ausgeführt

  $\Rightarrow$ zeitliche Paralellität

Grundlegende Begriffe

TECHNISCHE UNIVERSITÄT DARMSTADT

- Datensatz:
  - Vektor aus Eingabewerten, die zu Vektor aus Ausgabewerten bearbeitet werden
- Latenz:
  - Zeit von der Eingabe eines Datensatzes bis Ausgabe der zugehörigen Ergebnisse
- Durchsatz
  - Anzahl von Datensätzen, die pro Zeiteinheit bearbeitet werden können

$\Rightarrow$ Parallelität erhöht Durchsatz

Beispiel Parallelität
Plätzchen backen

TECHNISCHE UNIVERSITÄT DARMSTADT

- Weihnachtszeit steht vor der Tür, also rechtzeitig anfangen!
- Annahmen
  - genug Teig ist fertig
  - 5 Minuten um ein Blech mit Teig zu bestücken
  - 15 Minuten Backzeit
- Vorgehensweise
  - ein Blech nach dem anderen vorbereiten und backen
  - Latenz = 20 min
  - Durchsatz = 3/h

Beispiel Parallelität
Plätzchen backen

TECHNISCHE UNIVERSITÄT DARMSTADT

- Weihnachtszeit steht vor der Tür, also rechtzeitig anfangen!
- Annahmen
  - genug Teig ist fertig
  - 5 Minuten um ein Blech mit Teig zu bestücken
  - 15 Minuten Backzeit
- Vorgehensweise
  - ein Blech nach dem anderen vorbereiten und backen
  - Latenz = 5 + 15 = 20 Minuten = 1/3 h
  - Durchsatz = 1 Blech alle 20 Minuten = 3 Bleche/h

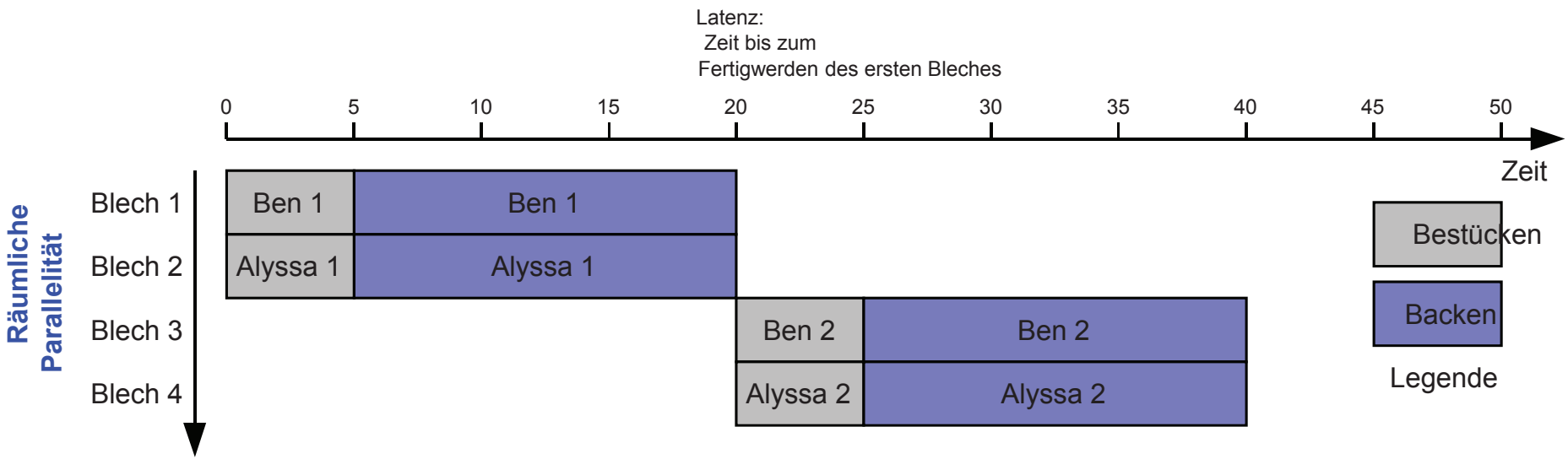
Beispiel Parallelität
Plätzchen backen (parallel)

TECHNISCHE UNIVERSITÄT DARMSTADT

- gleiche Annahmen wie eben
  - 5 Minuten Blech bestücken, 15 Minuten Backen
- Alternative Vorgehensweisen
  - Räumliche Parallelität: Zwei Bäcker (Ben & Alyssa), jeder mit einem eigenen Ofen
  - Zeitliche Parallelität: Aufteilen der Keksherstellung in Unteraufgaben
    - Blech bestücken
    - Backen
    - Nächstes Blech bestücken, während erstes noch im Ofen gebacken wird

$\Rightarrow$ Latenz und Durchsatz?

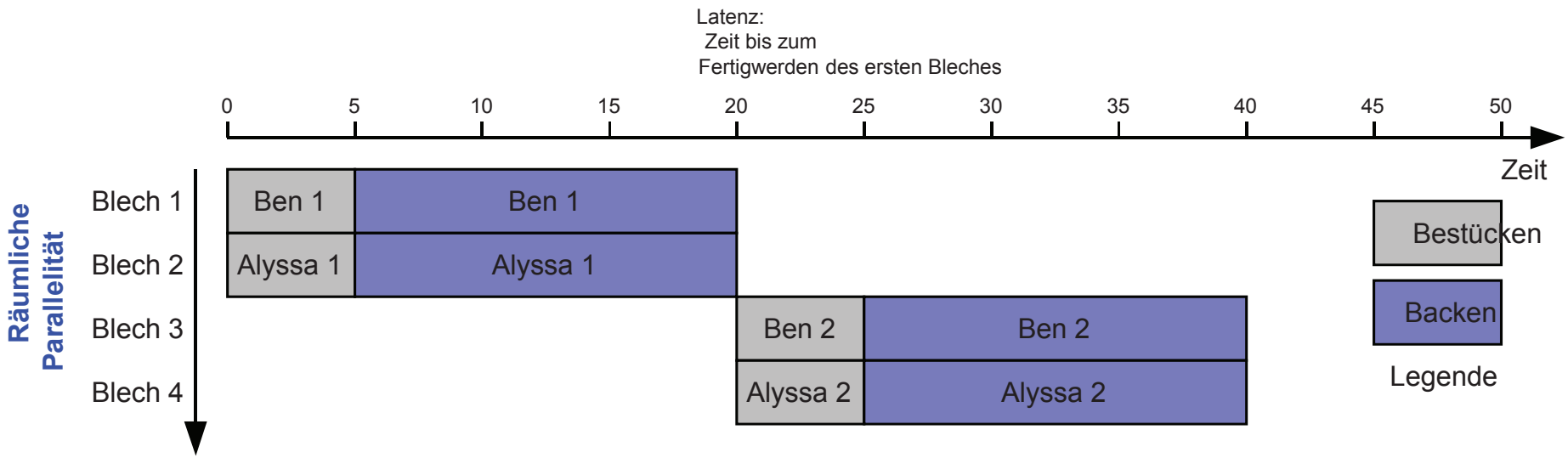
Räumliche Parallelität



Latenz = 20 min

Durchsatz = 6 / h

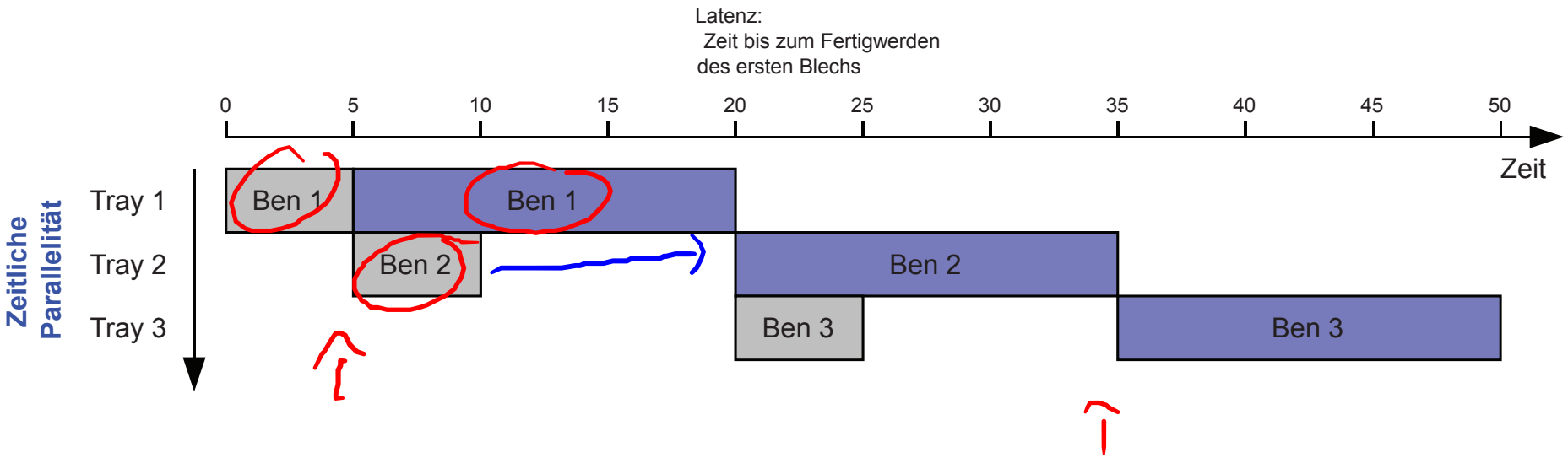
Räumliche Parallelität



Latenz = 5 + 15 = 20 Minuten = 1/3 h

Durchsatz = 2 Bleche alle 20 Minuten = 6 Bleche/h

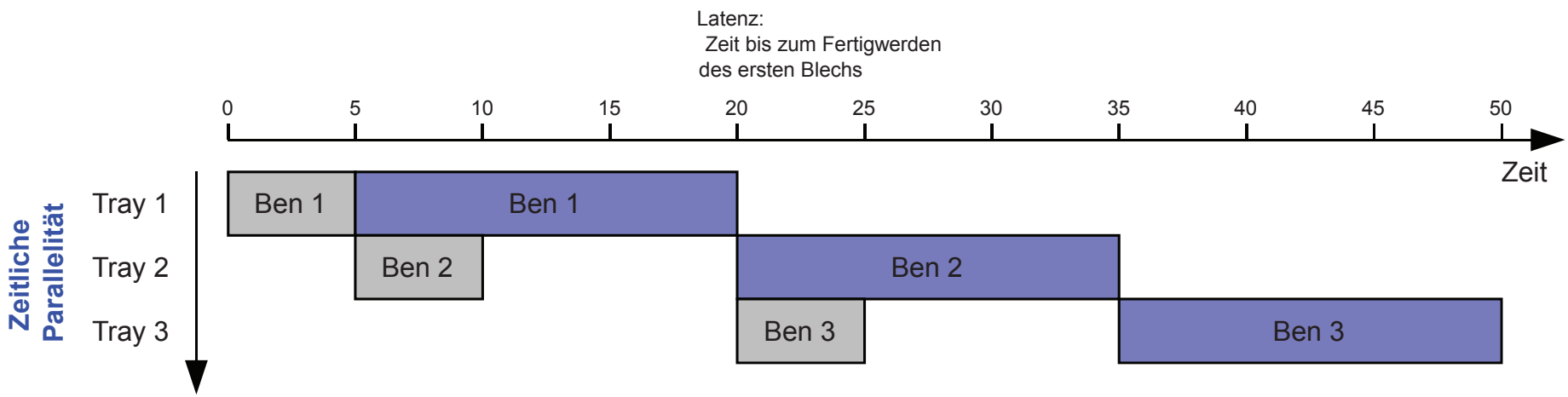
Zeitliche Parallelität



Latenz = 30 min

Durchsatz = 4 h

Zeitliche Parallelität



Latenz = 15 + 15 = 30 Minuten = 1/2 h

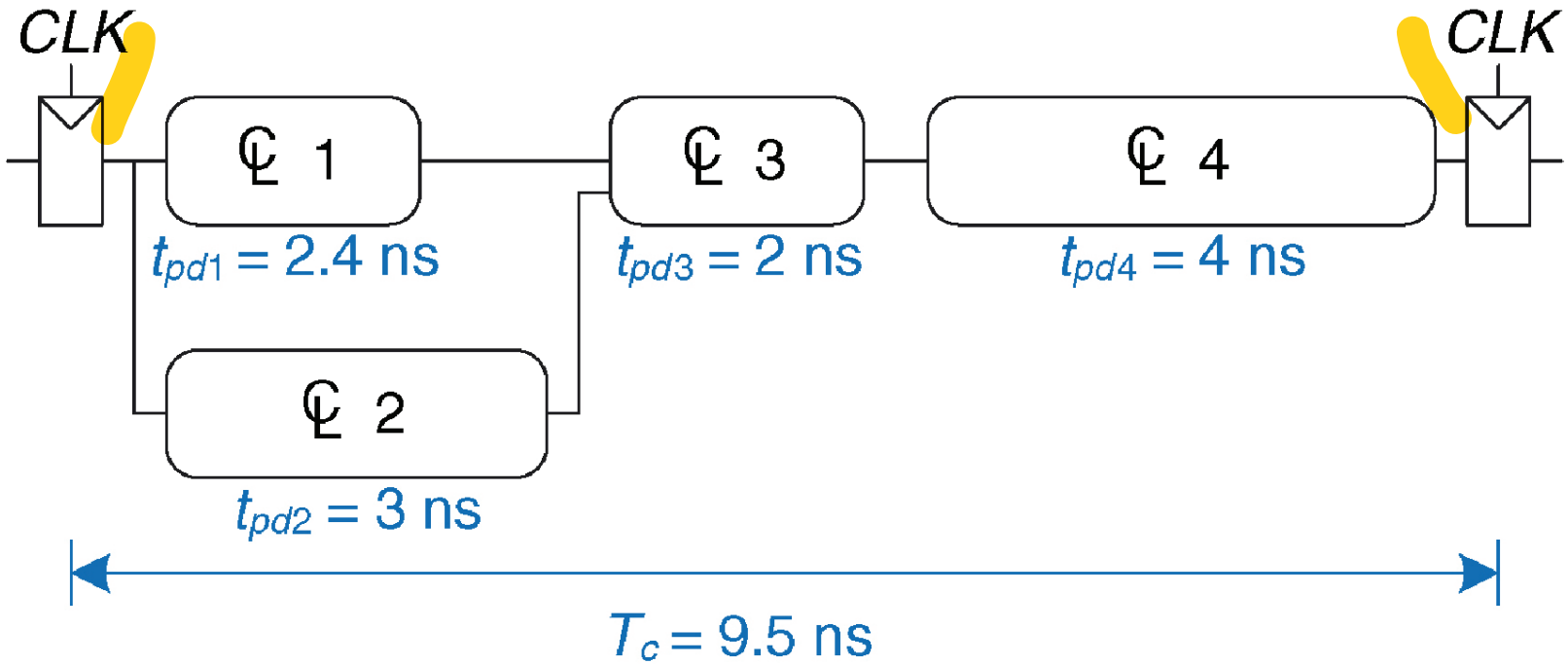
Durchsatz = 1 Blech alle 15 Minuten = 4 Bleche/h

Kombination der Parallelitäten

TECHNISCHE UNIVERSITÄT DARMSTADT

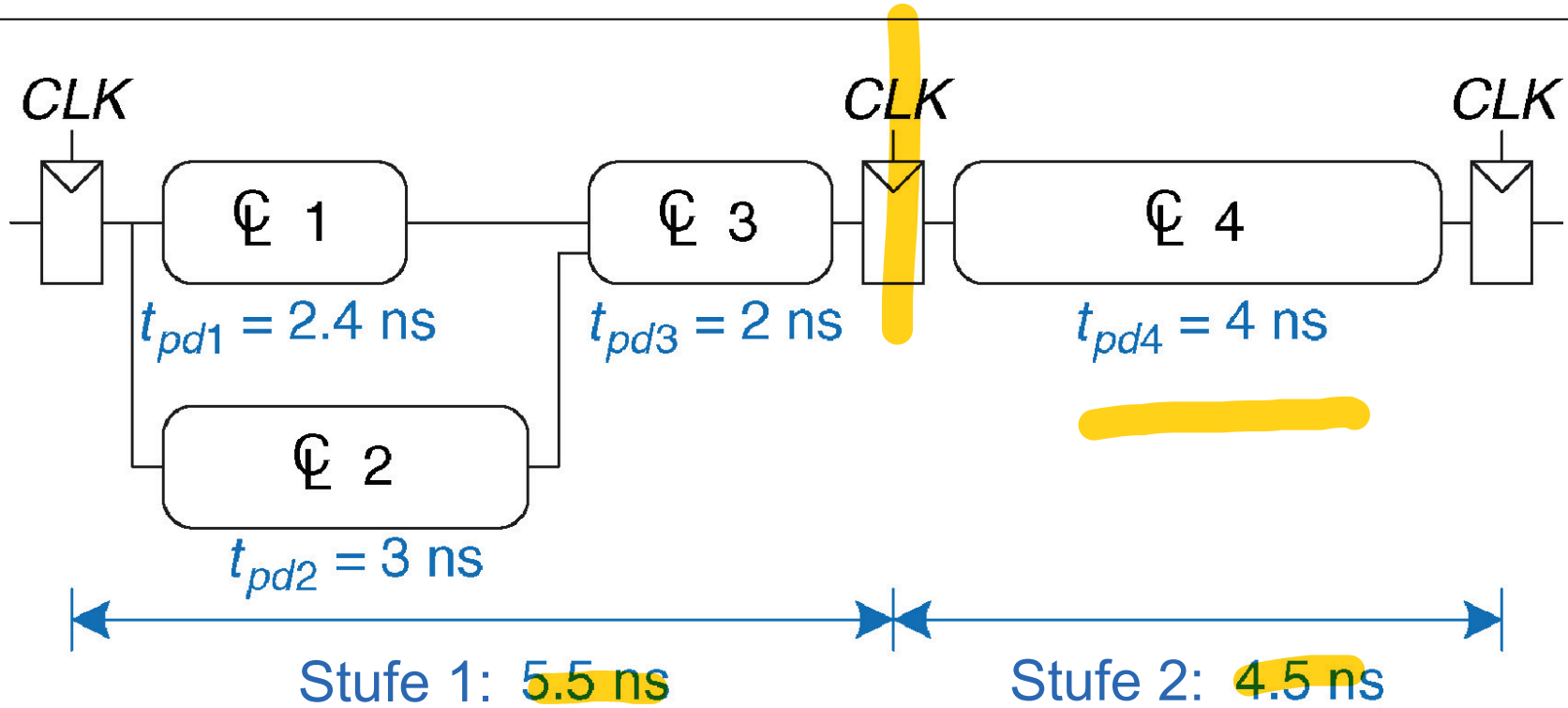
- Zeitliche und räumliche Parallelität können miteinander kombiniert werden
- Hier:
  - Zwei Bäcker und Öfen
  - Nächstes Blech bestücken während altes gebacken wird
- Latenz = 30 Minuten
- Durchsatz = 8 Bleche/h

Schaltung ohne Pipelining



- Kritischer Pfad durch Elemente 2, 3, 4: 9 ns
- t_{setup} = 0,2 ns und t_{pcq} =0,3 ns ➔ T_c = 9+0,2+0,3 = 9,5 ns
- Latenz = 9,5ns ; Durchsatz = 1 / 9,5ns = 105 MHz

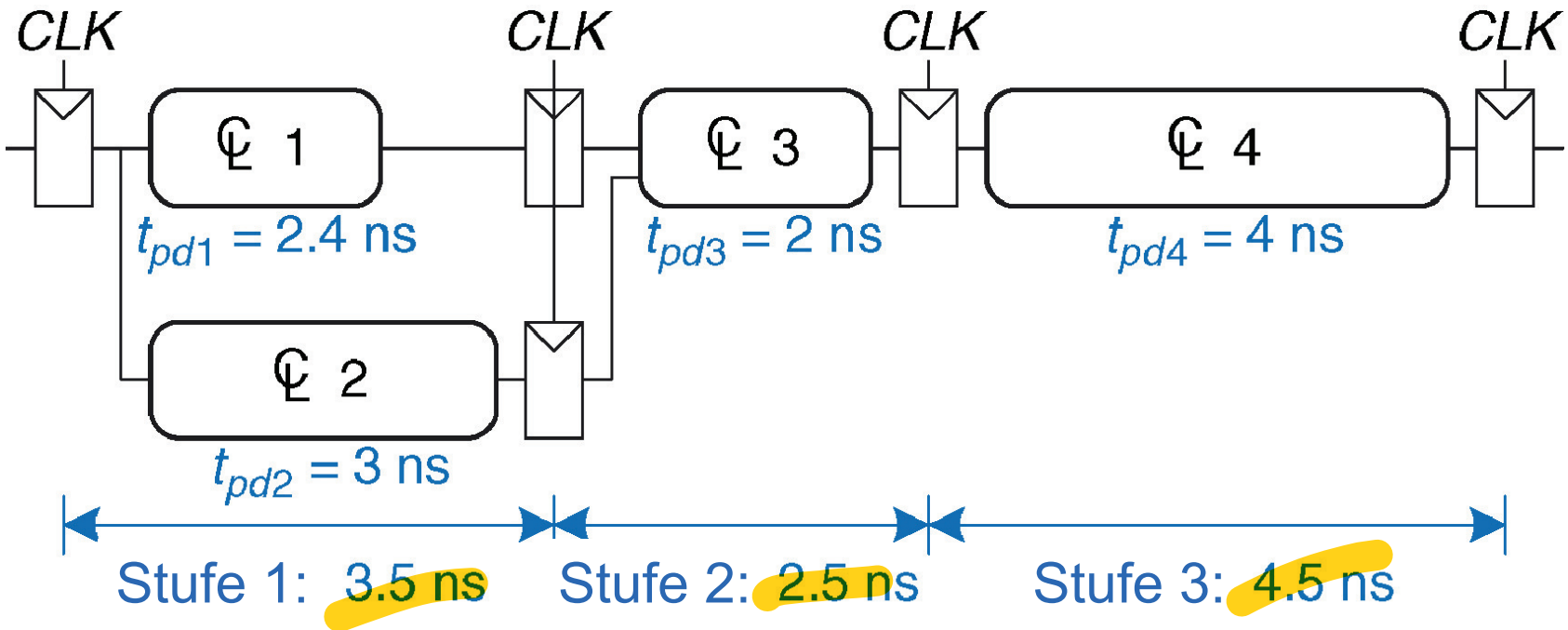
Schaltung mit (mehrstufiger) Pipeline



- Stufe 1: 3+2+0,2+0,3 = 5,5 ns
- ➔ T_c = 5,5 ns
- Latenz = 2 Takte = 11 ns
- Durchsatz = 1 / 5,5 ns = 182 MHz

Stufe 2: 4+0,2+0,3 = 4,5 ns

Schaltung mit (mehrstufiger) Pipeline



- T_c = 4,5ns
- Latenz = 3 Takte = 13,5 ns
- Durchsatz = 1 / 4,5 ns = 222 MHz

Bewertung Pipelining

TECHNISCHE UNIVERSITÄT DARMSTADT

- mehr Pipelinestufen
  - höherer Durchsatz (mehr Ergebnisse pro Zeiteinheit)
  - aber auch höhere Latenz (länger warten auf das Ergebnis)

  $\Rightarrow$ lohnt sich nur, wenn viele Datensätze bearbeitet werden müssen
- Klappt aber nicht immer
- Problem: Abhängigkeiten
- Beispiel Kekse: Erstmal schauen wie ein Blech geworden ist, bevor das nächste bestückt wird
- wird noch intensiv beim Thema Befehlsverarbeitung von Prozessoren behandelt

Zusammenfassung

TECHNISCHE UNIVERSITÄT DARMSTADT

Zusammenfassung und Ausblick

TECHNISCHE UNIVERSITÄT DARMSTADT

- Zeitverhalten synchroner sequentieller Schaltungen
- Parallelität

- Nächste Vorlesung behandelt
  - Hardwarebeschreib mit SystemVerilog