

Umgang mit dem Kursmaterial

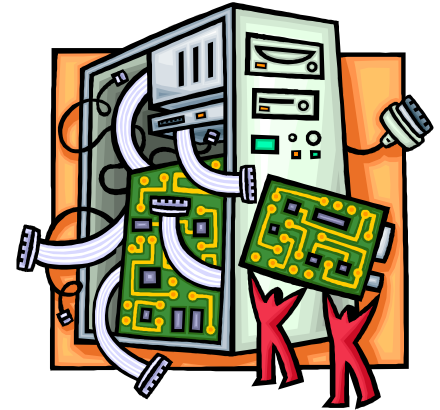
- Das Kursmaterial ist durch das Copyright des Dozenten und/oder anderer Autoren geschützt.
- Das Material darf nur von Studenten der angegebenen Fachhochschule und nur zu Ausbildungszwecken im Rahmen des angegebenen Kurses verwendet werden.
- Die Veröffentlichung oder Verbreitung des Materials ist strafbar und ausdrücklich untersagt. Dazu zählen das Veröffentlichen auf Webseiten, Onlinespeichern wie Dropbox, Verbreiten per Email, die Verwendung in Vorträgen oder Publikationen etc.

*Es gibt 10 Gruppen
von Menschen:*

*Solche, die
die Binärcodierung verstehen
und die anderen!*

Vorlesung_07

Speicher: Latch und Flipflop

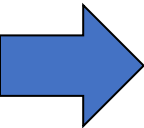


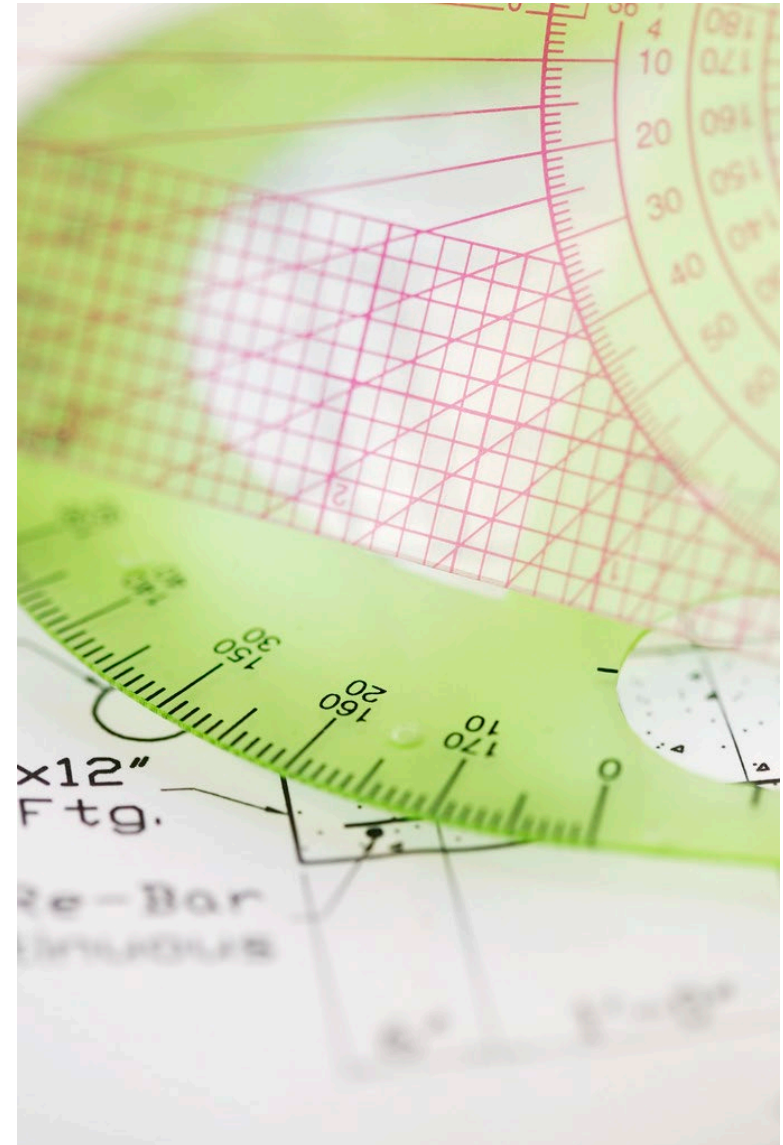
Technische Grundlagen der Informatik

Prof. Dr. Claus Fühner
Fakultät Informatik

Ostfalia Hochschule für angewandte Wissenschaften
Braunschweig/Wolfenbüttel

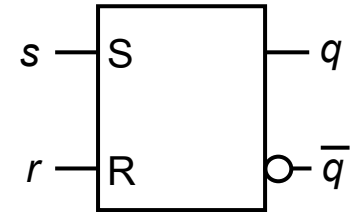
Agenda

- 
- Einfache Speicherelemente
 - Taktgesteuerte Elemente
 - Register



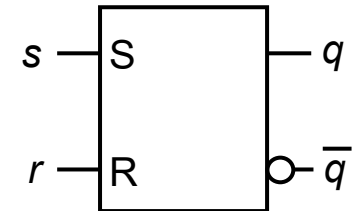
Definition

- **Einfachstes Speicherelement**
 - Einnahme der Zustände “0” oder “1”
 - 1-Bit Speicher
- **Ein- und Ausgänge**
 - s: Speicherinhalt auf 1 setzen (Set)
 - r: Speicherinhalt auf 0 zurücksetzen (Reset)
 - s=r=0: Speicherinhalt halten
 - q = aktueller Speicherinhalt
 - \bar{q} = negierter Speicherinhalt
- **Asynchrones Verhalten**: Set und Reset können zu beliebigen Zeitpunkten erfolgen
- **Spezielle Eigenschaft**: Verhalten des RS-Latches ist *undefiniert*, wenn Set und Reset gleichzeitig erfolgt.



Verhalten

- Ein Speicherelement hat einen **Zustand**.
- Das Verhalten eines Speicherelements (d.h. der Folgezustand) hängt von den Eingängen und dem **aktuellen Zustand** ab.
 - q^t bezeichnet den aktuellen Zustand
 - q^{t+1} den Folgezustand



s	r	q_t	q_{t+1}	Eigenschaft
0	0	0	0	Speichern
0	0	1	1	Speichern
0	1	0	0	Zurücksetzen
0	1	1	0	Zurücksetzen
1	0	0	1	Setzen
1	0	1	1	Setzen
1	1	0	?	Nicht erlaubt
1	1	1	?	Nicht erlaubt

Vereinfachte Wahrheitstabelle

s	r	q_{t+1}	Eigenschaft
0	0	q_t	Speichern
0	1	0	Zurücksetzen
1	0	1	Setzen
1	1	?	Nicht erlaubt

Asynchrones RS-Latch (Reset/Set)

Beispielhaftes Zeitdiagramm

s

r

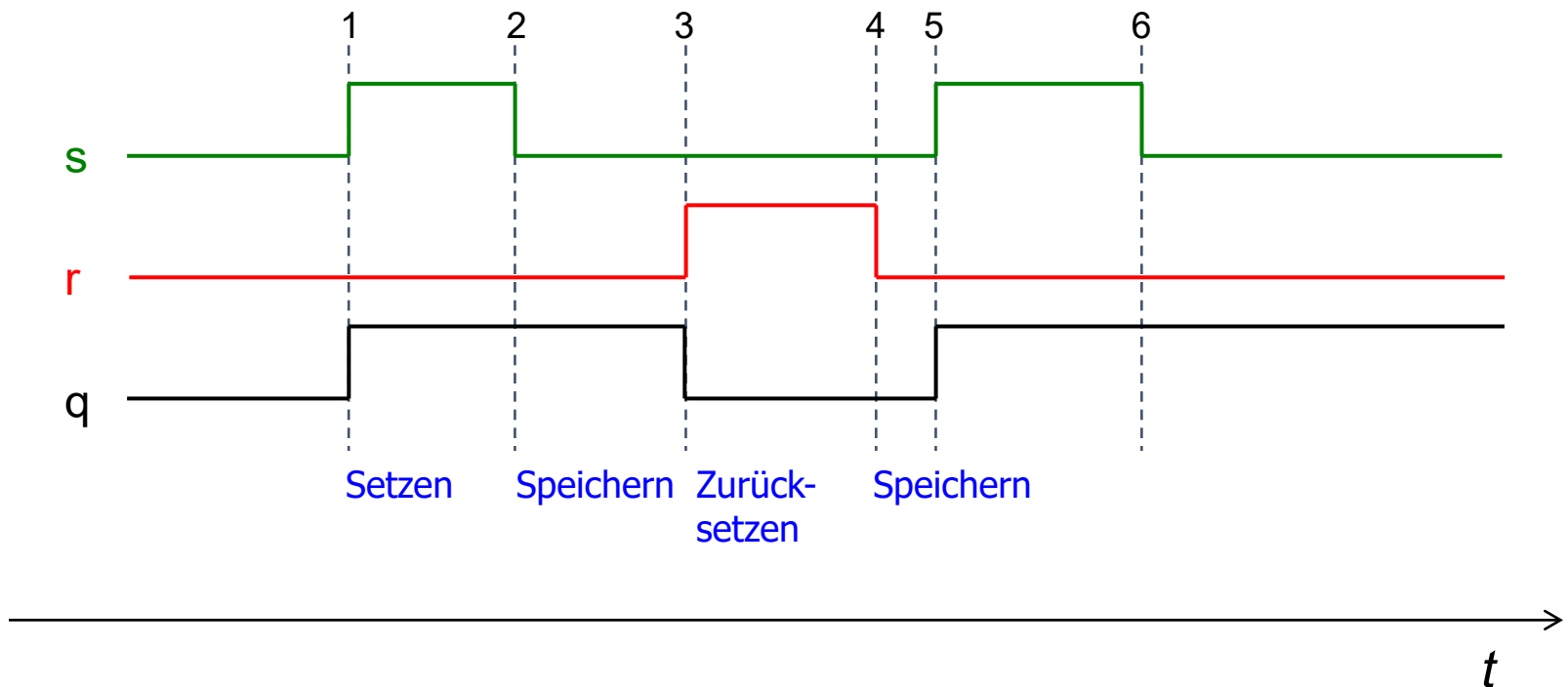
q

t

Asynchrones RS-Latch (Reset/Set)

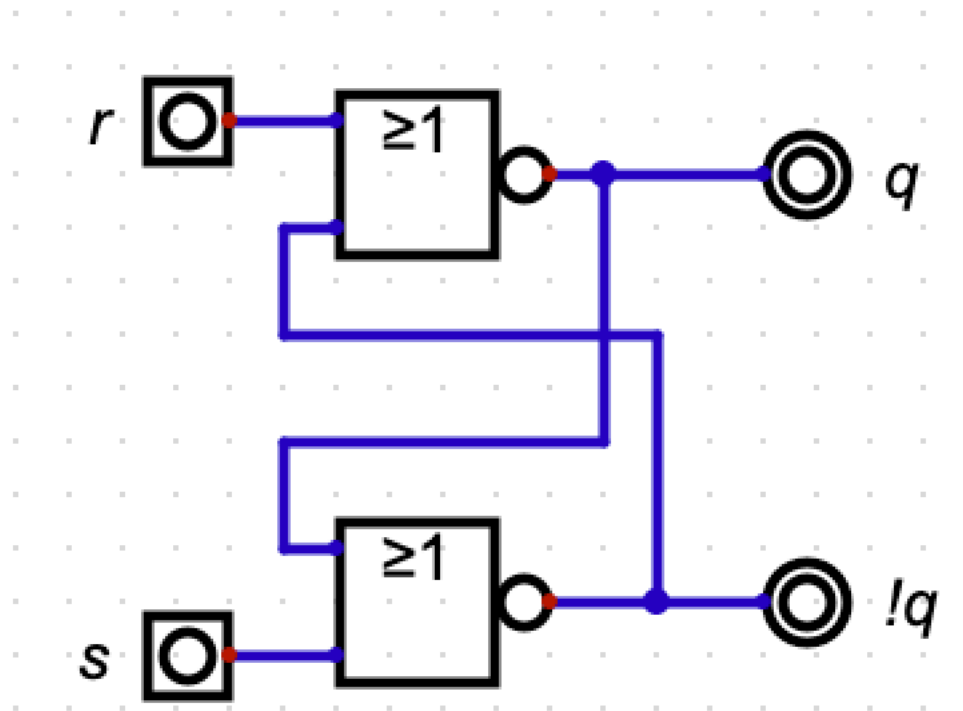


Beispielhaftes Zeitdiagramm



Realisierung

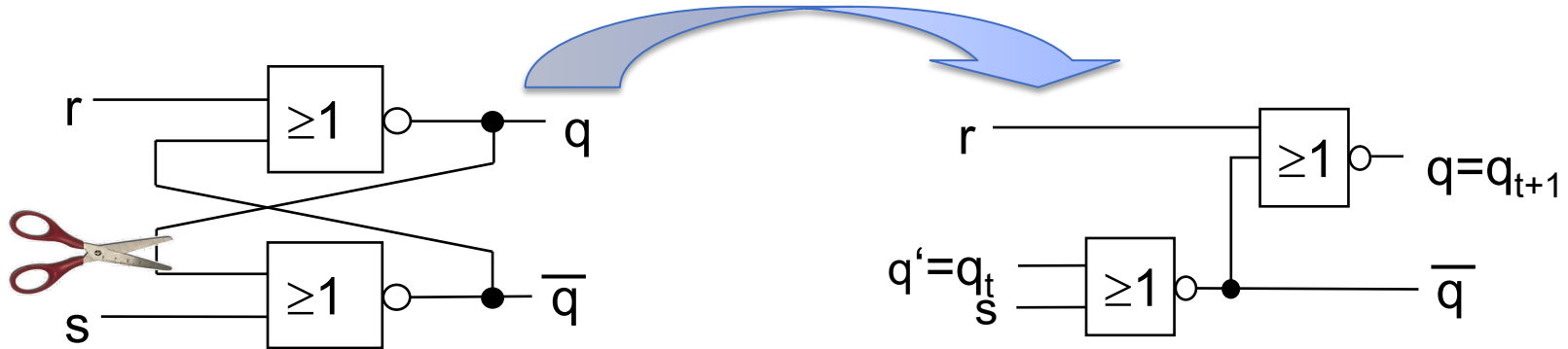
Aufbauvariante mit NOR-Gattern:



Wie verhält sich die Schaltung für $r=s=0$ mit (a) $q=1$ und (b) $q=0$

Asynchrones RS-Latch (Reset/Set)

Analyse der Rückkopplung



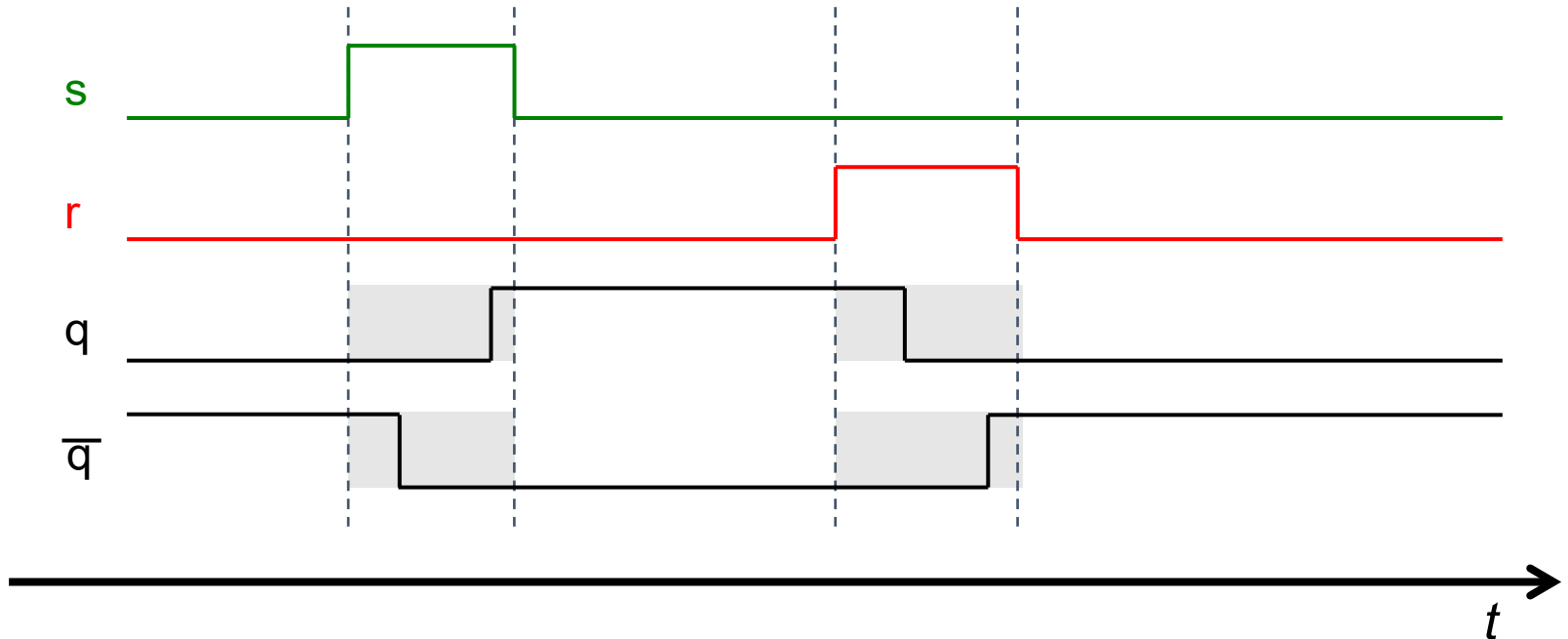
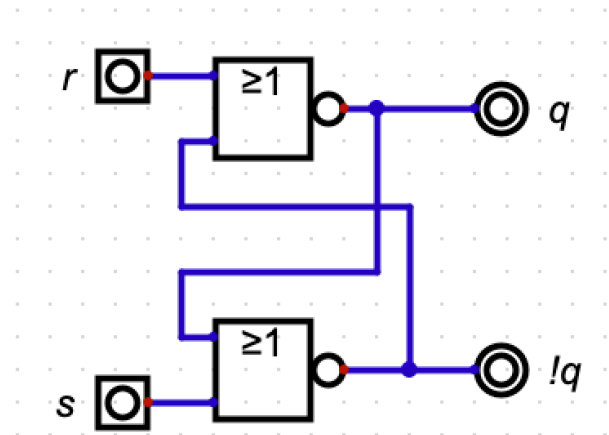
	r	s	q'		q	\bar{q}		Zustand
0	0	0	0					
1	0	0	1					
2	0	1	0					
3	0	1	1					
4	1	0	0					
5	1	0	1					
6	1	1	0					
7	1	1	1					

Bildquelle: http://commons.wikimedia.org/wiki/File:Scissors_left_and_right_handed.jpg

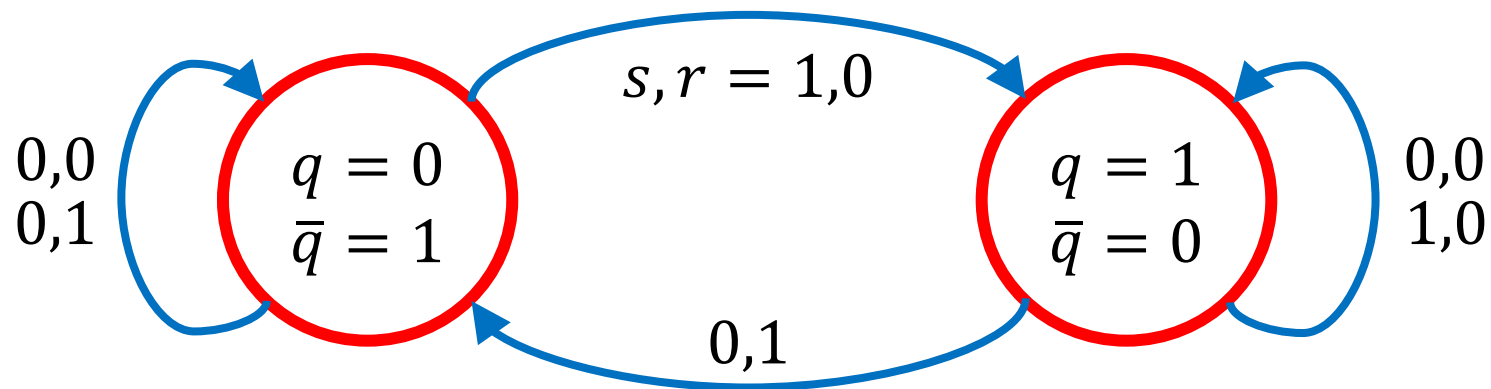


Zeitverhalten

- Reale Gatter schalten im Gegensatz zu idealen Gattern nicht beliebig schnell, sondern leicht verzögert



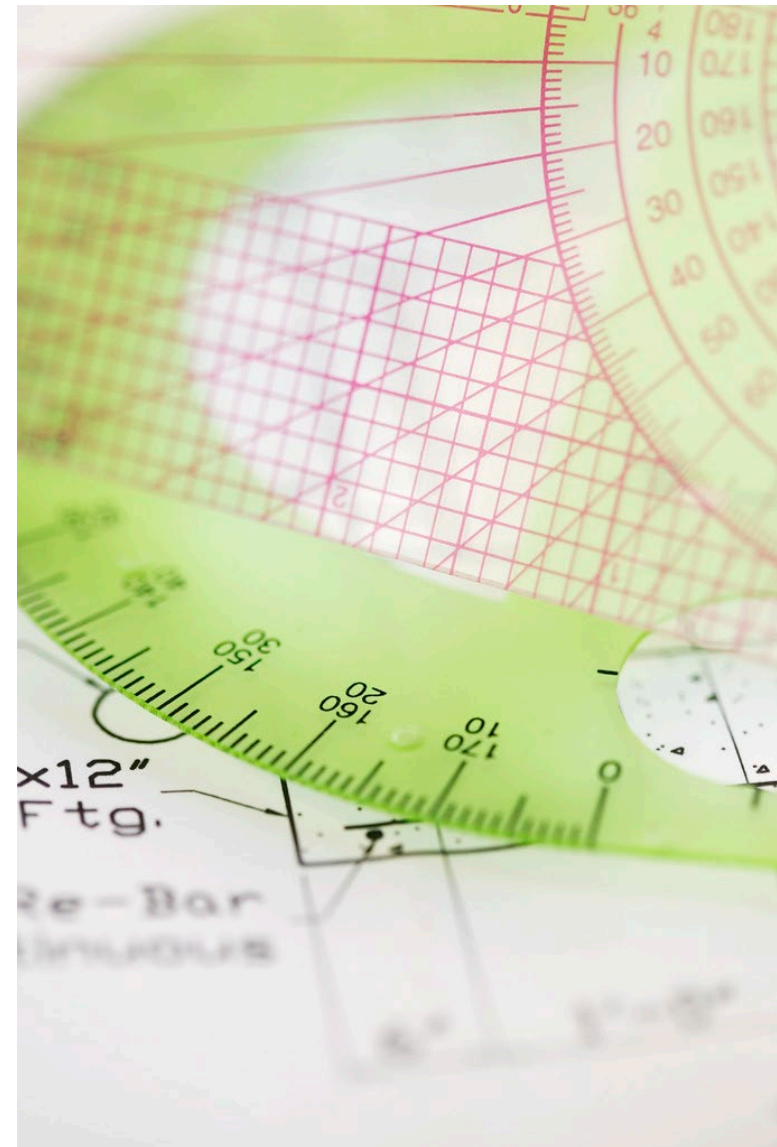
Zustandsdiagramm RS-Latch



- Zustände sind mit **Kreisen** markiert
- Für jeden Ausgangszustand sind mögliche **Zustandstransitionen** mit Pfeilen (und entsprechenden Bedingungen) markiert
- Verhalten für $(s, r) = (1, 1)$ war nicht definiert
- Die theoretisch darstellbaren Zustände $(q, \bar{q}) = (0, 0)$ und $(1, 1)$ werden hier nicht betrachtet – diese würden evtl. für eine Analyse der Verhaltens bei $(s, r) = (1, 1)$ relevant.

Agenda

- Einfache Speicherelemente
- ➔ Taktgesteuerte Elemente
- Register



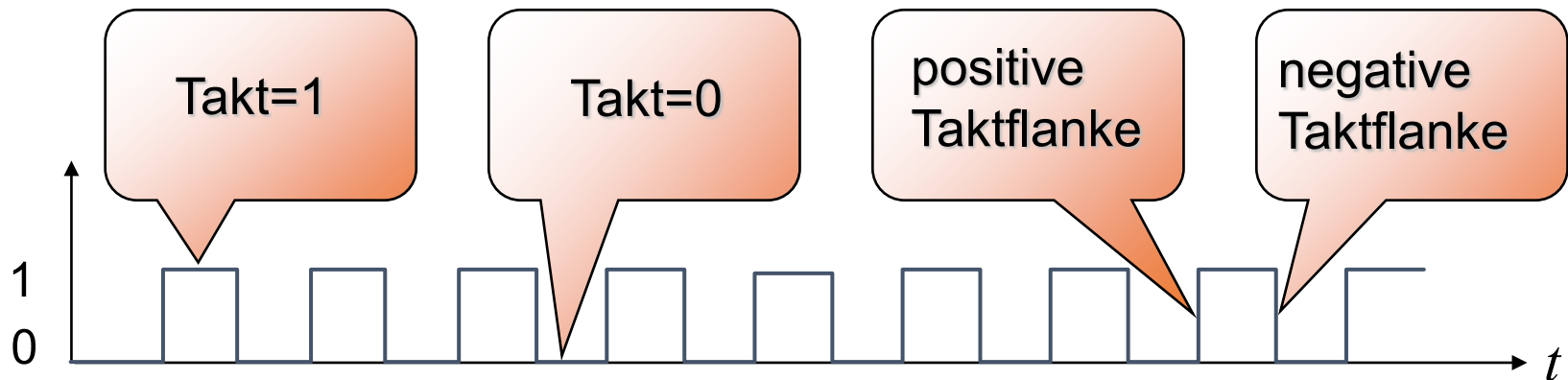
- Der Takt ist ein **periodisches Rechtecksignal**, das in einen **zusätzlichen Eingang** (z. B. des Speicherelements) eingespeist wird
- Eine Periode (1-Phase mit anschließender 0-Phase) heißt auch Zyklus
- **Periodenlänge**: Dauer einer Periode, Einheit: **s (Sekunde)**
- **Taktfrequenz**: Anzahl der Zyklen pro Sekunde, Einheit: **Hz („Hertz“)**
- **Periodenlänge ist Kehrwert der Taktfrequenz**



- **Asynchrone Speicherelemente** (z.B. RS-Latch) ändern ihren Zustand zu *beliebigen* Zeitpunkten
- **Synchrone Speicherelemente** ändern ihren Zustand nur zu *definierten* Zeitpunkten
- Zeitpunkte werden durch einen **Takt** (zentral oder heute auch häufiger dezentral) definiert
- Digitale Schaltungen müssen sich *einschwingen* bis ein gültiges Ergebnis vorliegt - Taktsteuerung definiert *Übernahmezeitpunkte*
- Schaltgeschwindigkeit der Bausteine (z. B. 64-Bit Carry-Ripple Addierer) bestimmt maximale Taktfrequenz



- Pegelsteuerung
 - Zustandsänderung während eine 1 auf der Taktleitung anliegt
- Flankensteuerung
 - Zustandsänderung bei positiver und/oder negativer Taktflanke



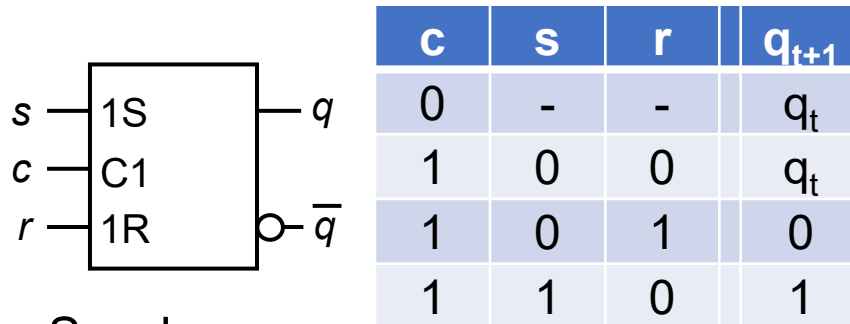
Taktzustands(-pegel)steuerung

- Eingänge wirken sich während der positiven **Taktphase** aus
- Zustand kann sich mehrmals innerhalb der selben Phase ändern
 - Auswertungszeitpunkte sind nicht klar definiert, nur Intervall
- Wird auch als „Pegelsteuerung“ bezeichnet
- Taktzustandsgesteuerte Elemente heißen (synchrone) „**Latches**“

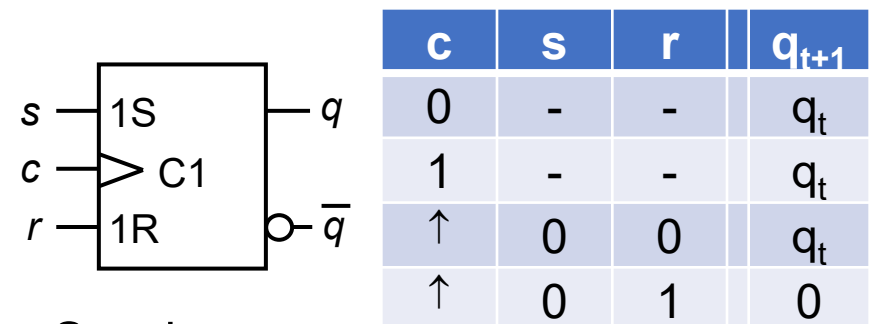
Taktflankensteuerung

- Eingänge wirken sich nur während der **Taktflanke** aus
 - Positive Flanke: $0 \rightarrow 1$
 - Negative Flanke: $1 \rightarrow 0$
- Vorteil: Auswertungszeitpunkte sind exakter definiert
- Taktflankengesteuerte Elemente heißen „**Flipflops**“

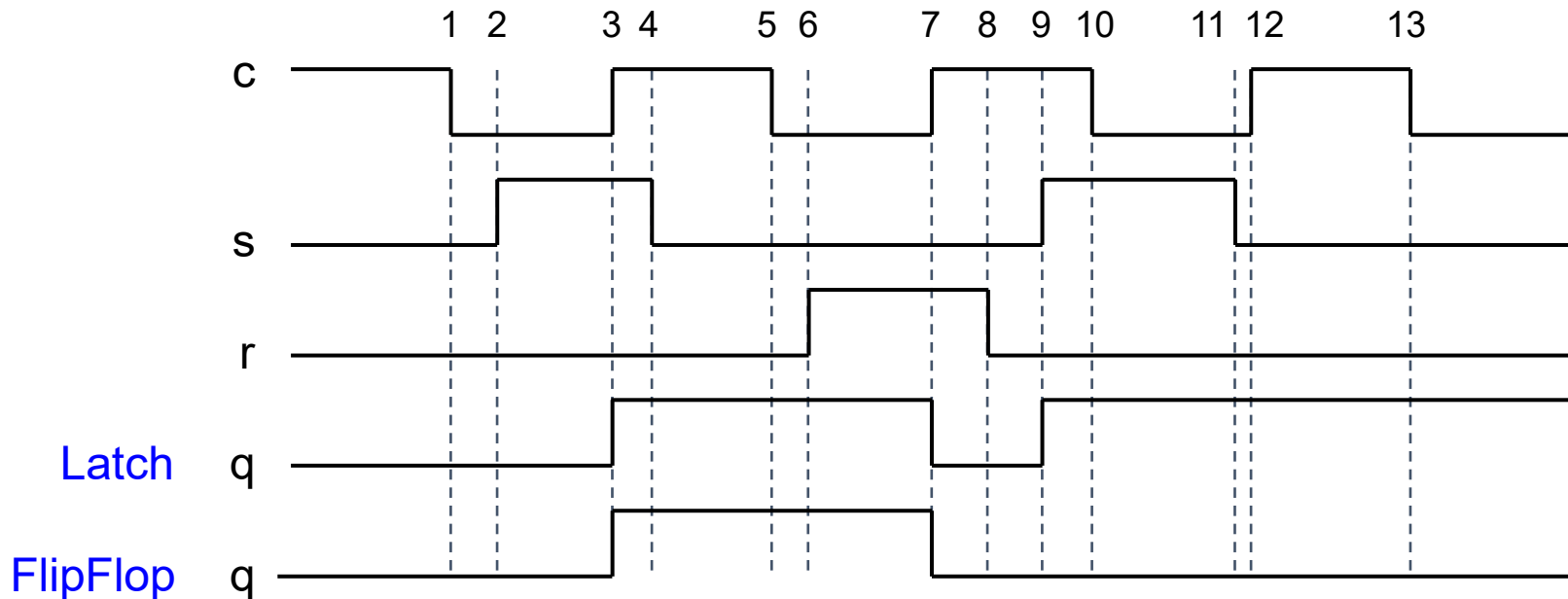
Synchrones RS-Latch und RS-Flipflop



Synchrones
RS-Latch



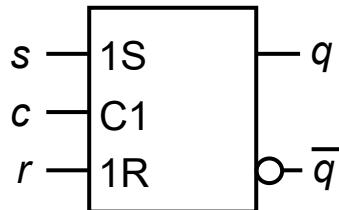
Synchrones
RS-Flipflop



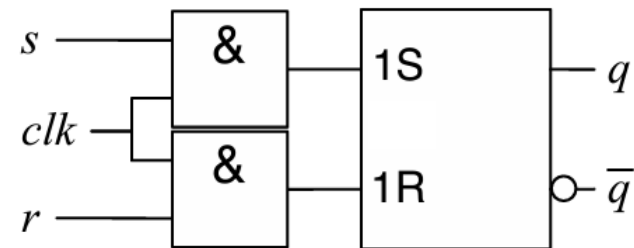


Synchrones RS-Latch (taktzustandsgesteuert)

Schaltsymbol



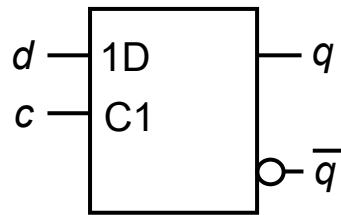
Interner Aufbau



Synchrones RS-Latch
= Asynchrones RS-Latch
+ Vorverarbeitung für Takt

Synchrones D-Latch (taktzustandsgesteuert)

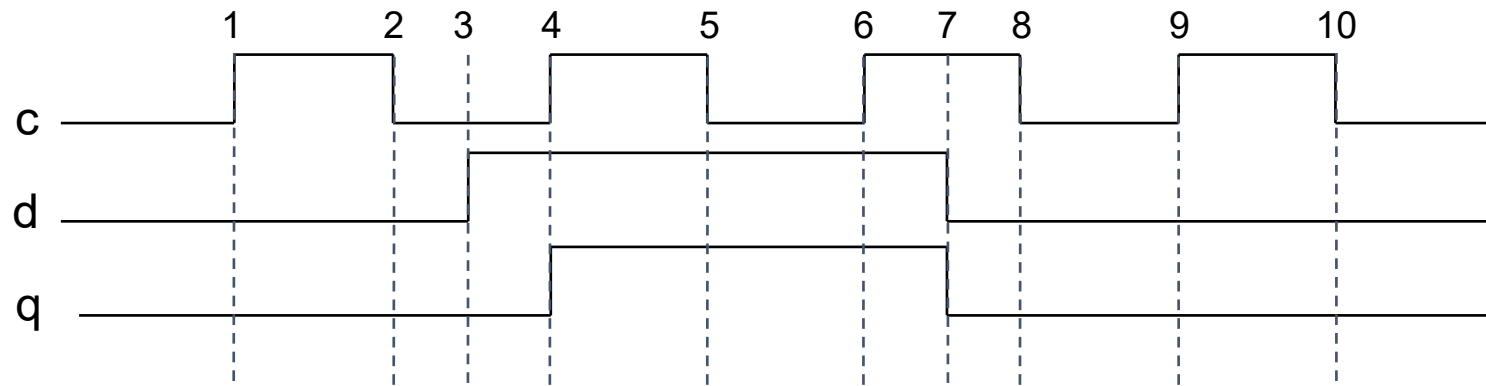
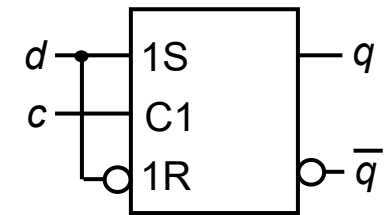
Schaltsymbol



Schaltverhalten

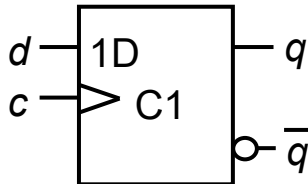
c	d	q_{t+1}
0	-	q_t
1	0	0
1	1	1

Innenansicht

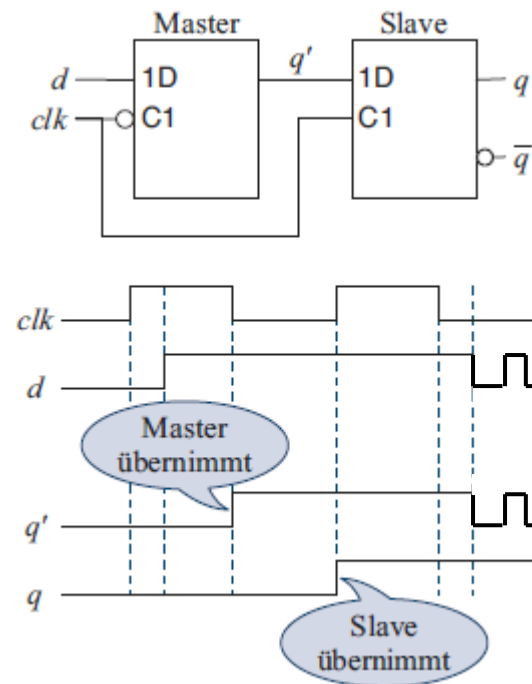


D-Flipflop (taktflankengesteuert)

Schaltsymbol



Interner Aufbau

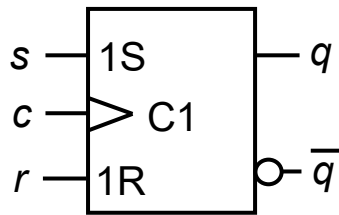


Synchrones D-Flipflop
aus 2 * Synchrones D-Latch

(Synchrones) RS-Flipflop

RS-Flipflop (taktflankengesteuert)

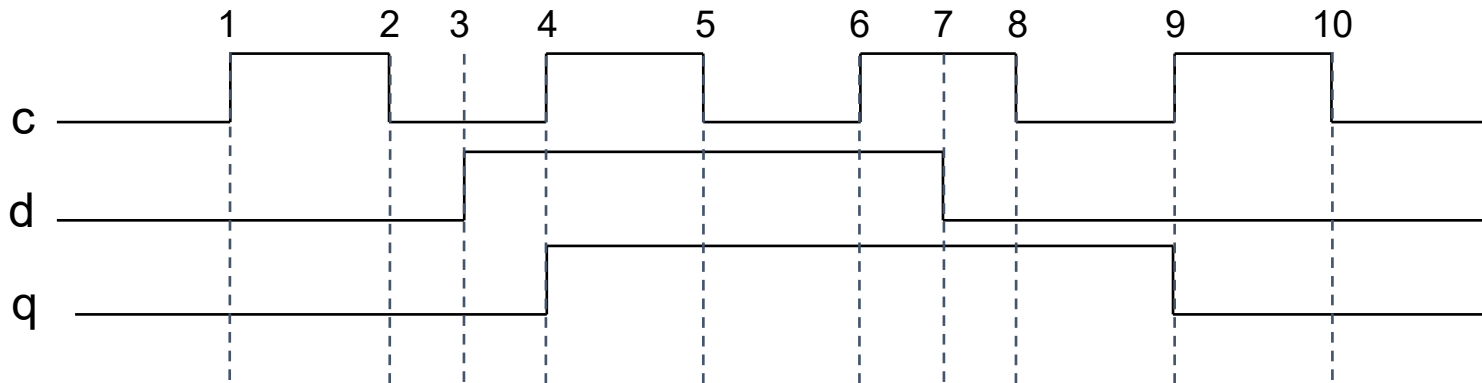
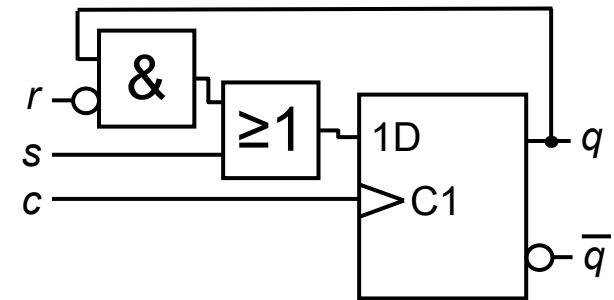
Schaltsymbol



Schaltverhalten

c	s	r	q_{t+1}
0/1/↓	-	-	q_t
↑	0	0	q_t
↑	0	1	0
↑	1	0	1

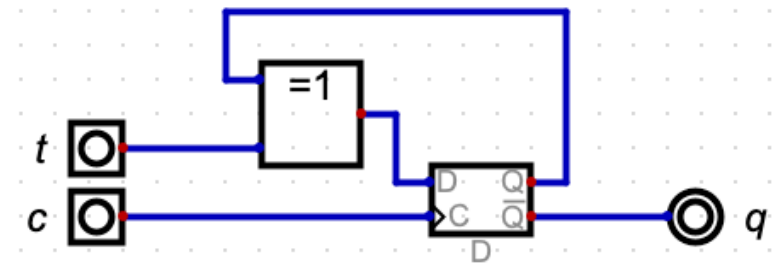
Innenansicht



Schaltsymbol

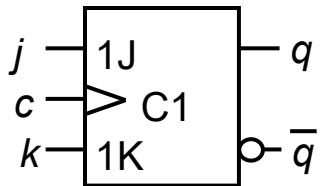


Innenansicht



JK-Flipflop – Kombination aus RS-Flipflop und T-Flipflop

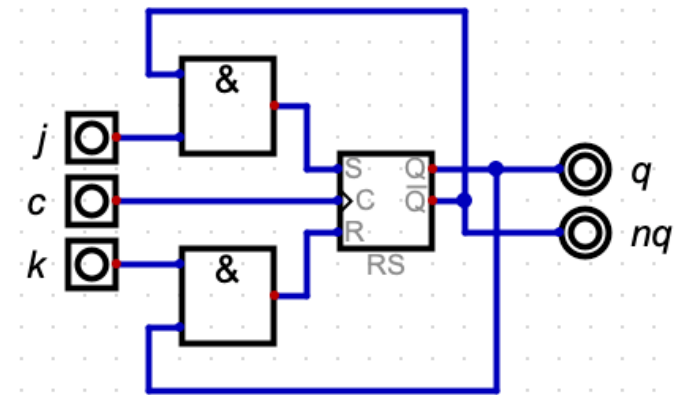
Schaltsymbol

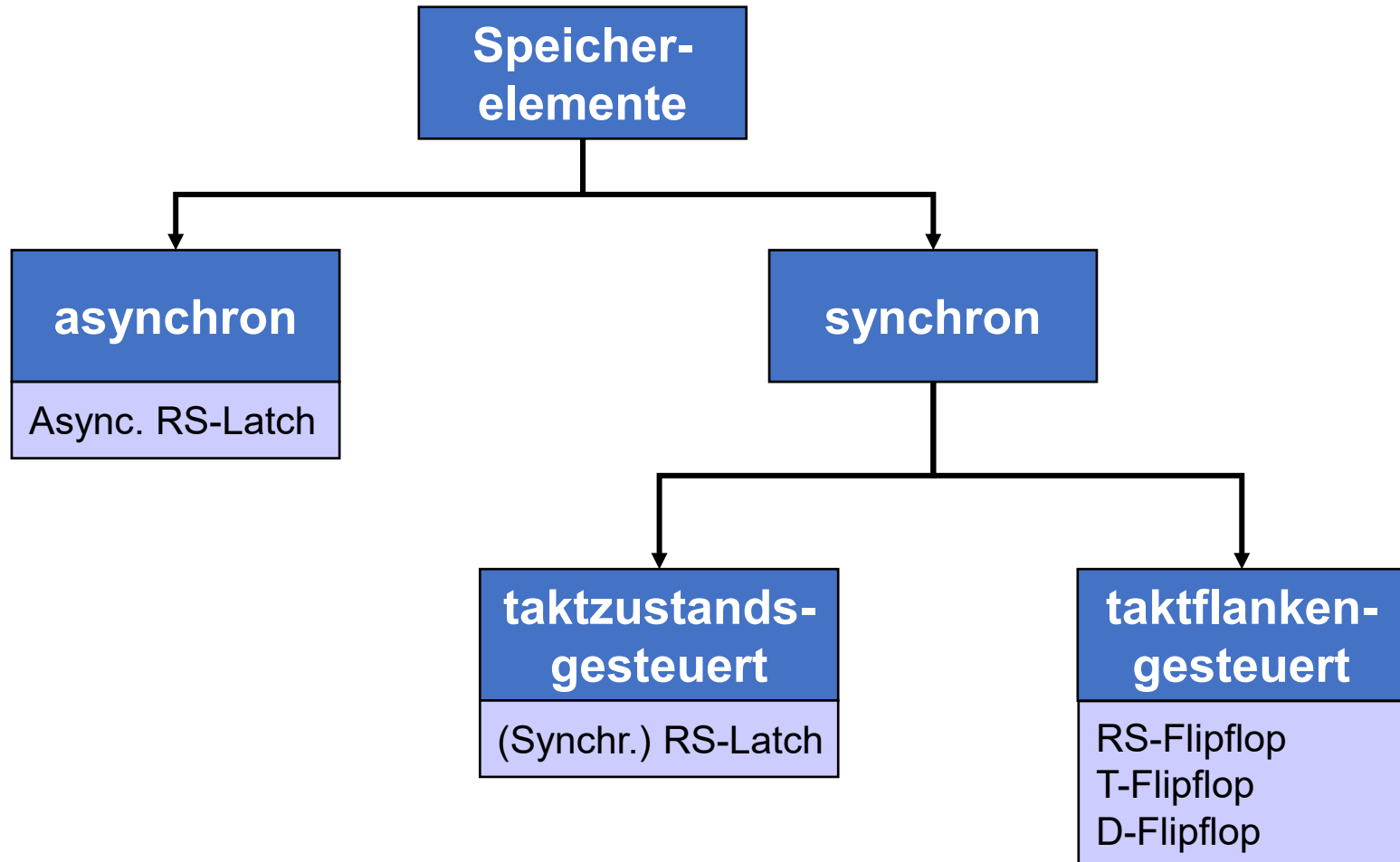


Schaltverhalten

c	j	k	q_{t+1}
0/1/↓	-	-	q_t
↑	0	0	q_t
↑	0	1	0
↑	1	0	1
↑	1	1	$\neg q_t$

Innenansicht





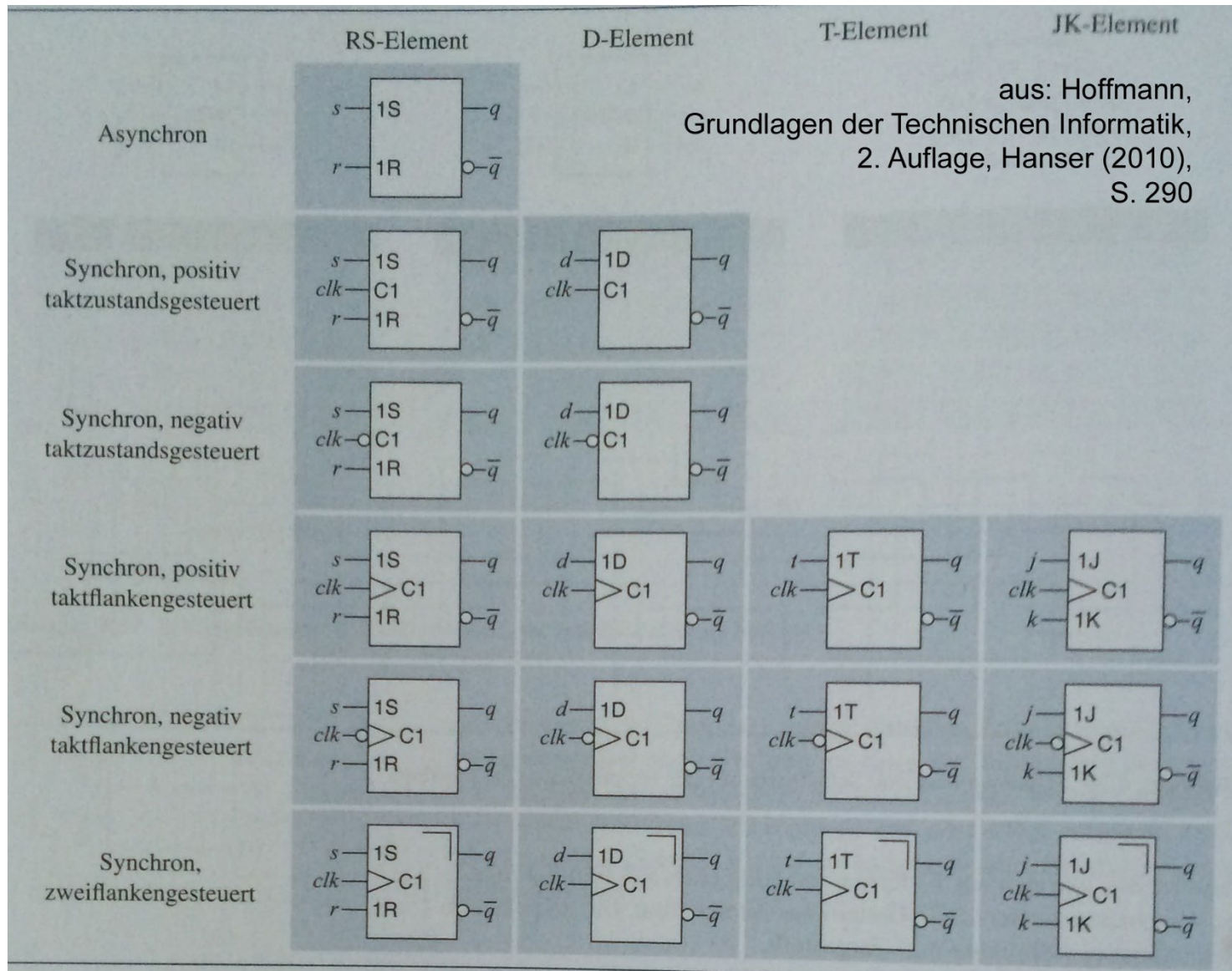
Schaltnetze (Kombinatorische Logik)

- In diese Gruppe fallen die zuvor betrachteten Schaltungen
- Aufbau durch elementare logische Funktionen (Gatter)
- Ausgabe hängt nur und unmittelbar von der **aktuellen** Eingabe ab (keine Rückkopplung von Signalen)
- Kein Gedächtnis (keine Speicherelemente, **zustandslos**)
- Asynchron (keine Taktleitung)

Schaltwerke (Sequentielle Logik)

- Ausgabe hängt von **allen vorangegangenen** Eingaben ab
- Schaltungen verfügen über ein „Gedächtnis“ (**zustandsbehaftet**)
- **Einsatz von Speicherelementen** (Flipflops oder Latches)
- häufig Synchron (Taktsignal)

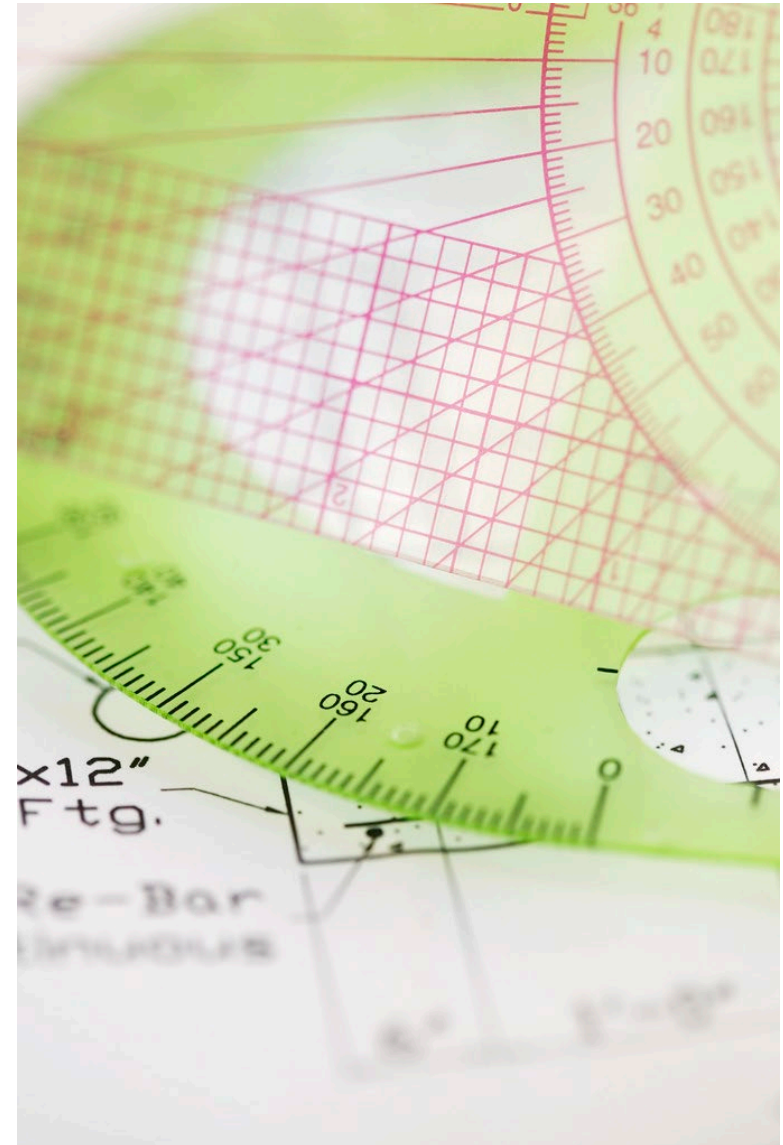
Übersicht Latches und Flipflops



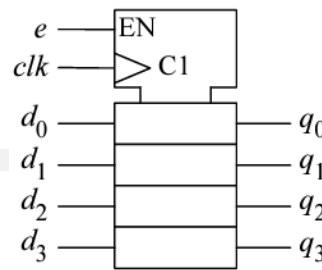
Agenda

- Einfache Speicherelemente
- Taktgesteuerte Elemente

 Register

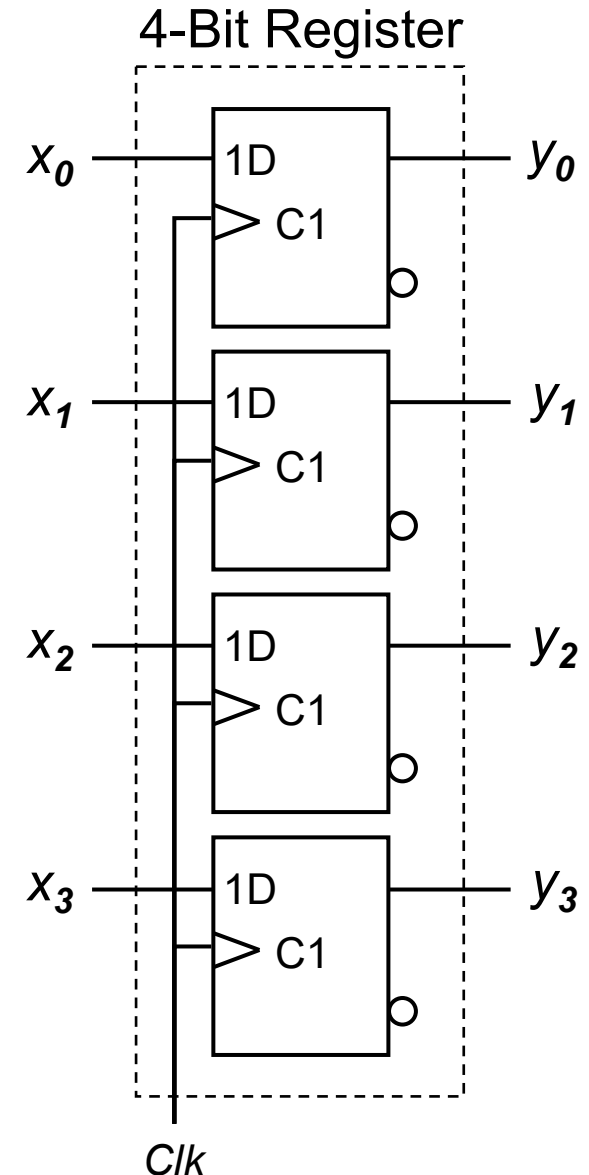


Register



Anwendungsbeispiel

- Aufbau
 - Aneinanderreihung von einzelnen Flipflops
 - "Breite" des Registers = Anzahl der Flipflops
 - Typische Bit-Breiten: 8, 16, 32, 64, 128
 - Alle Flipflops teilen sich dieselbe Taktleitung
- Anwendung: Schneller Speicher in Prozessoren
 - für den Programmierer und direkt ansprechbar
 - für Zwischenergebnisse, Anzahl variiert
- Eingänge und Ausgänge
 - n-fach entsprechend den Einzelflipflops:
z.B. D, Set oder Reset
 - 1-fach (ein Eingang wirkt auf alle Flipflops des Registers): Clock, Enable
- Variationen (später!)
 - Schieberegister
 - Universalregister, Akkumulator



Agenda

- Einfache Speicherelemente
- Taktgesteuerte Elemente
- Register

