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 KW 50 im Moodle

Rückblick auf die letzten Vorlesungen

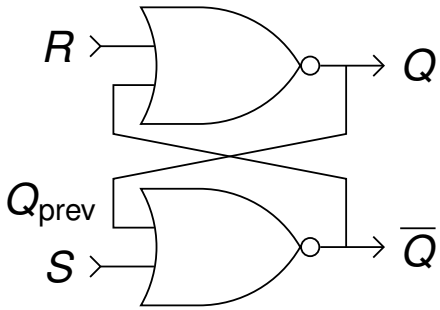
TECHNISCHE UNIVERSITÄT DARMSTADT

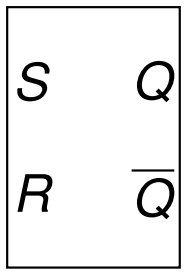
- Sequentielle Logik
  - Sequentielle Schaltungen
  - Speicherelemente
  - Synchrone sequentielle Logik
- Endliche Zustandsautomaten
  - Konzept, Notationen und Anwendungsbeispiele
  - Moore vs. Mealy
  - Zerlegen von Zustandsautomaten

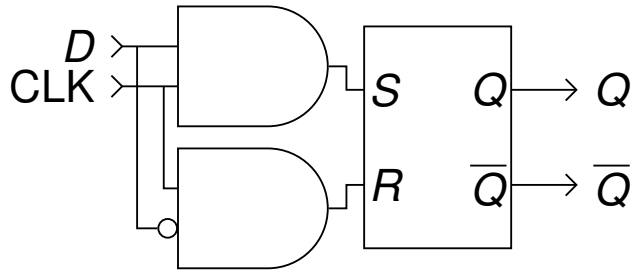
Harris 2013
Kap. 3.1 - 3.4

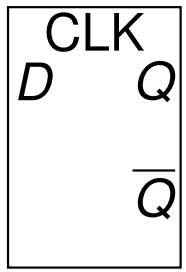
Wiederholung: Speicherelemente

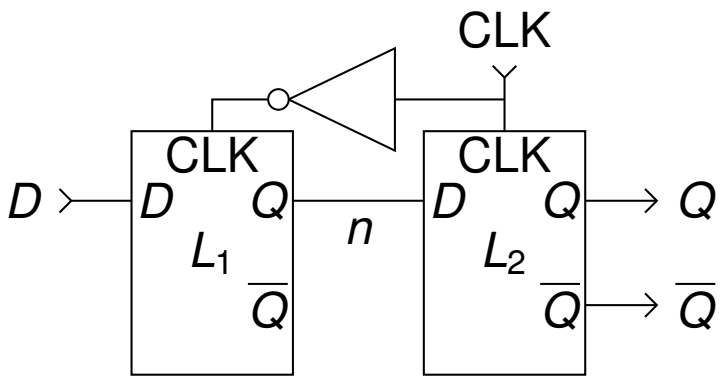
TECHNISCHE UNIVERSITÄT DARMSTADT

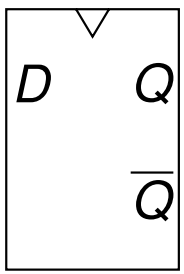






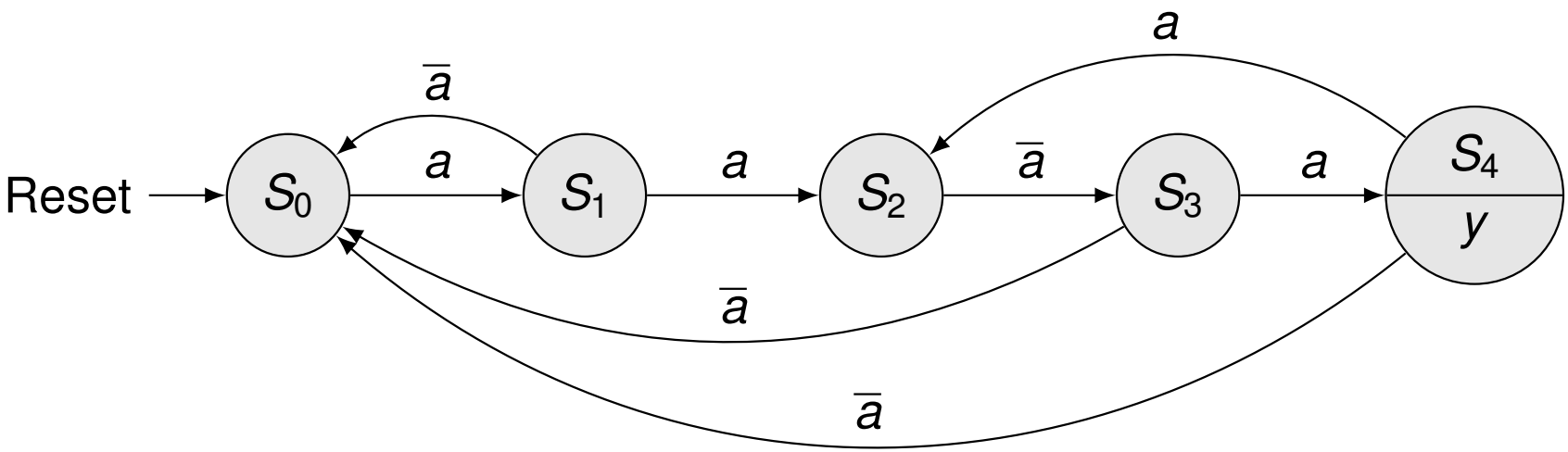


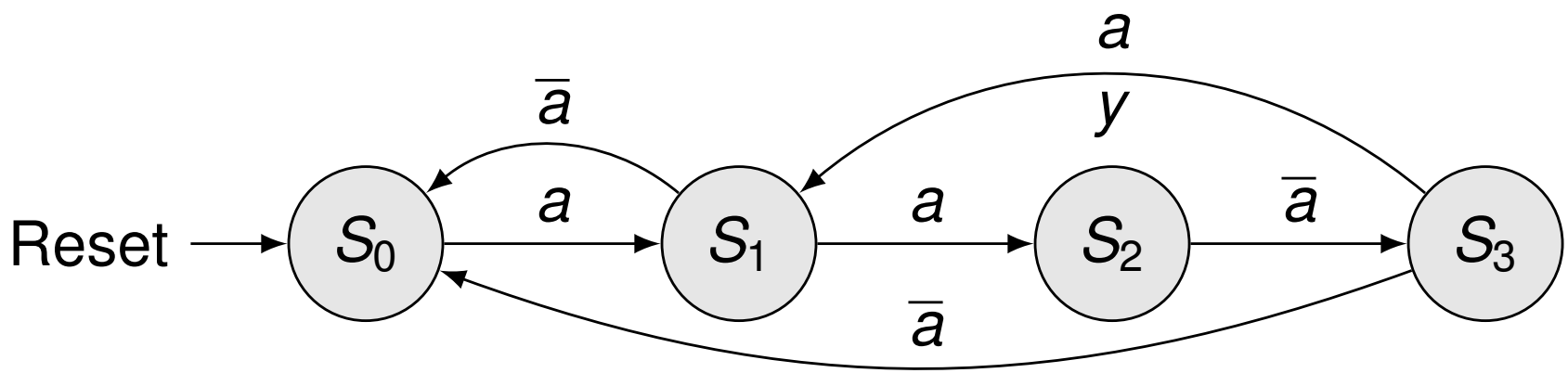




Wiederholung: Endliche Automaten
Notation, Überführung

TECHNISCHE UNIVERSITÄT DARMSTADT



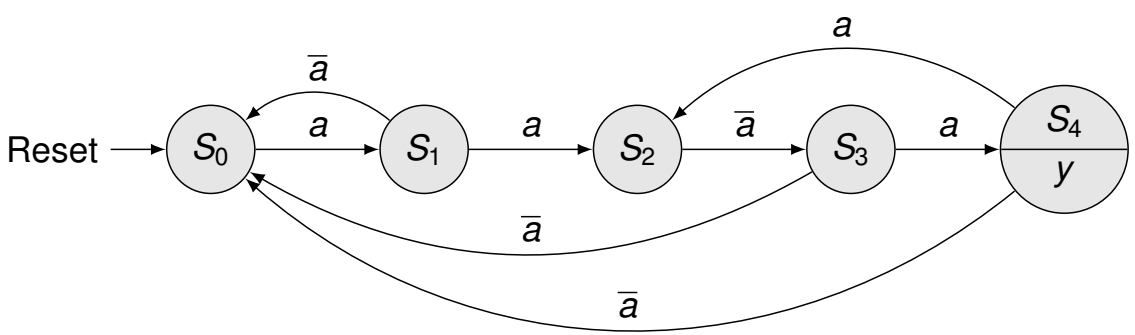


Wiederholung: Endliche Automaten
Don't Cares

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

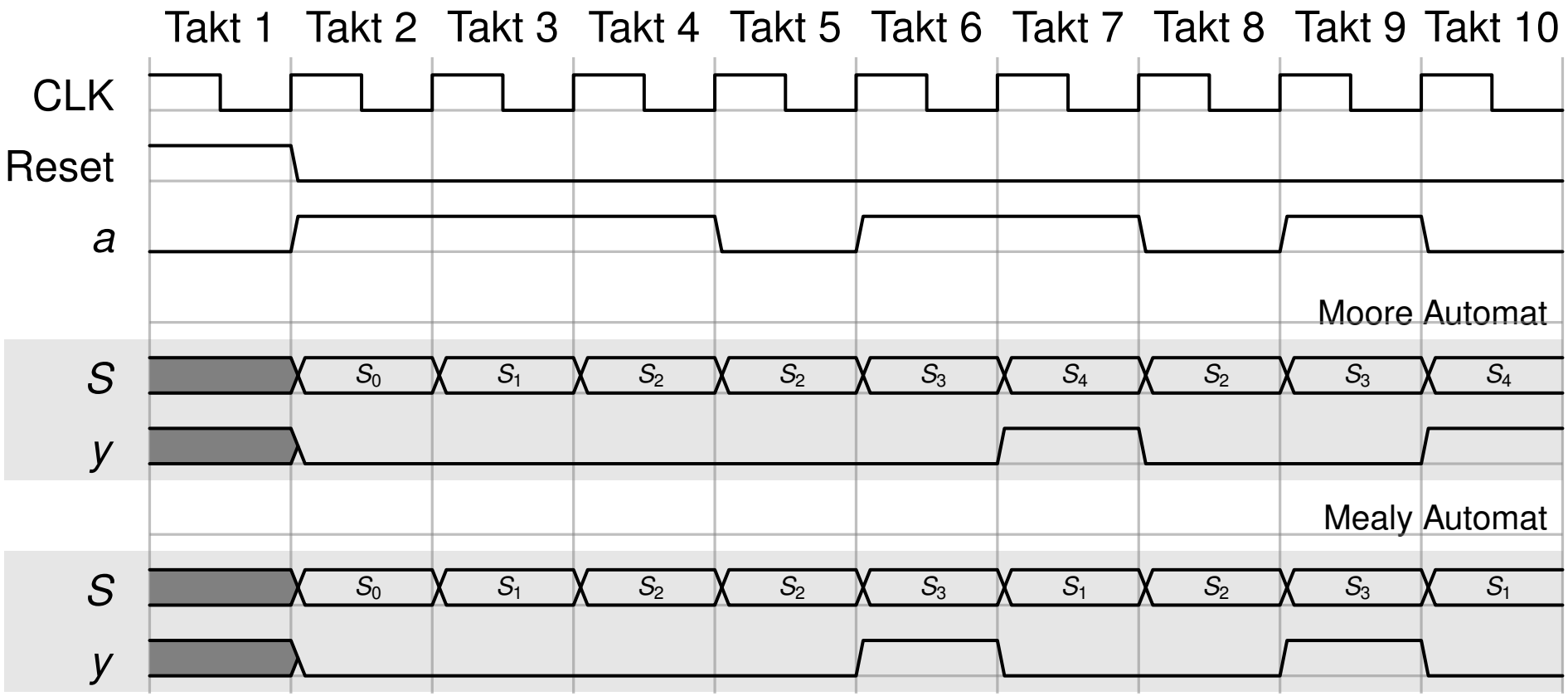


pattern/moore/state.esp

```
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 ---
```

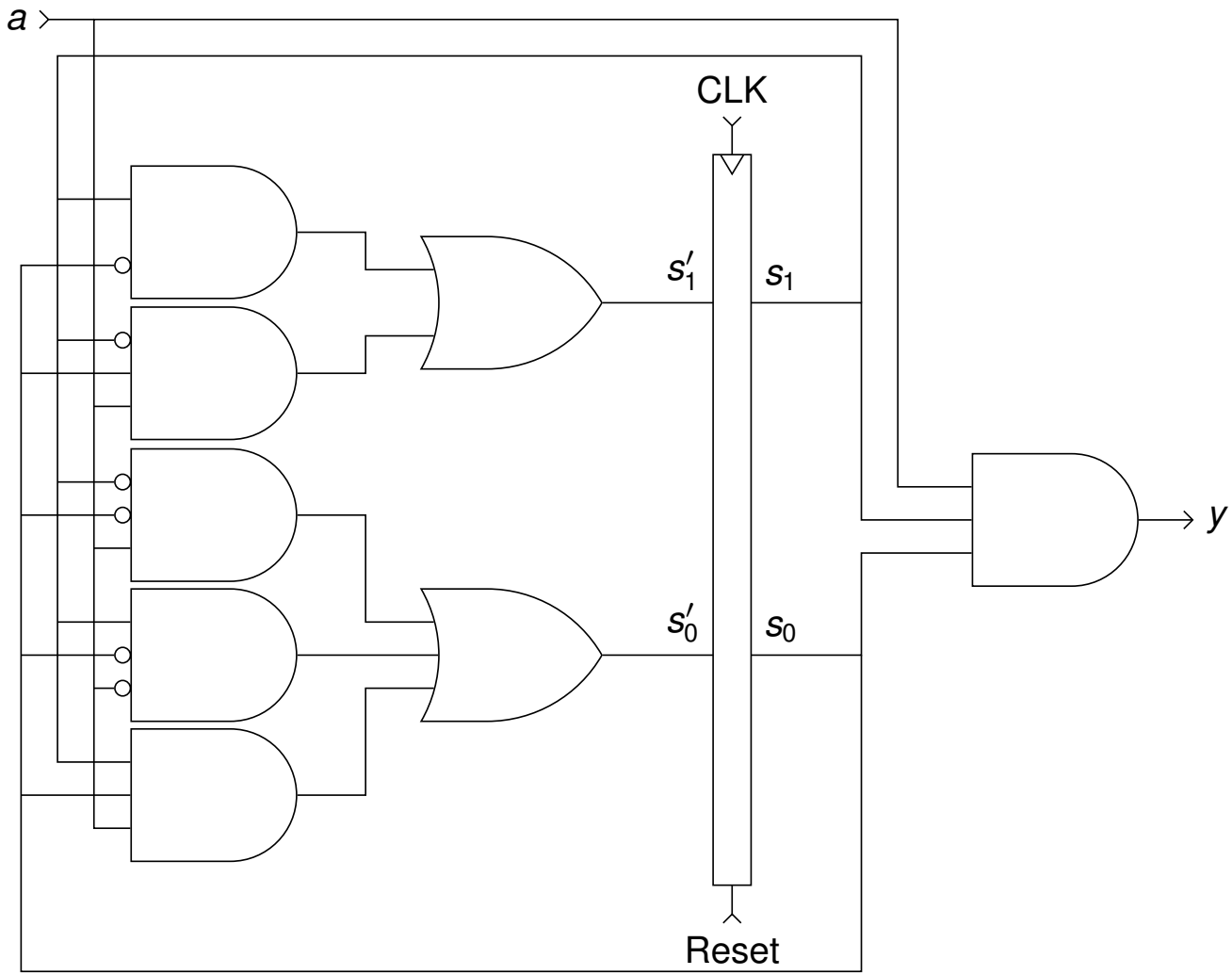
Wiederholung: Endliche Automaten
Mealy vs. Moore

TECHNISCHE UNIVERSITÄT DARMSTADT



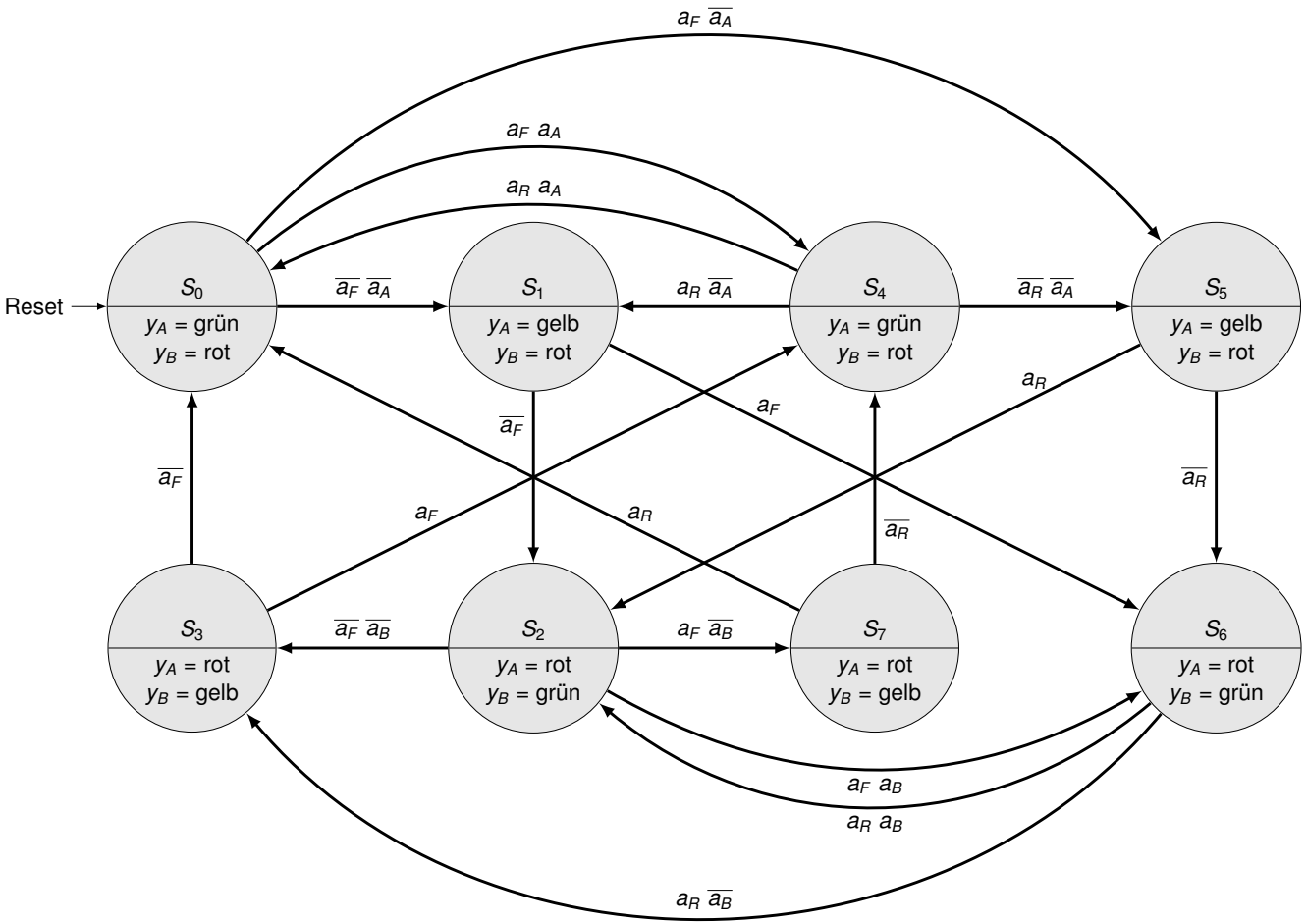
Wiederholung: Endliche Automaten
Mealy vs. Moore

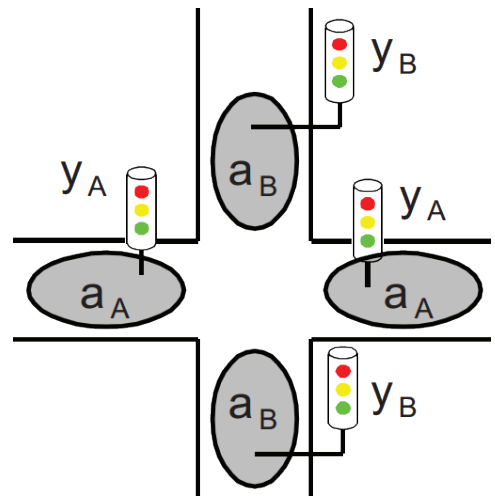
TECHNISCHE UNIVERSITÄT DARMSTADT



Wiederholung: Endliche Automaten
Zerlegung

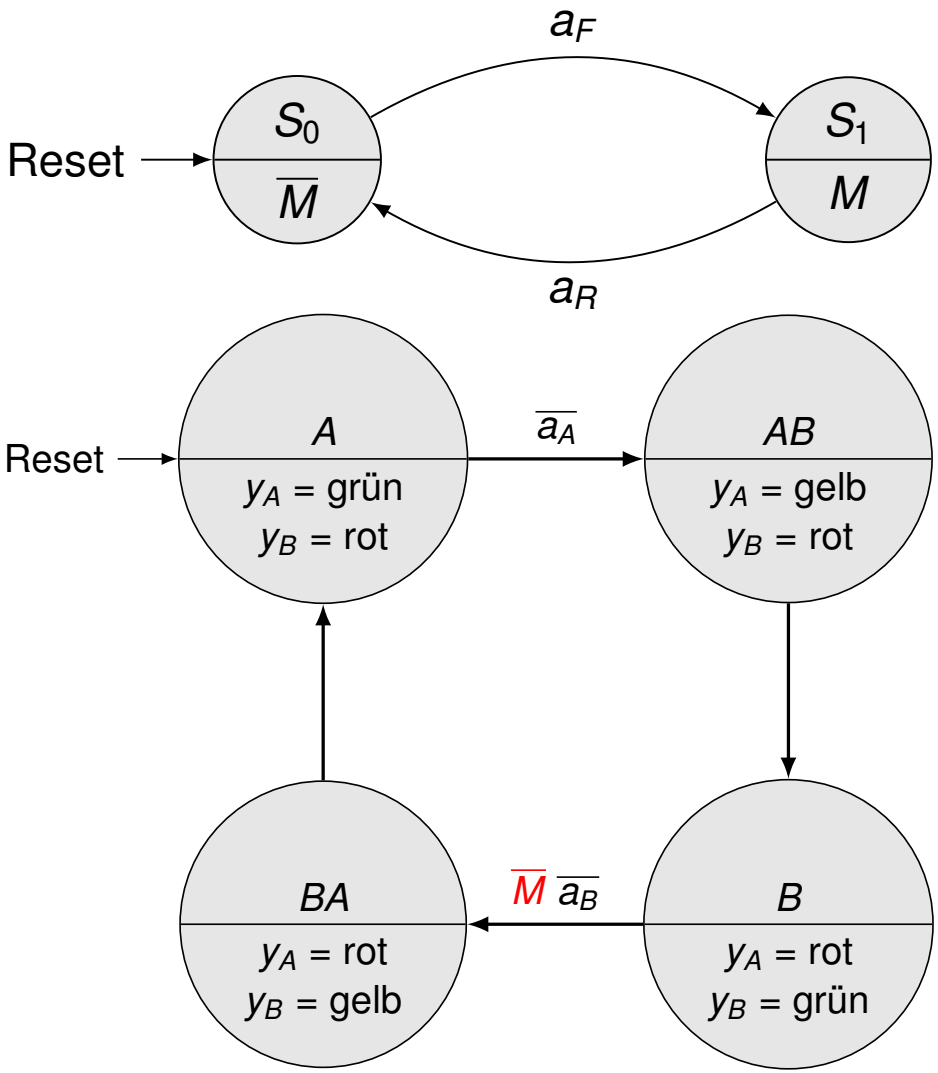
TECHNISCHE UNIVERSITÄT DARMSTADT

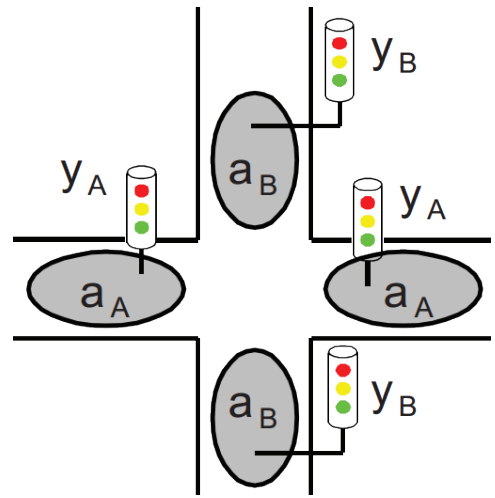




Wiederholung: Endliche Automaten
Zerlegung

TECHNISCHE UNIVERSITÄT DARMSTADT





Überblick der heutigen Vorlesung

TECHNISCHE UNIVERSITÄT DARMSTADT

- Zeitverhalten synchroner sequentieller Schaltungen
- Parallelität

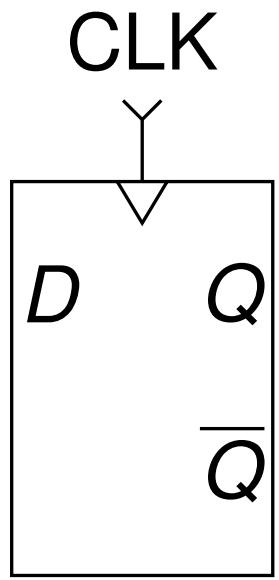
Harris 2013
Kap. 3.5-3.6
Seite 133 - 153

Zeitverhalten synchroner sequentieller Logik

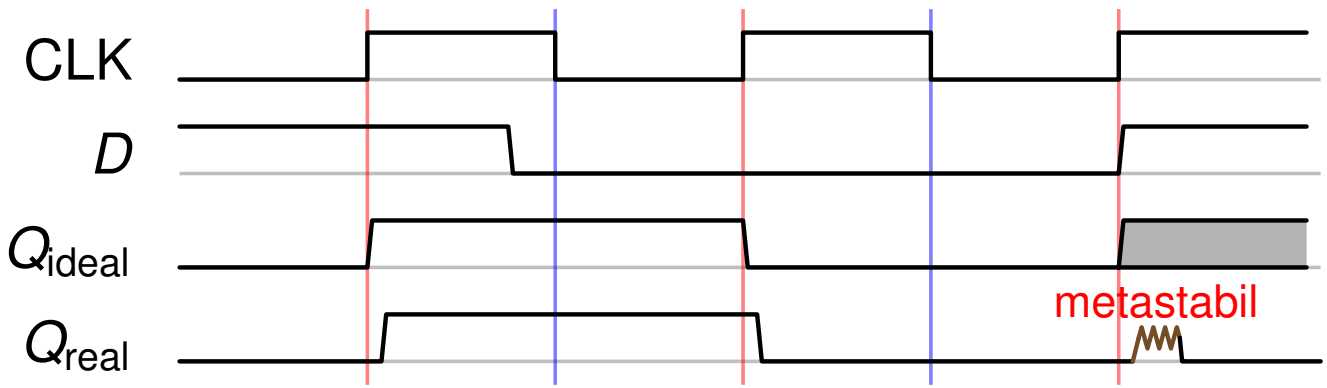
TECHNISCHE UNIVERSITÄT DARMSTADT

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?



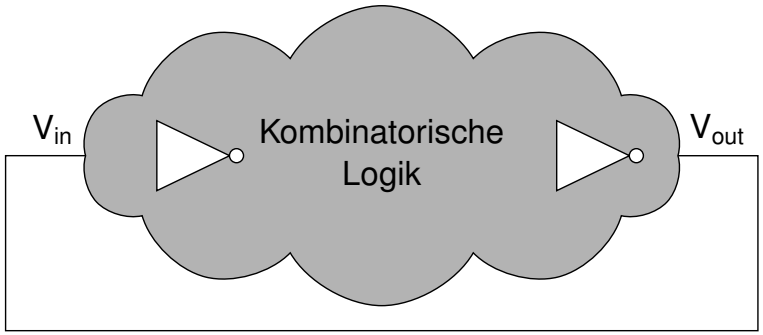
Analogie zur Fotografie: Bewegungsunschärfe (Bewegung während Belichtung)

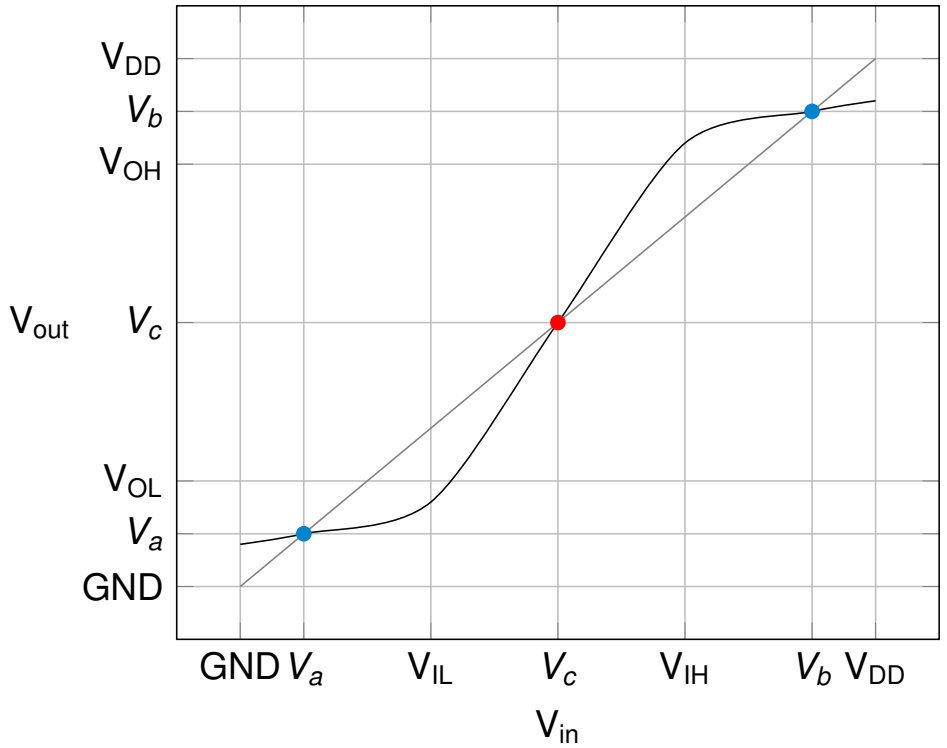
TECHNISCHE UNIVERSITÄT DARMSTADT

Quelle: `http://lightwatching.de`

Metastabilität durch Rückkopplung kombinatorischer Logik

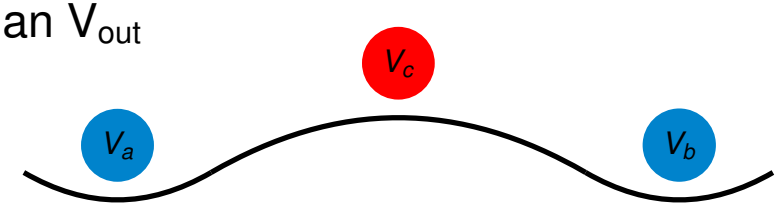
TECHNISCHE UNIVERSITÄT DARMSTADT





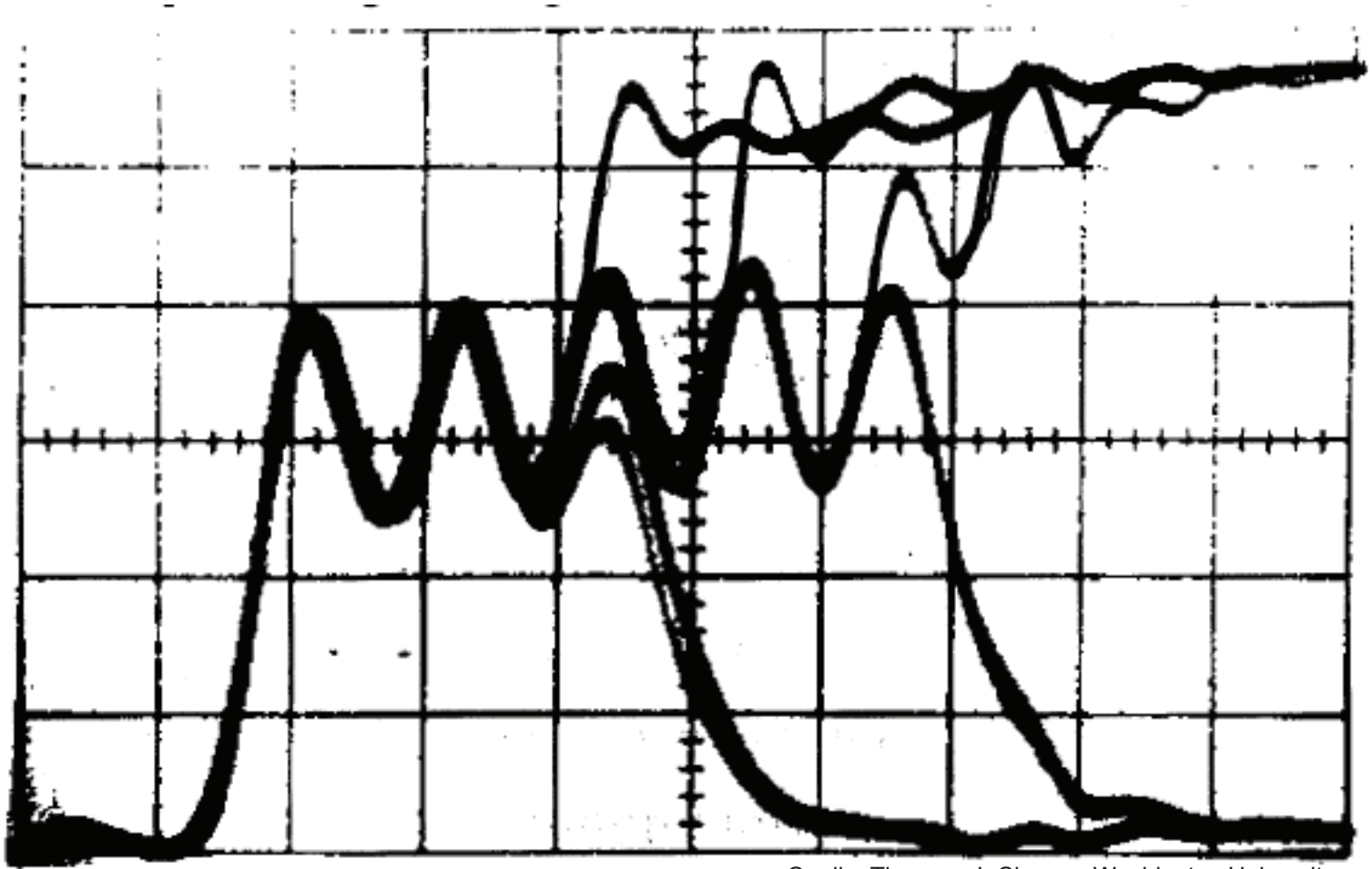
- Rückkopplung stabil für $V_{out} = V_{in}$
- Transferfunktion schneidet Hauptdiagonale in
  - V_a repräsentiert 0
  - V_b repräsentiert 1
  - V_c im „verbotenen“ Spannungsbereich
- V_c ist „metastabil“, da
  - kleine Änderung an V_{in} $\rightarrow$ große Änderung an V_{out}

$\Rightarrow$ 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

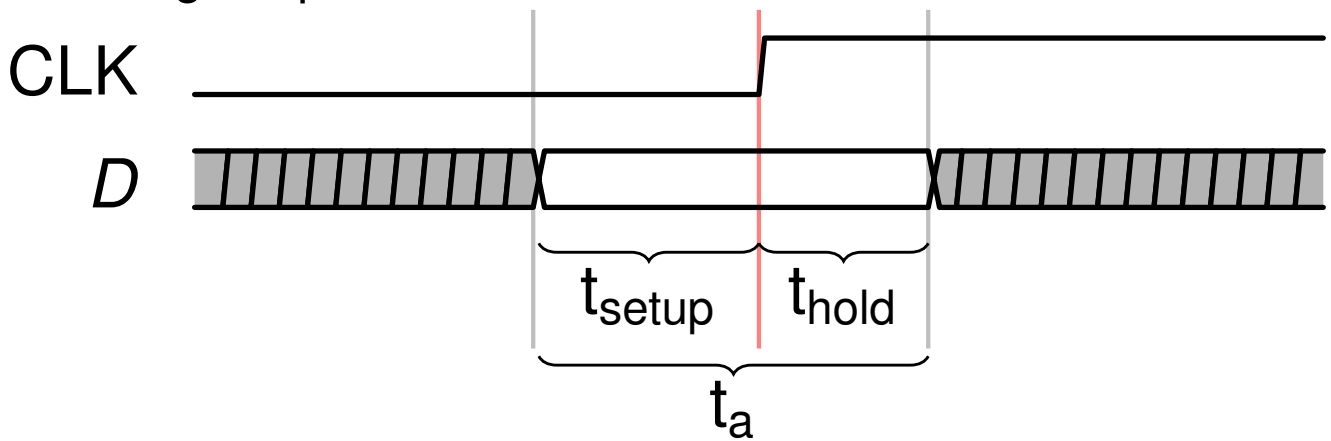
- Dateneingang *D* muss in Abtast-Zeitfenster um Taktflanke stabil sein, um Metastabilität zu vermeiden

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

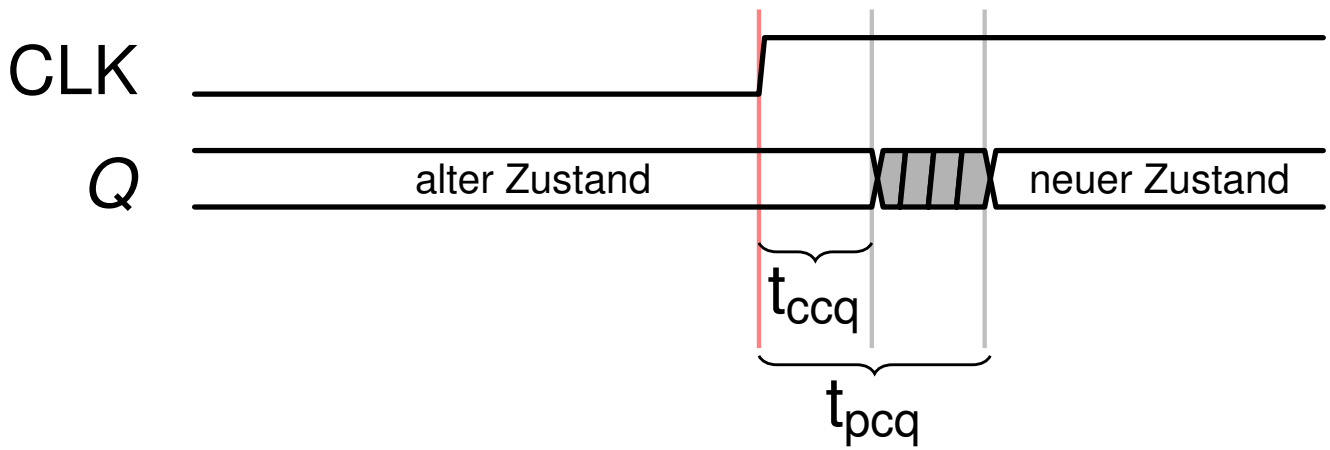
- Größenordnung: 10 ps



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

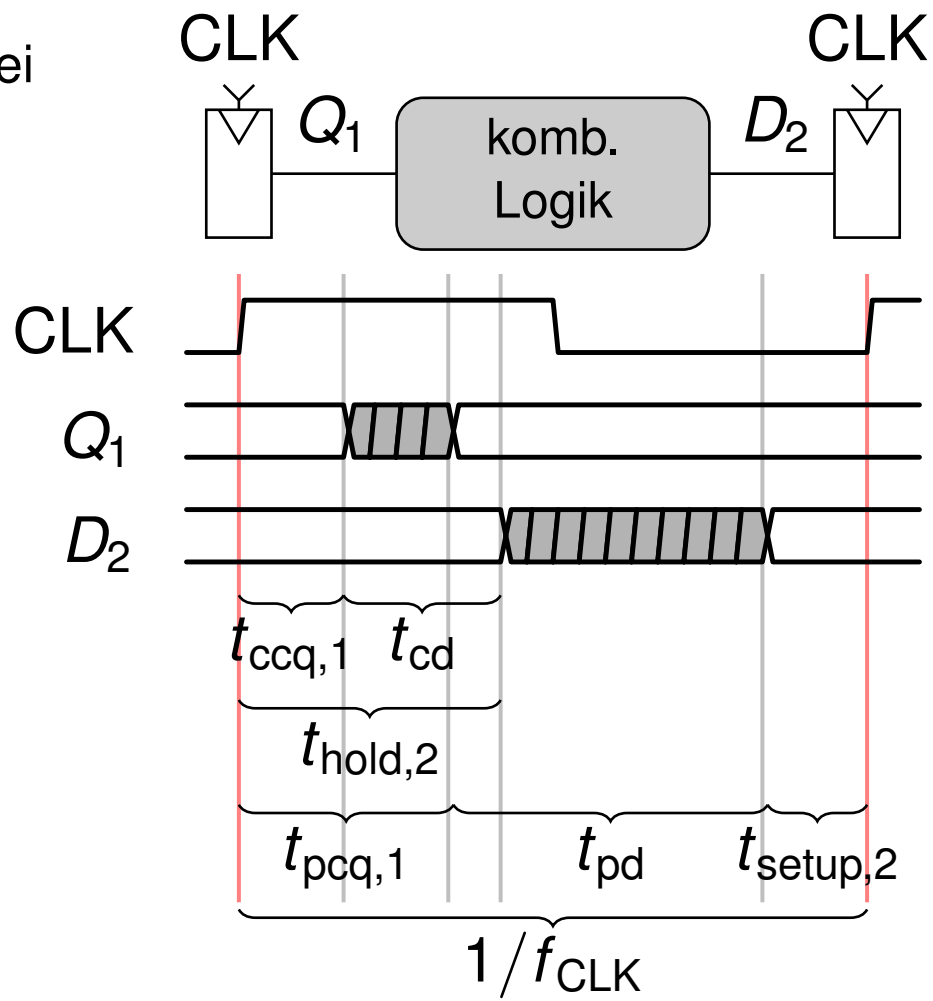
- kombinatorische Logik zwischen zwei Registern hat max/min Verzögerung
- abhängig von Verzögerungen der Gatter und des *ersten* Registers

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

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

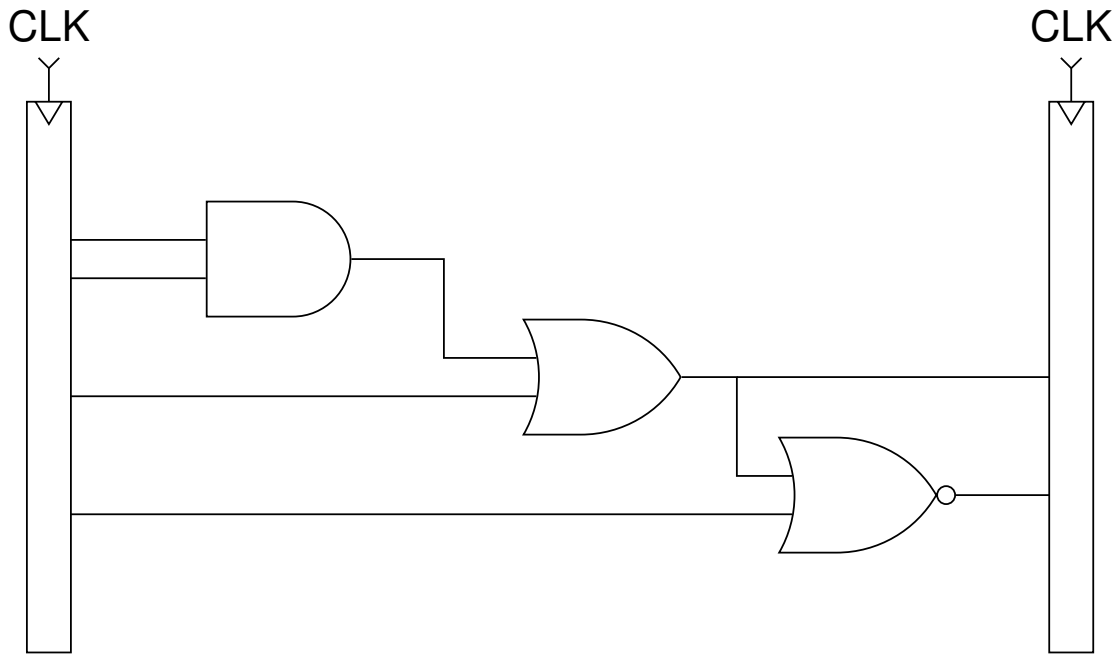
$\Rightarrow$ maximale Taktrate wird durch *kritischen Pfad* bestimmt

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



Beispiel:
Analyse der Timing-Bedingungen

TECHNISCHE UNIVERSITÄT DARMSTADT

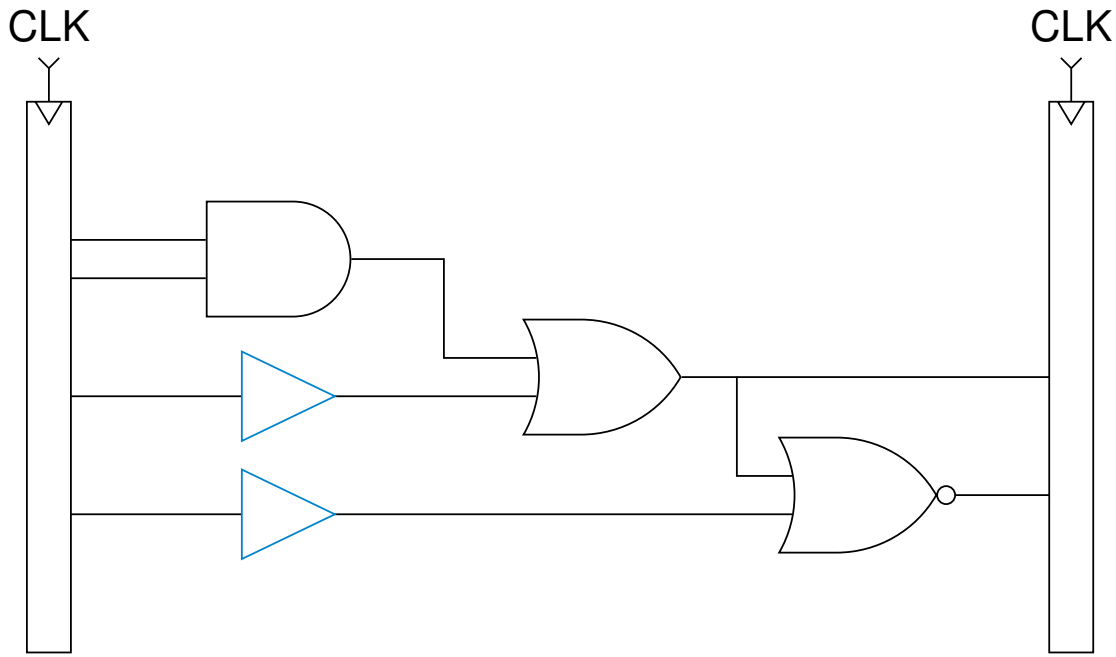


- Timing-Vorgaben:
  - $t_{ccq} = 30\,\text{ps}$
  - $t_{pcq} = 50\,\text{ps}$
  - $t_{setup} = 60\,\text{ps}$
  - $t_{hold} = 70\,\text{ps}$
  - $t_{cd,Gatter} = 25\,\text{ps}$
  - $t_{pd,Gatter} = 35\,\text{ps}$
- kombinatorischer Pfad:
  - $t_{cd} = 25\,\text{ps}$
  - $t_{pd} = 3 \cdot 35\,\text{ps} = 105\,\text{ps}$
- Timing-Bedingungen:
  - $f_{CLK} \leq \frac{1}{t_{pcq}+t_{pd}+t_{setup}} = \frac{1}{215\,\text{ps}} = 4{,}65\,\text{GHz}$
  - $t_{ccq} + t_{cd} = 55\,\text{ps} < t_{hold}$ ↯

Beispiel:
Beheben der verletzten Hold-Zeitanforderung

TECHNISCHE UNIVERSITÄT DARMSTADT

- Timing-Vorgaben:
  - $t_{ccq} = 30\,\text{ps}$
  - $t_{pcq} = 50\,\text{ps}$
  - $t_{setup} = 60\,\text{ps}$
  - $t_{hold} = 70\,\text{ps}$
  - $t_{cd,Gatter} = 25\,\text{ps}$
  - $t_{pd,Gatter} = 35\,\text{ps}$
- kombinatorischer Pfad:
  - $t_{cd} = 50\,\text{ps}$
  - $t_{pd} = 3{\cdot}35\,\text{ps} = 105\,\text{ps}$

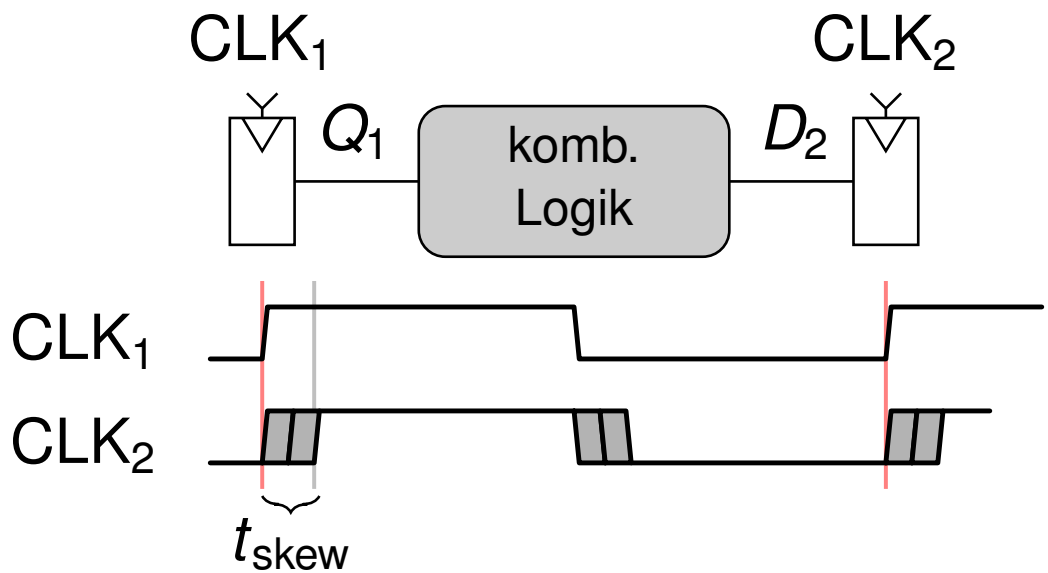


- Timing-Bedingungen:
  - $f_{CLK} \leq \frac{1}{t_{pcq}+t_{pd}+t_{setup}} = \frac{1}{215\,\text{ps}} = 4{,}65\,\text{GHz}$
  - $t_{ccq} + t_{cd} = 80\,\text{ps} > t_{hold}$ ✓

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



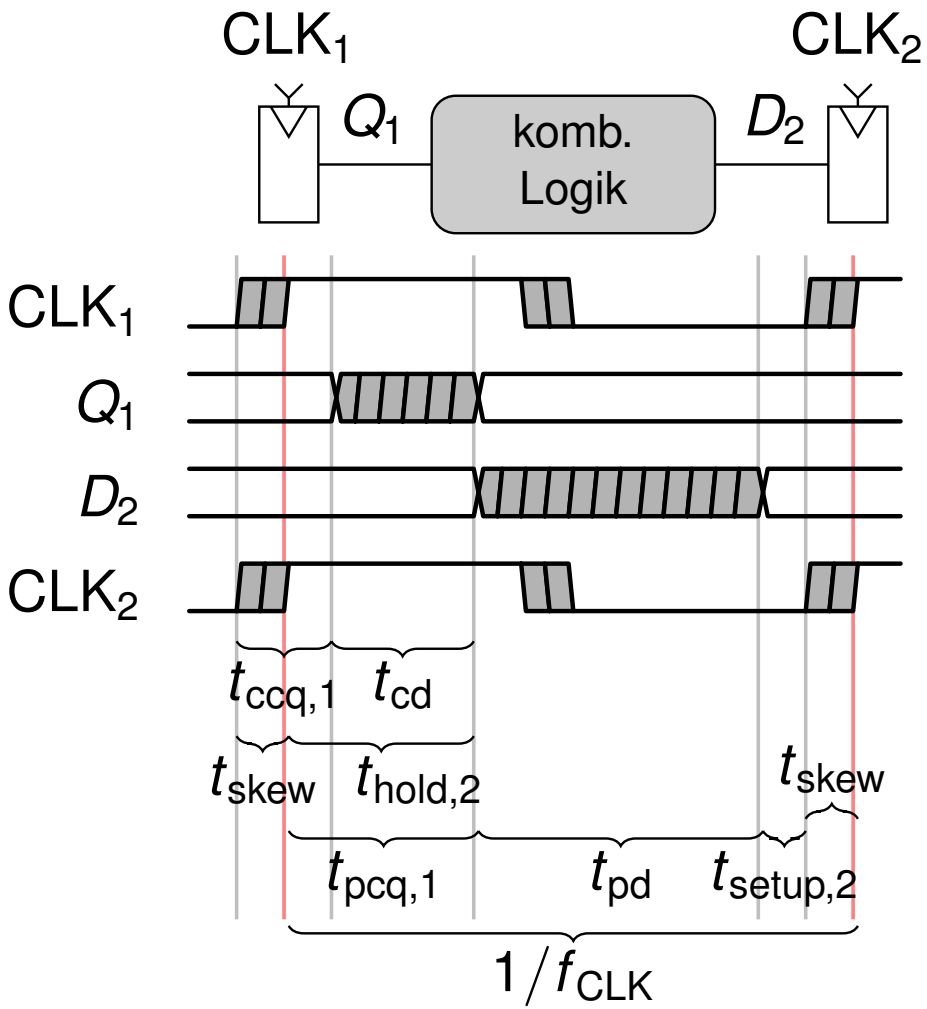
Timing-Bedingungen mit Taktverschiebung

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 CLK_2 sicher nach CLK_1 schaltet



Verletzung der dynamischen Entwurfsdisziplin

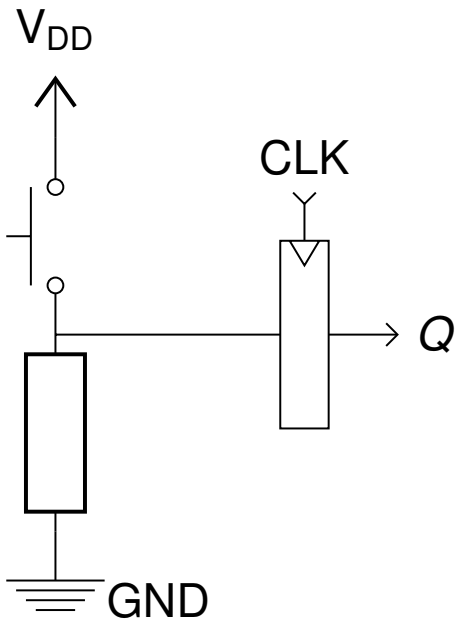
TECHNISCHE UNIVERSITÄT DARMSTADT

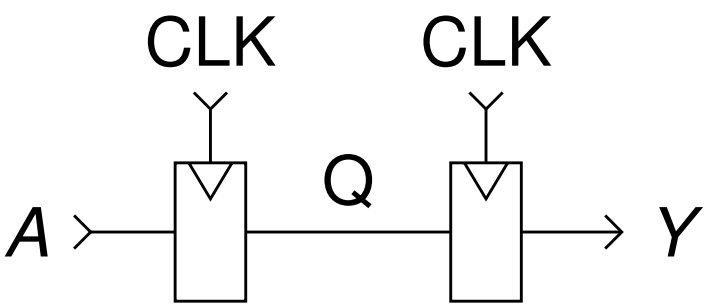
- asynchronen Eingänge:
  - Benutzereingaben
  - Kommunikationssignale von externen ICs

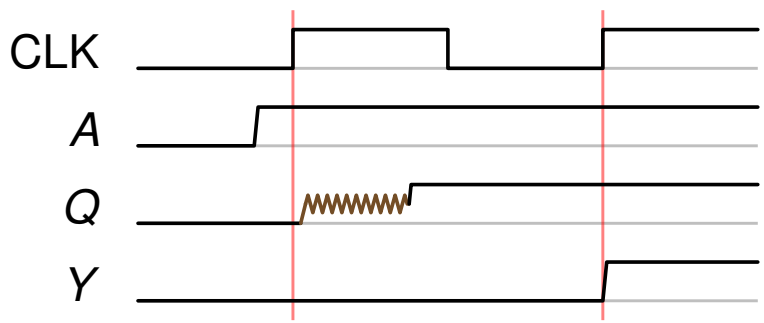
$\Rightarrow$ Timing-Bedingungen können nicht garantiert werden

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

  $\Rightarrow$ 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 ausführen
- Beispiel: Fließbandprinzip bei Autofertigung („Pipelining“)
  - nur eine Station für pro Arbeitsschritt
  - alle unterschiedlichen Arbeitsschritte für mehrere Autos parallel ausgeführt

  $\Rightarrow$ zeitliche Paralellität

Grundlegende Begriffe

TECHNISCHE UNIVERSITÄT DARMSTADT

Datensatz: Vektor aus Eingabewerten, zu denen ein Vektor aus Ausgabewerten berechnet wird

Latenz: Zeit von der Eingabe eines Datensatzes bis zur Ausgabe des zugehörigen Ergebnisses

Durchsatz: Anzahl von Datensätzen, die pro Zeiteinheit bearbeitet werden können

$\Rightarrow$ Parallelität erhöht Durchsatz

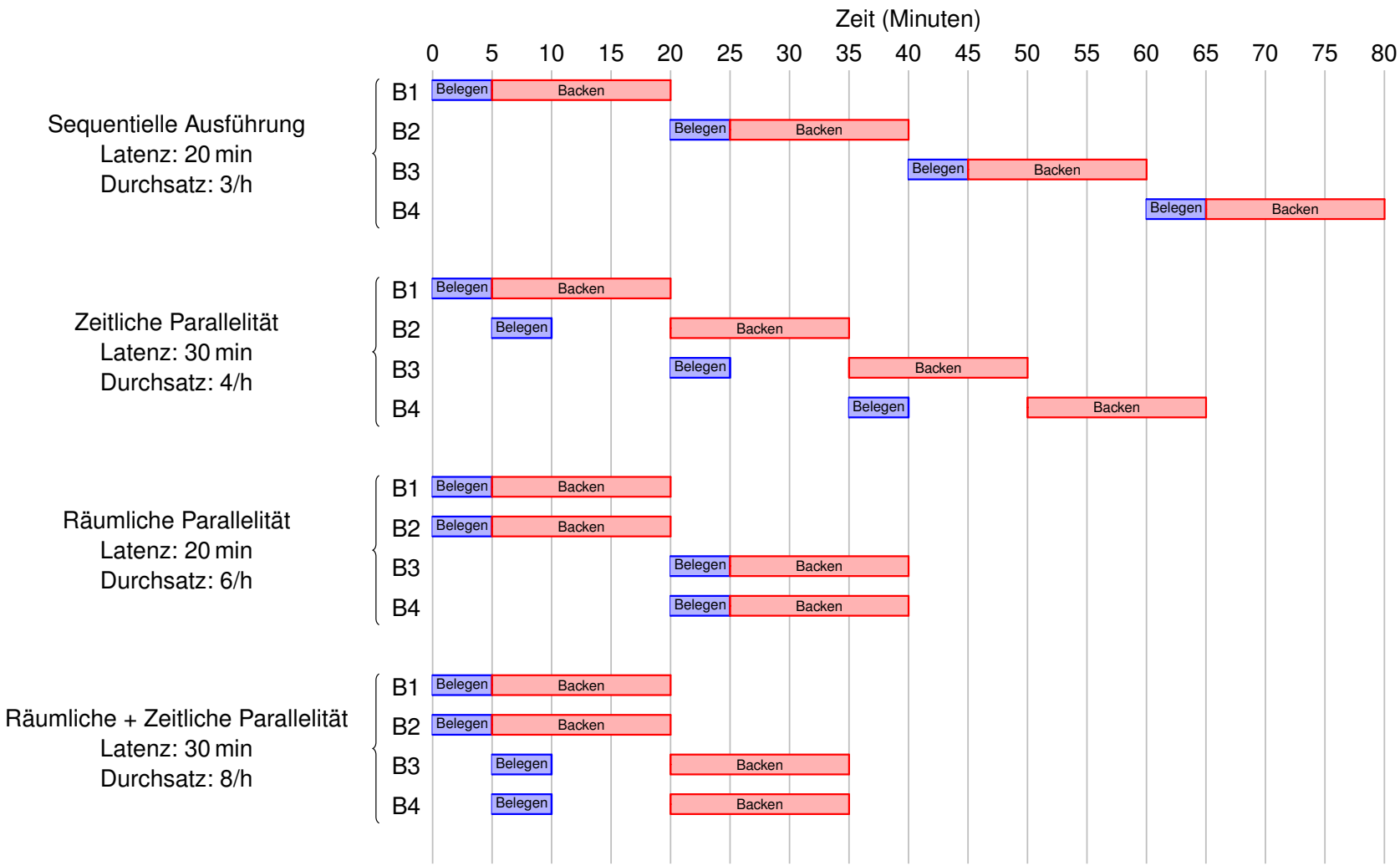
Beispiel: Plätzchen backen

TECHNISCHE UNIVERSITÄT DARMSTADT

- Annahmen:
  - genug Teig ist fertig
  - 5 Minuten zum Belegen eines Bleches
  - 15 Minuten Backzeit
- *sequentiel*: ein Blech nach dem anderen belegen und backen
- *zeitlich parallel:* nächstes Blech belegen, während erstes noch im Ofen ist
- *räumlich parallel:* zwei Bäcker, jeweils mit eigenem Ofen
- räumliche und zeitliche Parallelität kombiniert

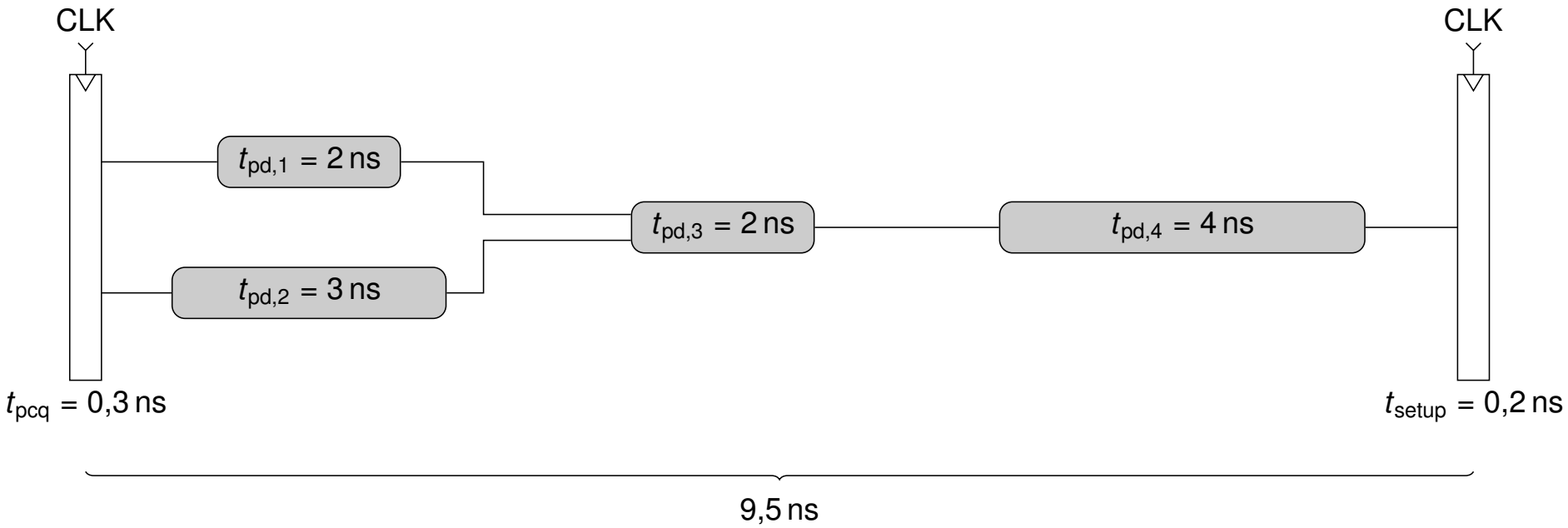
Beispiel: Plätzchen backen

TECHNISCHE UNIVERSITÄT DARMSTADT



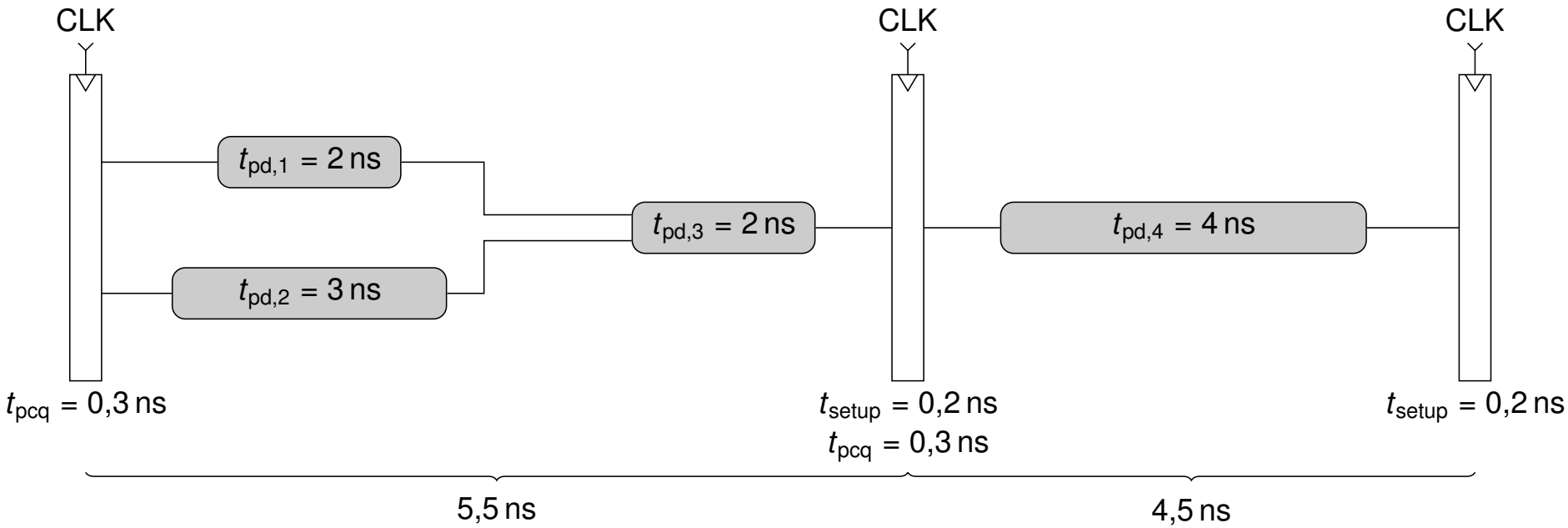
Beispiel: Pipelining in Schaltungen
Ohne Pipeline-Register

TECHNISCHE UNIVERSITÄT DARMSTADT



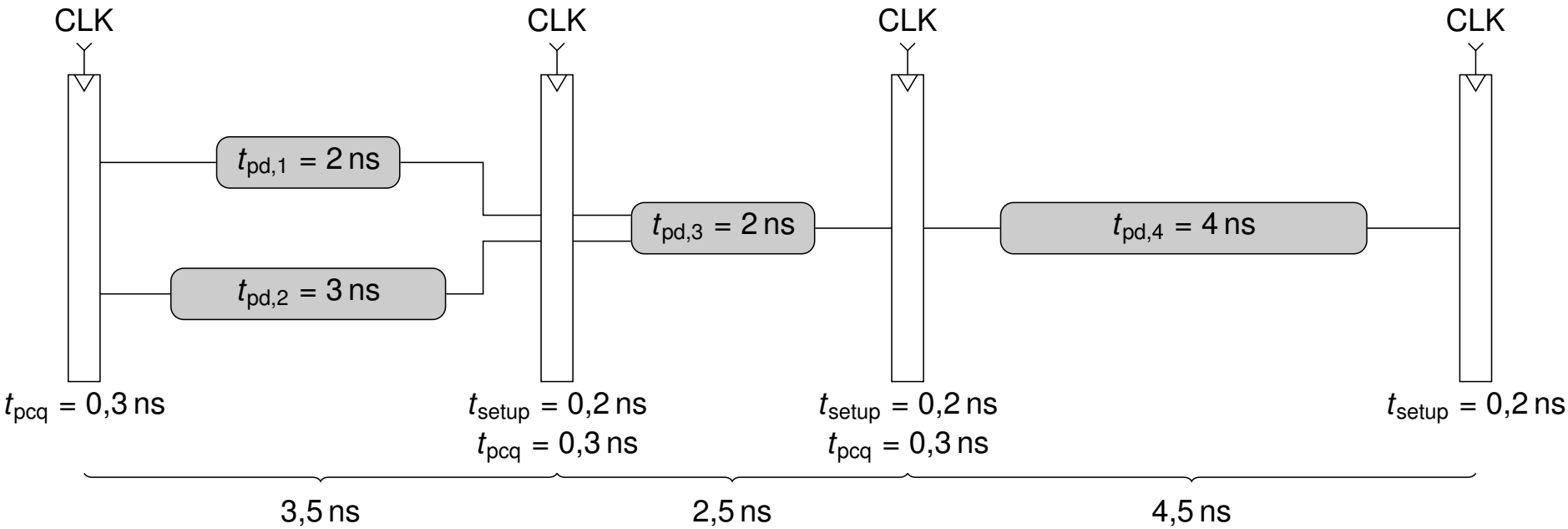
- $f_{CLK} \leq \frac{1}{9{,}5\,\text{ns}} = 105\,\text{MHz}$
- Latenz: 1 Takt $\equiv$ 9,5 ns

Beispiel: Pipelining in Schaltungen
Zwei Pipeline-Stufen



- $f_{CLK} \leq \max(\frac{1}{5{,}5\,\text{ns}}, \frac{1}{4{,}5\,\text{ns}}) = 182\,\text{MHz}$
- Latenz: 2 Takte $\equiv$ 11 ns

Beispiel: Pipelining in Schaltungen
Drei Pipeline-Stufen



- $f_{CLK} \leq \max(\frac{1}{3{,}5\,\text{ns}}, \frac{1}{2{,}5\,\text{ns}}, \frac{1}{4{,}5\,\text{ns}}) = 222\,\text{MHz}$
- Latenz: 3 Takte $\equiv$ 13,5 ns

Bewertung Pipelining

TECHNISCHE UNIVERSITÄT DARMSTADT

- Pipelinestufen sollten möglichst gleich lang sein („ausbalanciert“)
  - längste Stufe bestimmt Taktrate
  - Latenz = Pipelinestufen / Taktrate
- mehr Pipelinestufen
  - höherer Durchsatz (mehr Ergebnisse pro Zeiteinheit)
  - aber auch höhere Latenz (länger auf Ergebnis warten)

  $\Rightarrow$ lohnt sich nur, wenn viele Datensätze bearbeitet werden müssen
- Probleme bei Abhängigkeiten zwischen Teilaufgaben
  - bspw. erst Backergebnis prüfen, bevor nächstes Blech belegt 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
  - Hardwarebeschreibung mit SystemVerilog