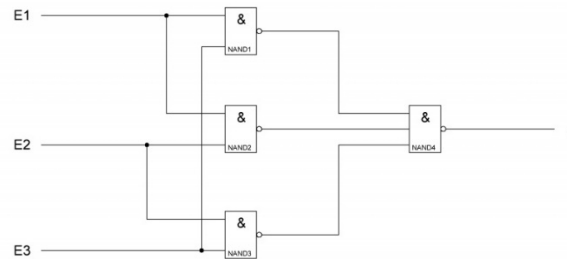


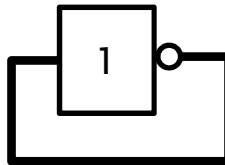
FlipFlops (FFs) / bistabile Kippschaltungen

Motivation - Rückkopplungen in Schaltnetzen

- Gatter können zu Schaltnetzen zusammen-geschaltet werden, um komplexe Funktionen zu realisieren



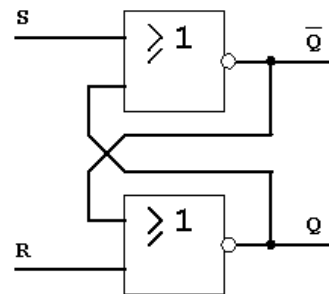
- Bisher: Informationsfluss vorwärts gerichtet. Was passiert, wenn es Rückführungen gibt?



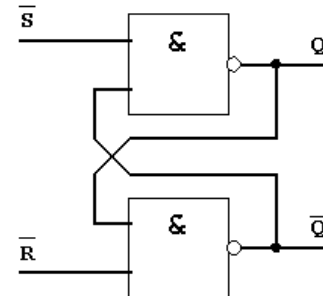
- Gibt es sinnvolle Anwendungen für Rückführungen?

Motivation - Rückkopplungen in Schaltnetzen - Anwendung Latches

NOR:



NAND:

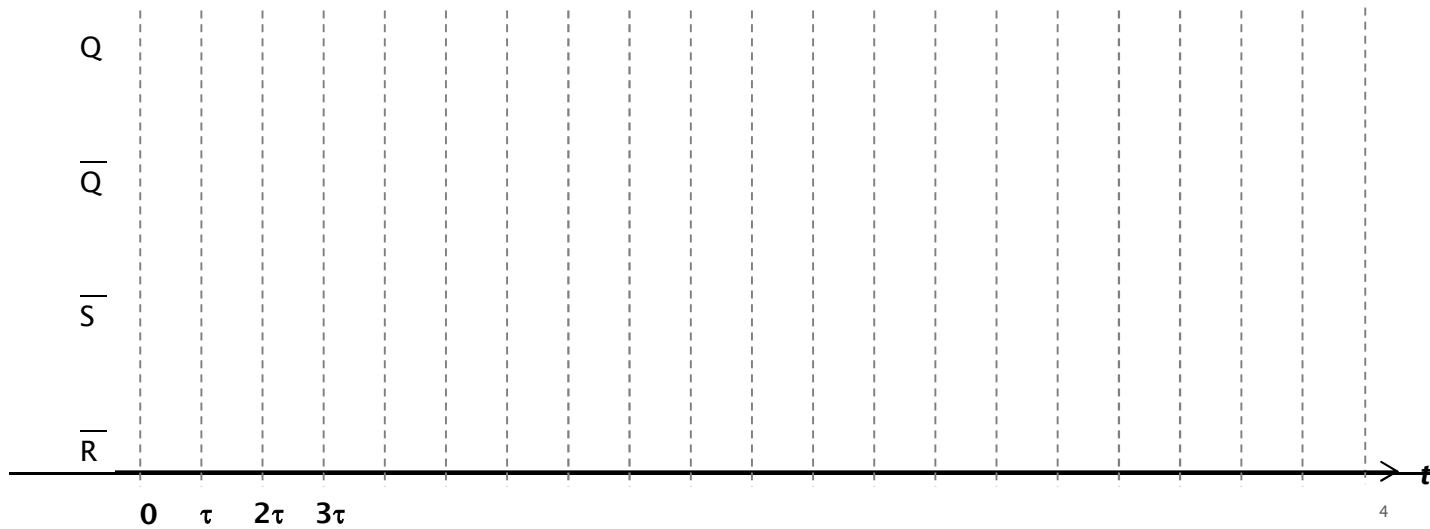
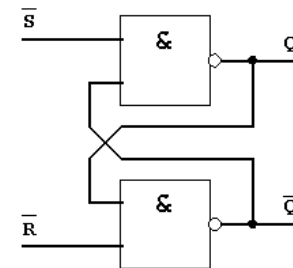


- Schaltungen dieser Art nennt man bistabiles Kippglied, bistabile Kippschaltung oder (asynchrones) Latch

NOR-FF			Funktion	NAND-FF		
S	R	Q ⁺		\bar{S}	\bar{R}	Q ⁺
1	0	1	Setzen	0	1	1
0	1	0	Rücksetzen	1	0	0
0	0	Q	Speichern	1	1	Q
1	1	-	(verboten)	0	0	-

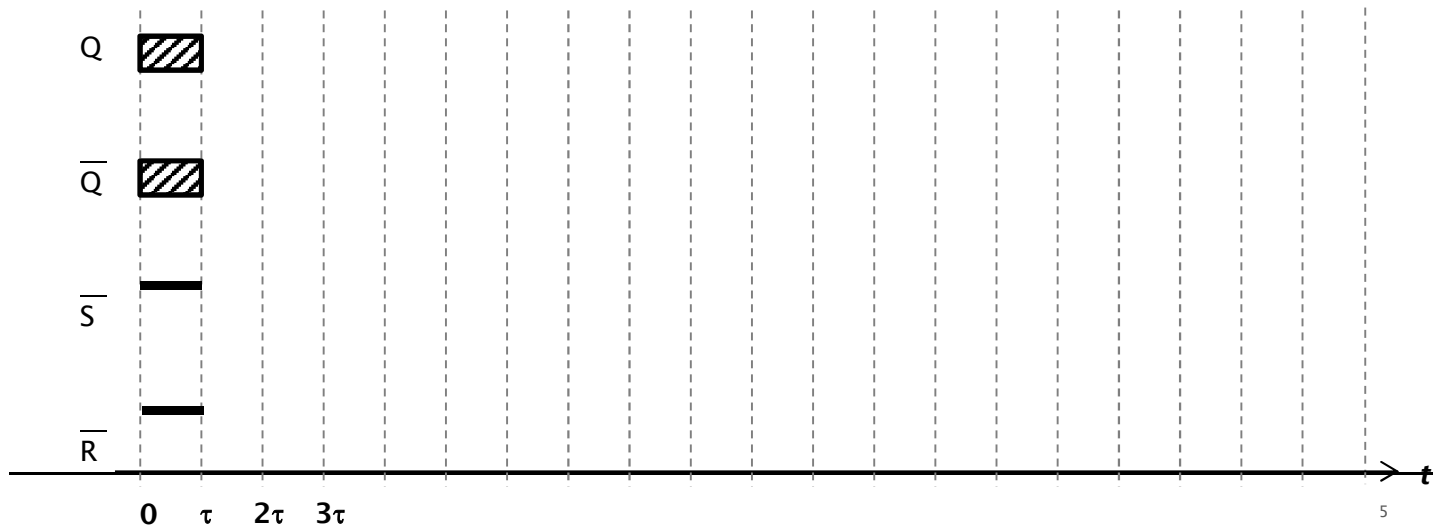
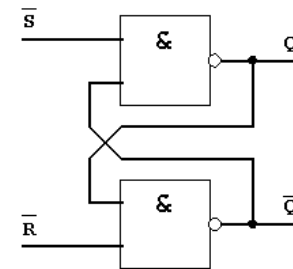
(Asynchrones) RS-Latch aus NAND-Gattern

A	B	NAND(A,B)
0	0	1
0	1	1
1	0	1
1	1	0



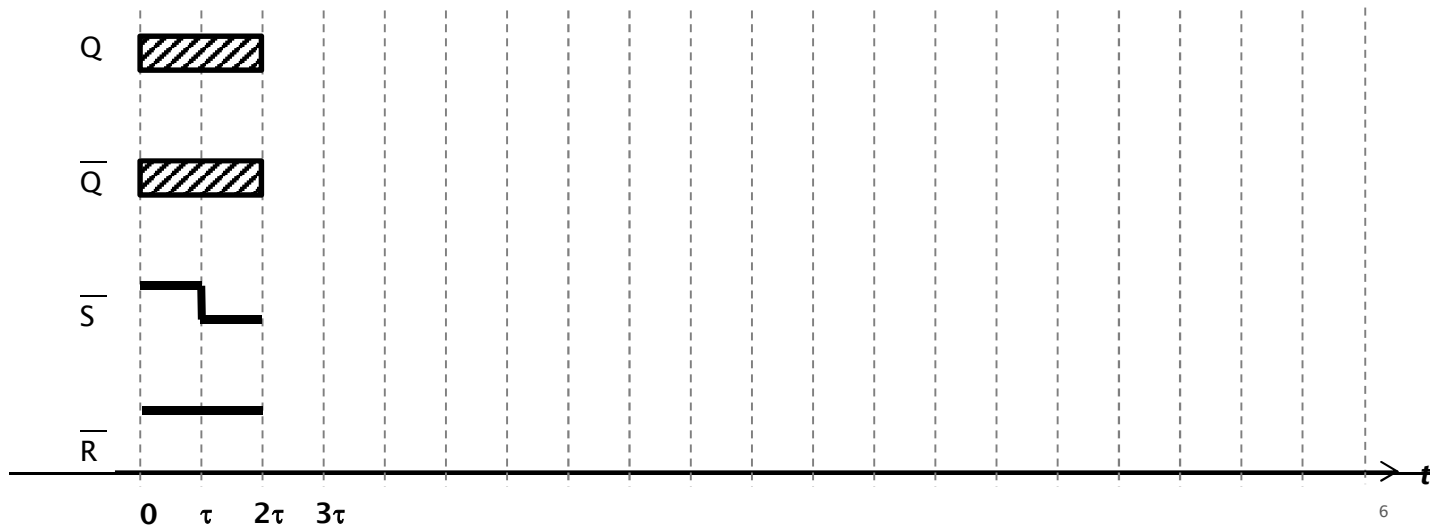
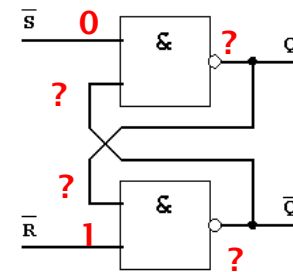
(Asynchrones) RS-Latch aus NAND-Gattern

A	B	NAND(A,B)
0	0	1
0	1	1
1	0	1
1	1	0



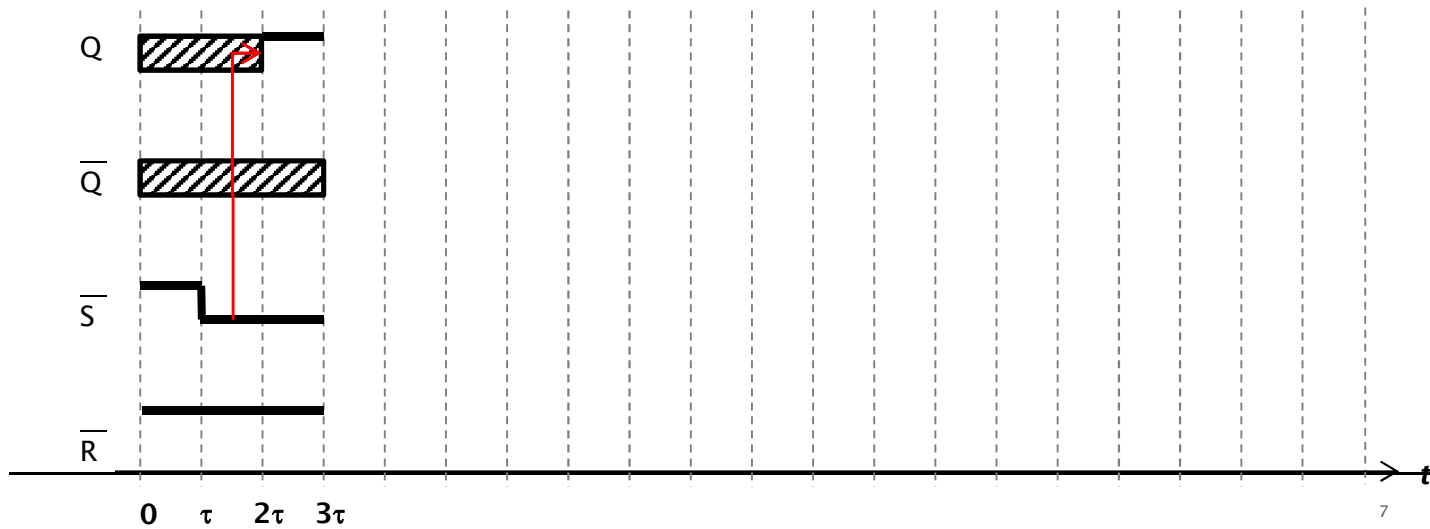
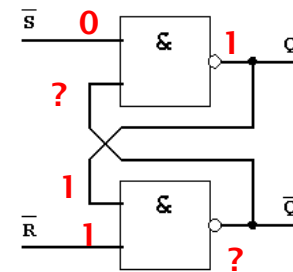
(Asynchrones) RS-Latch aus NAND-Gattern

A	B	NAND(A,B)
0	0	1
0	1	1
1	0	1
1	1	0



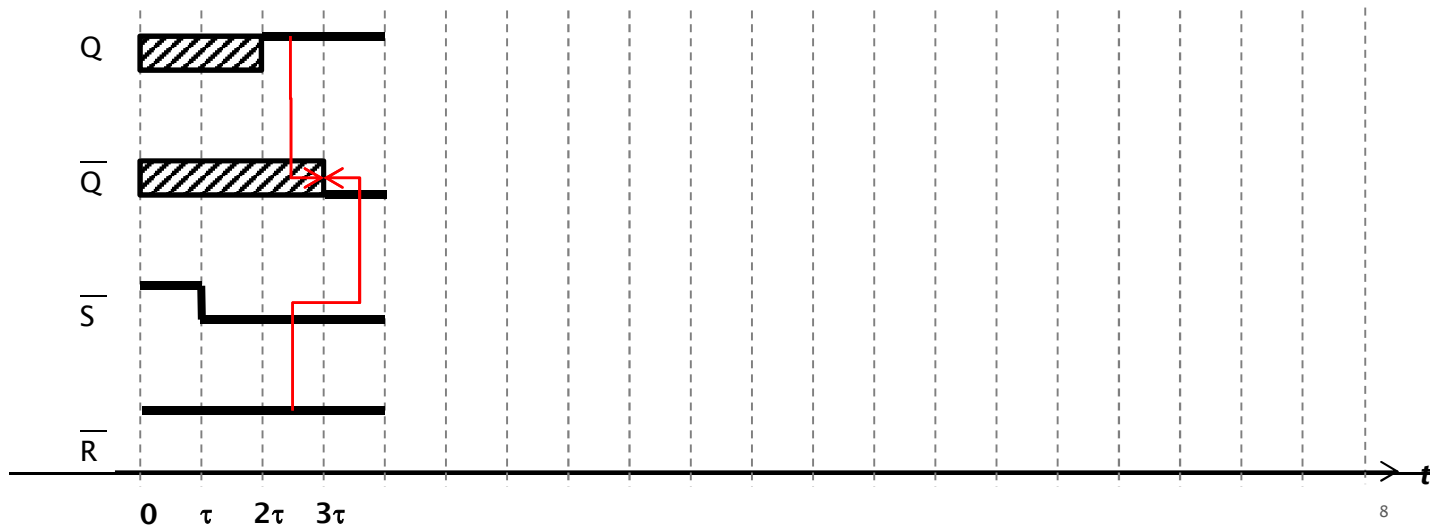
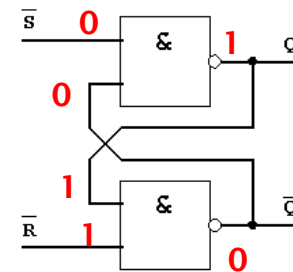
(Asynchrones) RS-Latch aus NAND-Gattern

A	B	NAND(A,B)
0	0	1
0	1	1
1	0	1
1	1	0



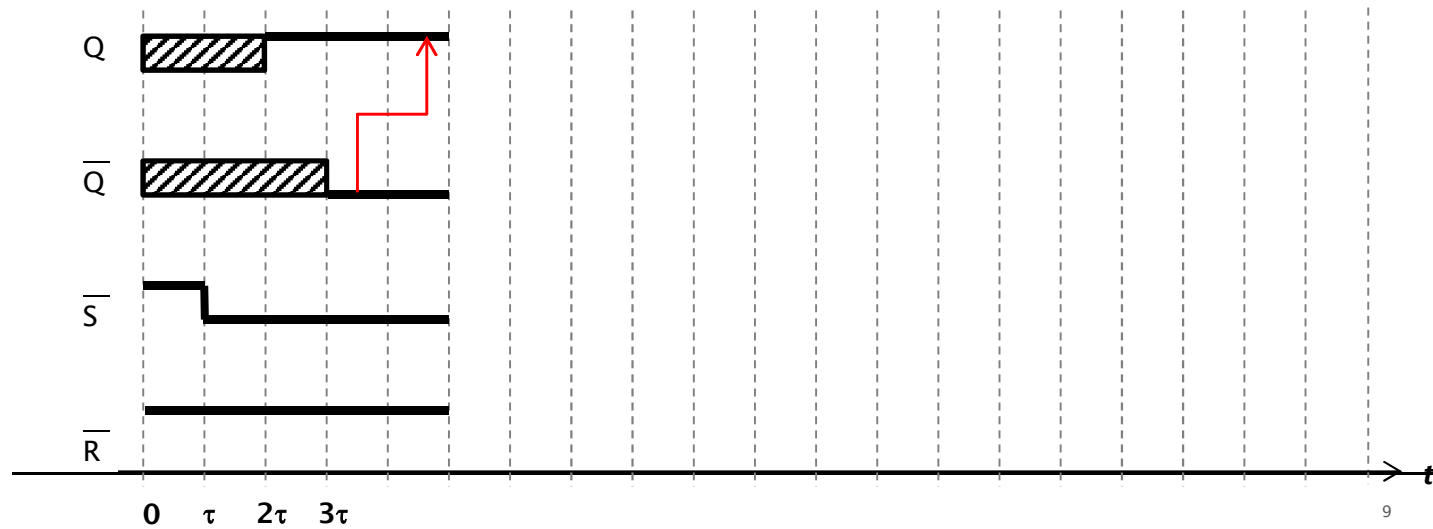
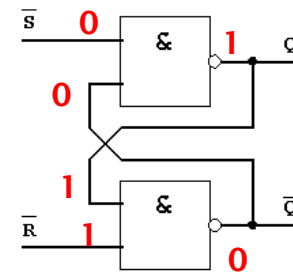
(Asynchrones) RS-Latch aus NAND-Gattern

A	B	NAND(A,B)
0	0	1
0	1	1
1	0	1
1	1	0



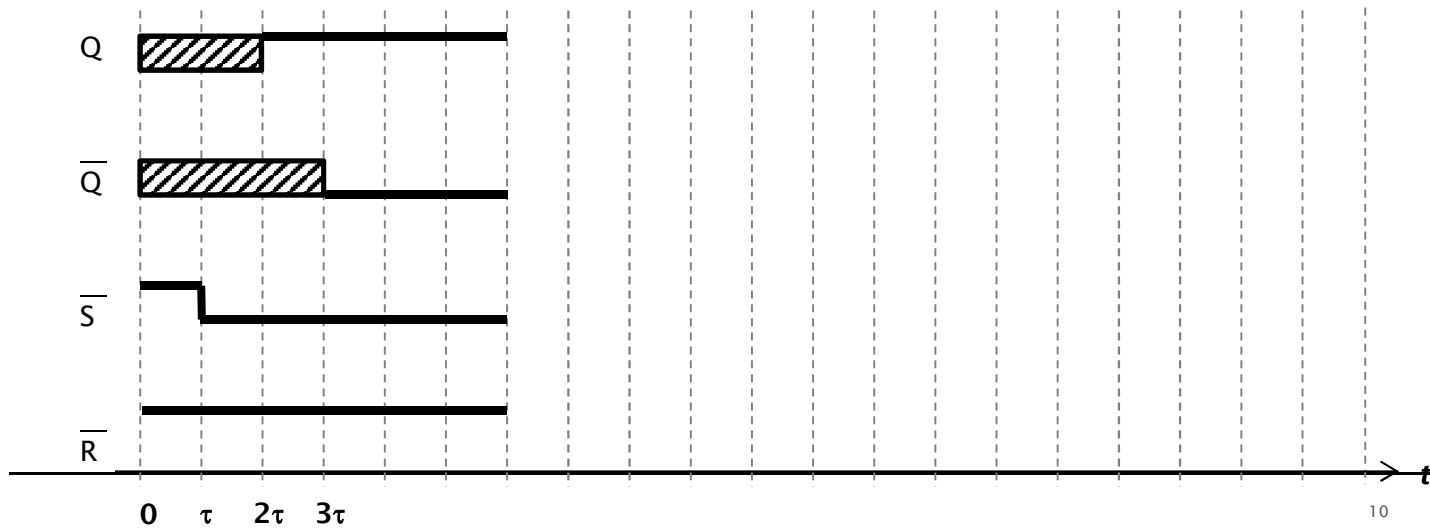
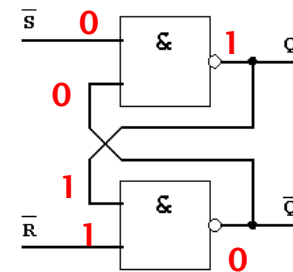
(Asynchrones) RS-Latch aus NAND-Gattern

A	B	NAND(A,B)
0	0	1
0	1	1
1	0	1
1	1	0



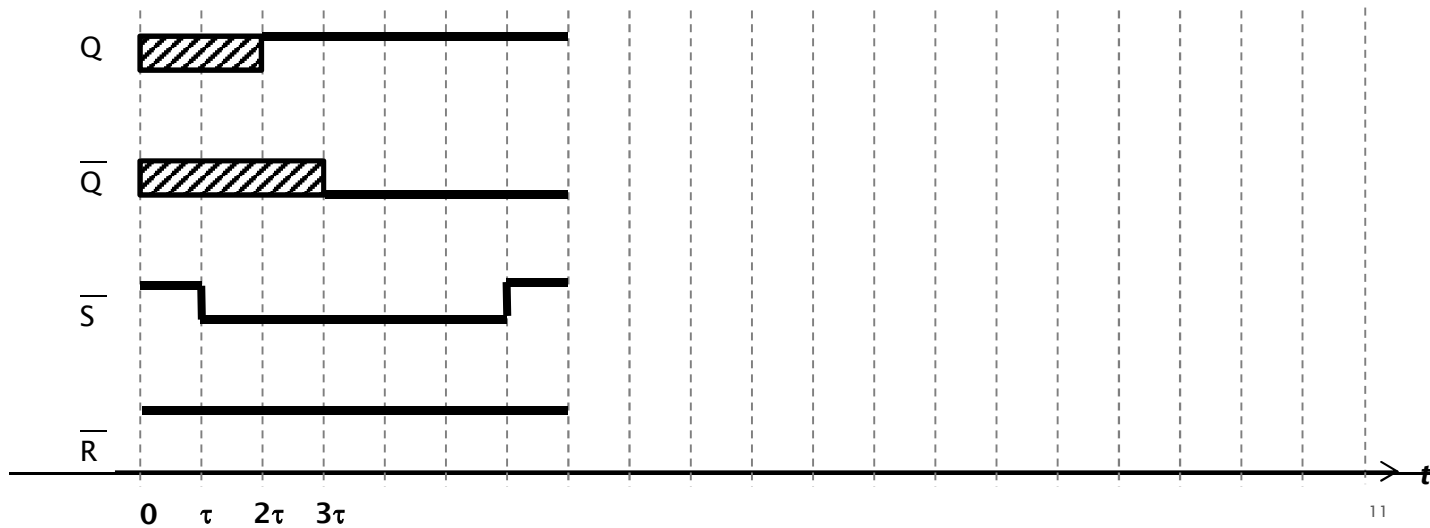
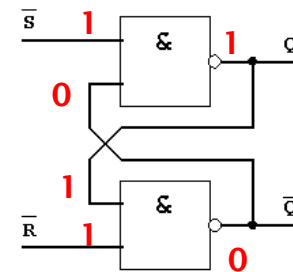
(Asynchrones) RS-Latch aus NAND-Gattern

A	B	NAND(A,B)
0	0	1
0	1	1
1	0	1
1	1	0



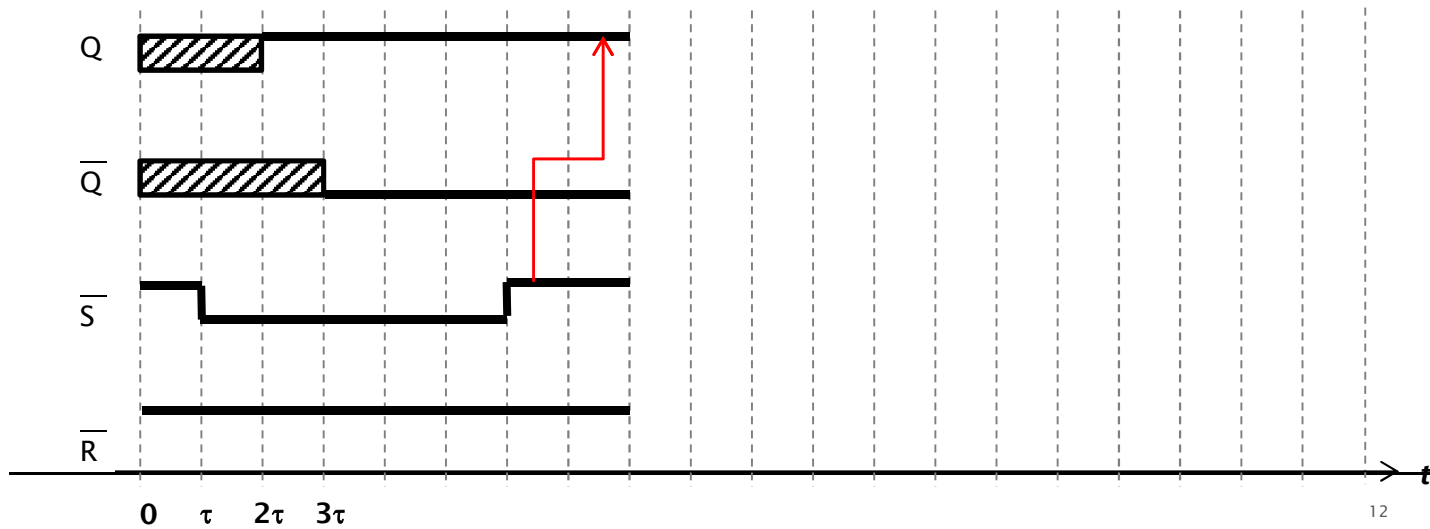
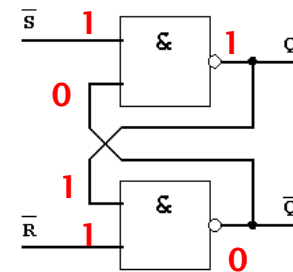
(Asynchrones) RS-Latch aus NAND-Gattern

A	B	NAND(A,B)
0	0	1
0	1	1
1	0	1
1	1	0



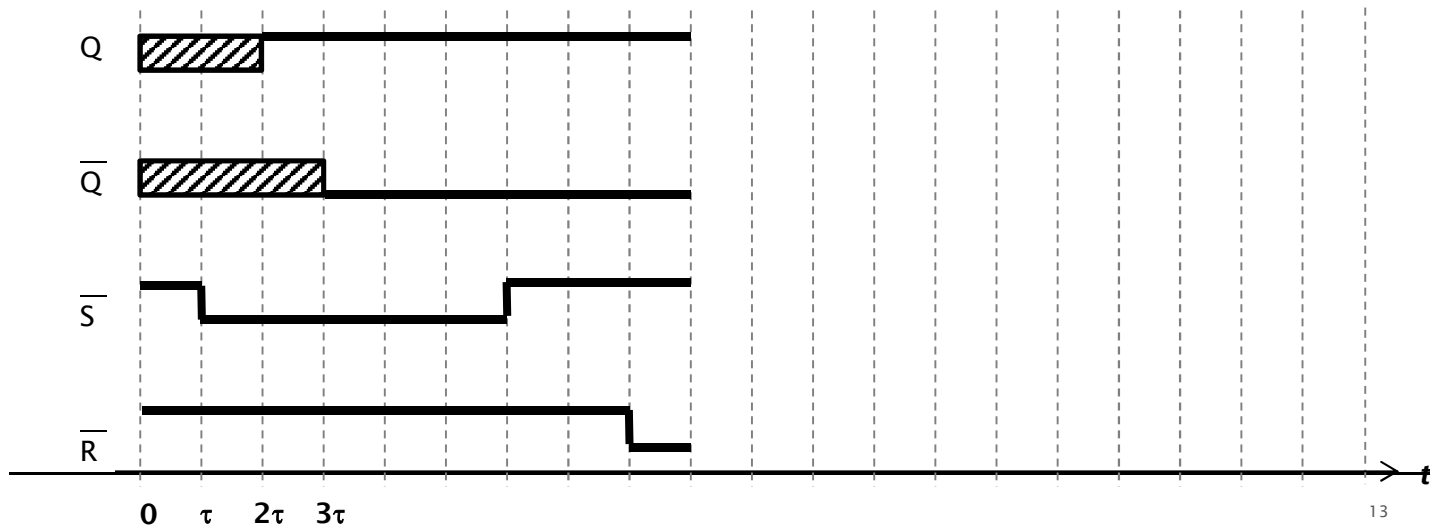
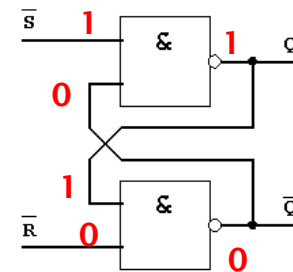
(Asynchrones) RS-Latch aus NAND-Gattern

A	B	NAND(A,B)
0	0	1
0	1	1
1	0	1
1	1	0



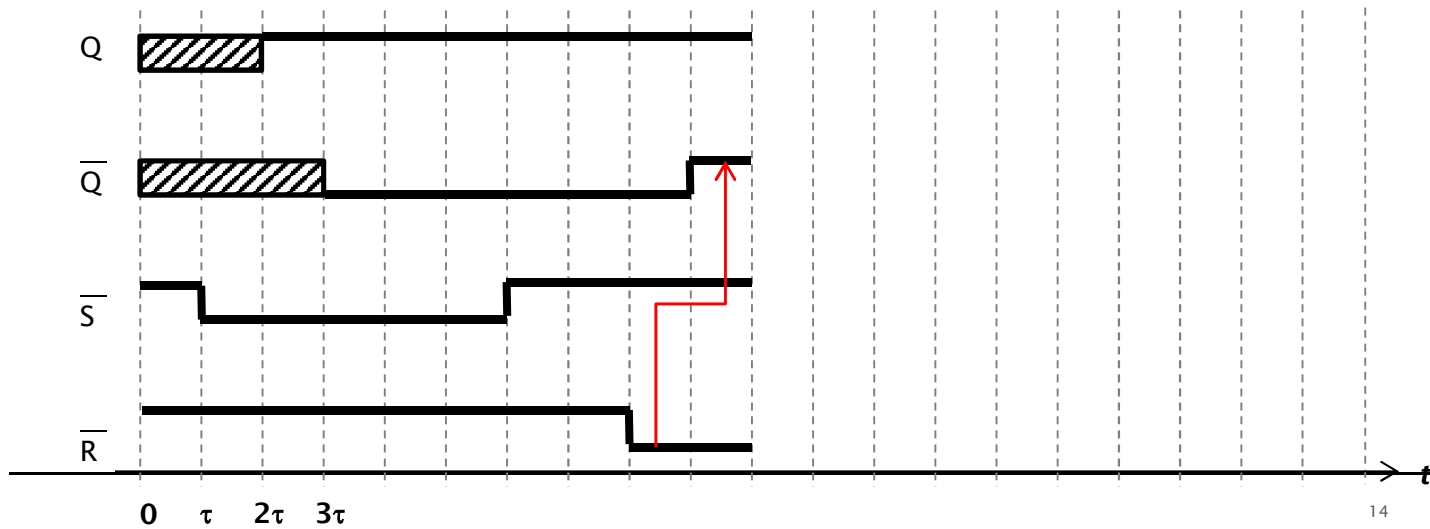
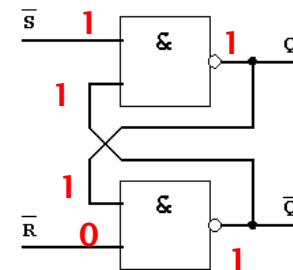
(Asynchrones) RS-Latch aus NAND-Gattern

A	B	NAND(A,B)
0	0	1
0	1	1
1	0	1
1	1	0



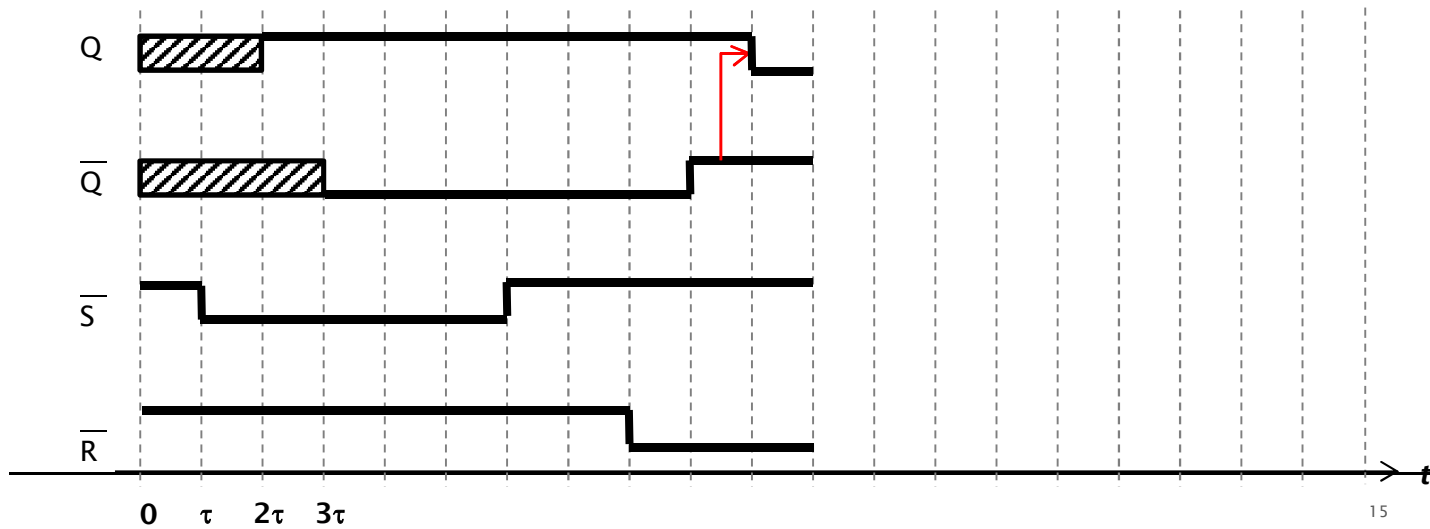
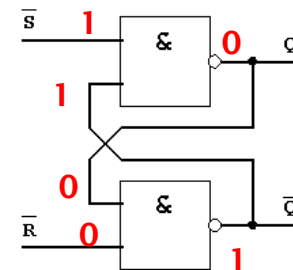
(Asynchrones) RS-Latch aus NAND-Gattern

A	B	NAND(A,B)
0	0	1
0	1	1
1	0	1
1	1	0



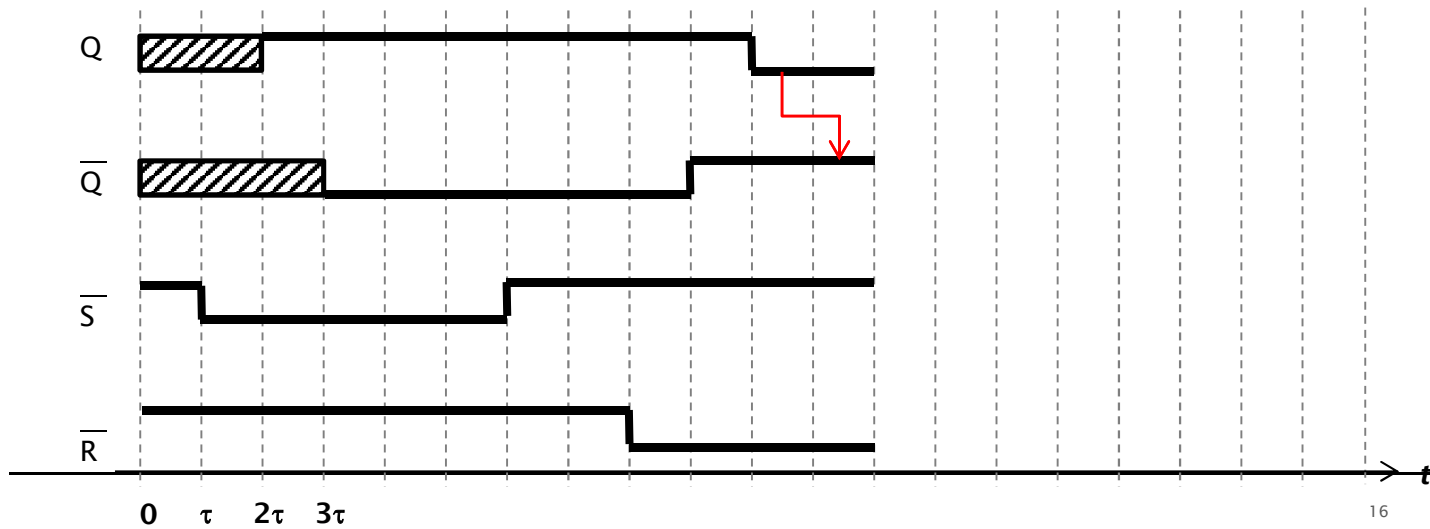
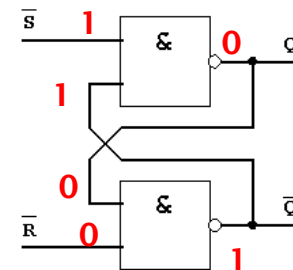
(Asynchrones) RS-Latch aus NAND-Gattern

A	B	NAND(A,B)
0	0	1
0	1	1
1	0	1
1	1	0



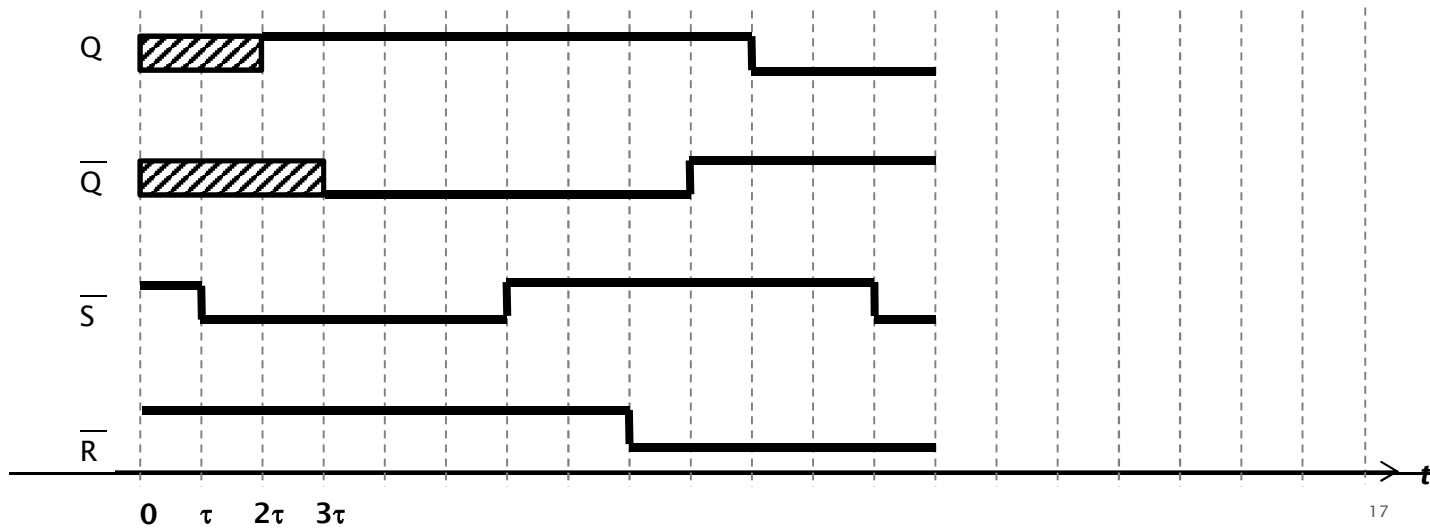
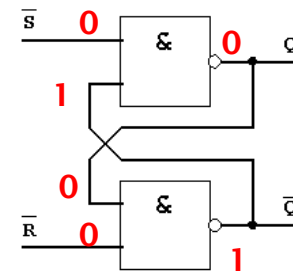
(Asynchrones) RS-Latch aus NAND-Gattern

A	B	NAND(A,B)
0	0	1
0	1	1
1	0	1
1	1	0



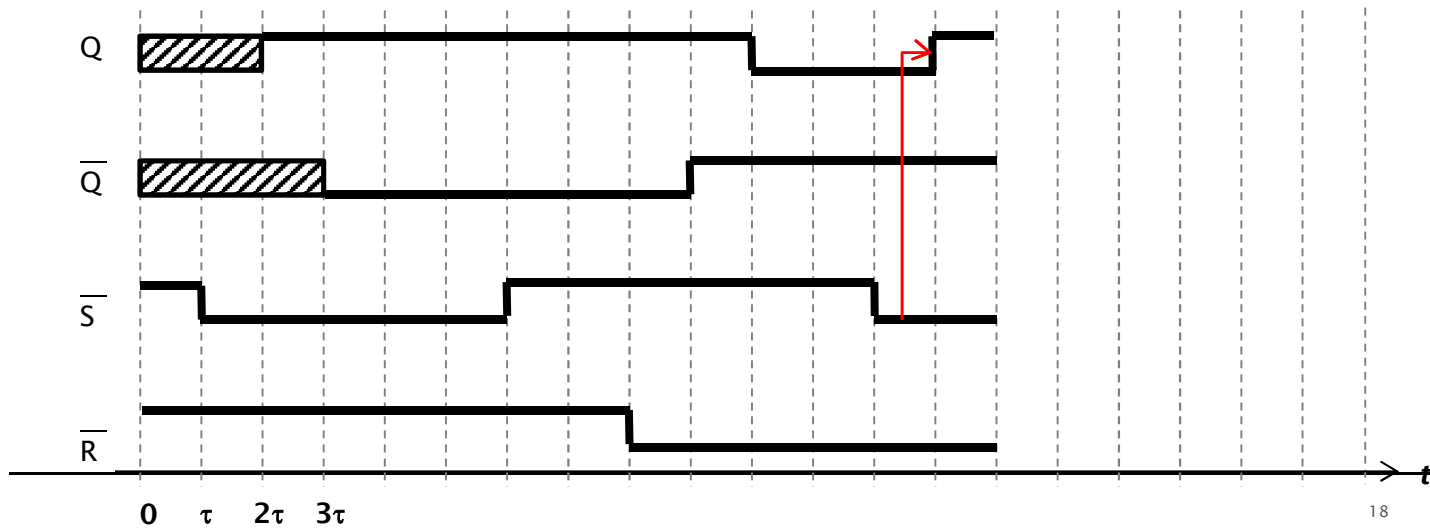
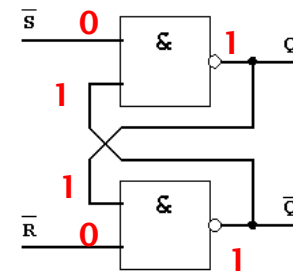
(Asynchrones) RS-Latch aus NAND-Gattern

A	B	NAND(A,B)
0	0	1
0	1	1
1	0	1
1	1	0



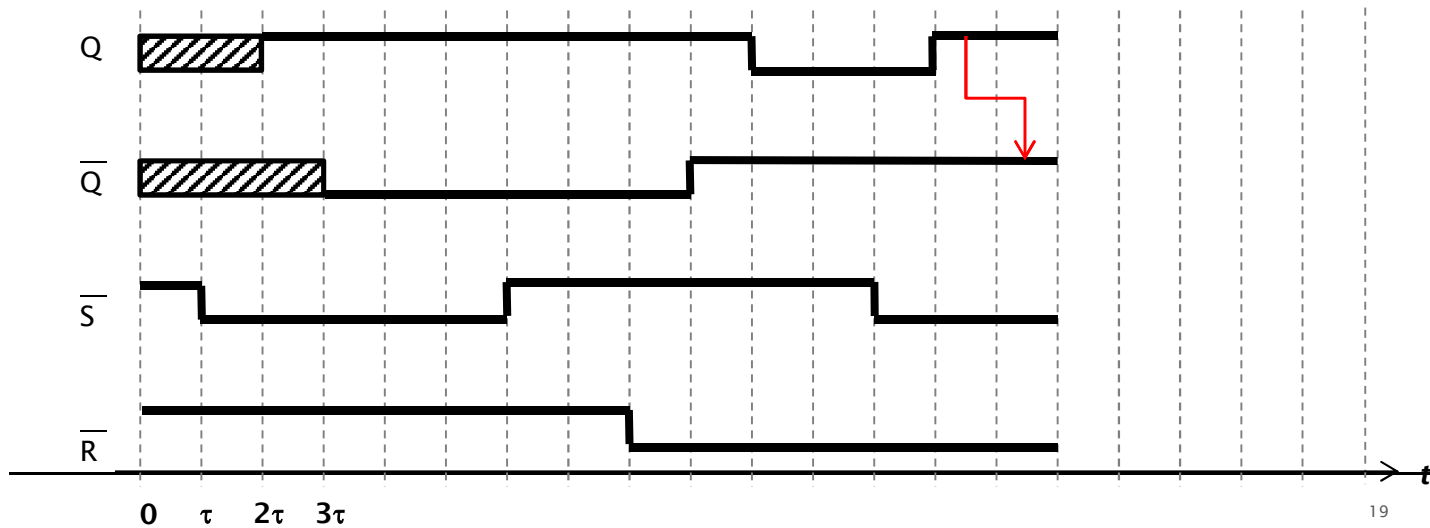
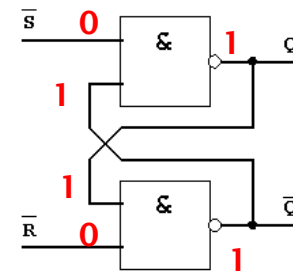
(Asynchrones) RS-Latch aus NAND-Gattern

A	B	NAND(A,B)
0	0	1
0	1	1
1	0	1
1	1	0



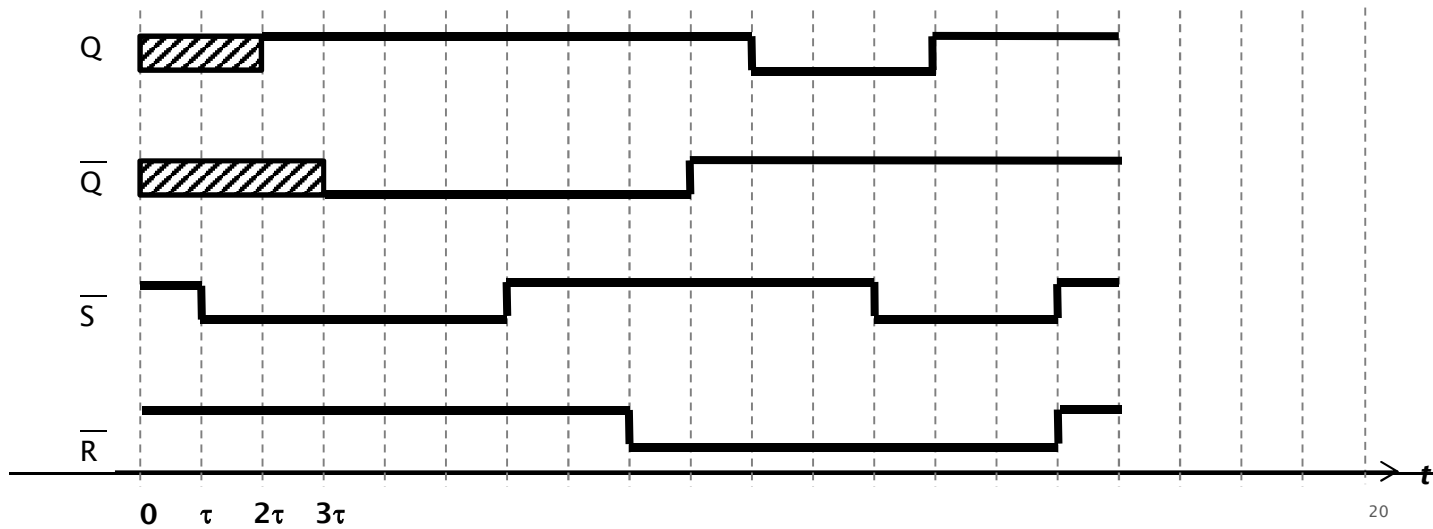
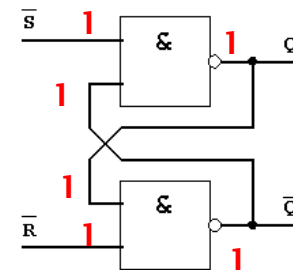
(Asynchrones) RS-Latch aus NAND-Gattern

A	B	NAND(A,B)
0	0	1
0	1	1
1	0	1
1	1	0



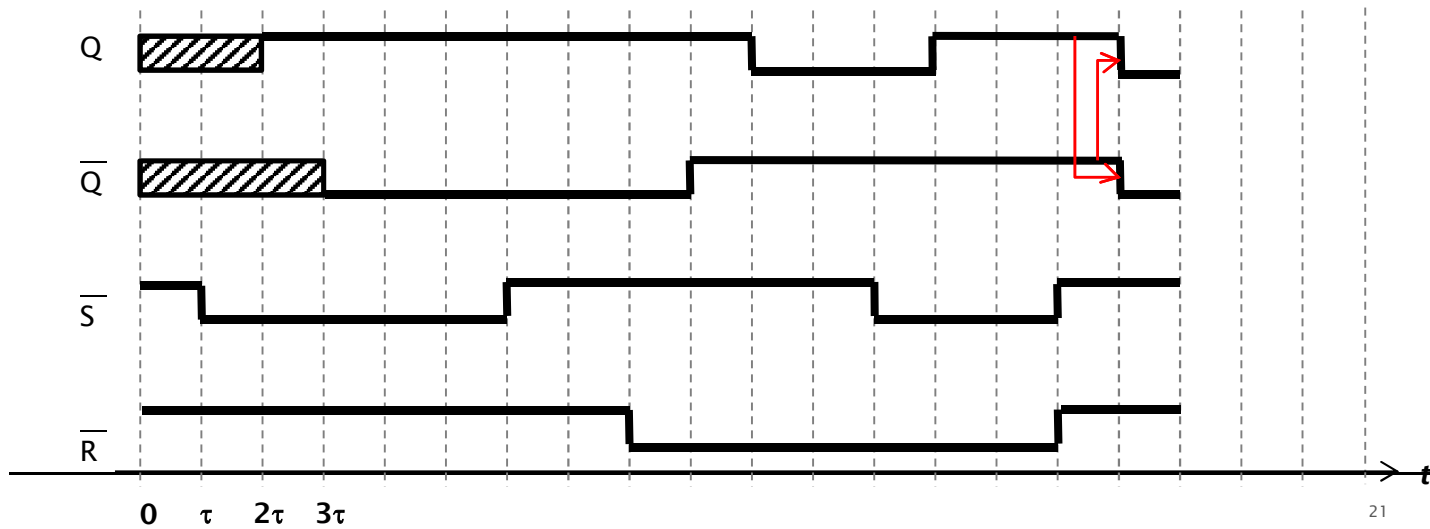
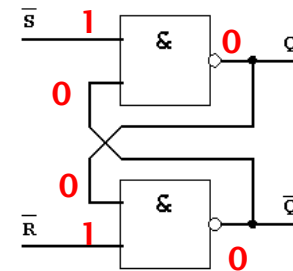
(Asynchrones) RS-Latch aus NAND-Gattern

A	B	NAND(A,B)
0	0	1
0	1	1
1	0	1
1	1	0



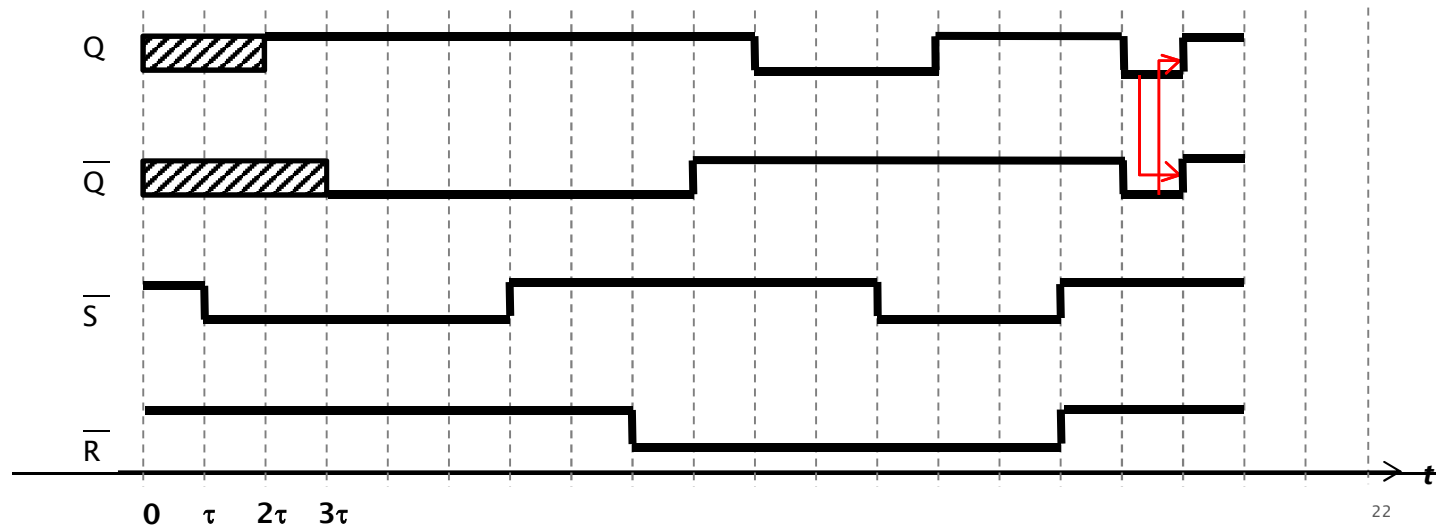
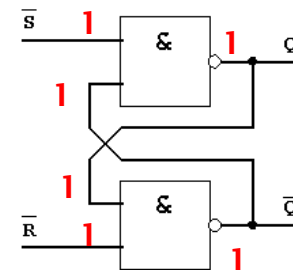
(Asynchrones) RS-Latch aus NAND-Gattern

A	B	NAND(A,B)
0	0	1
0	1	1
1	0	1
1	1	0



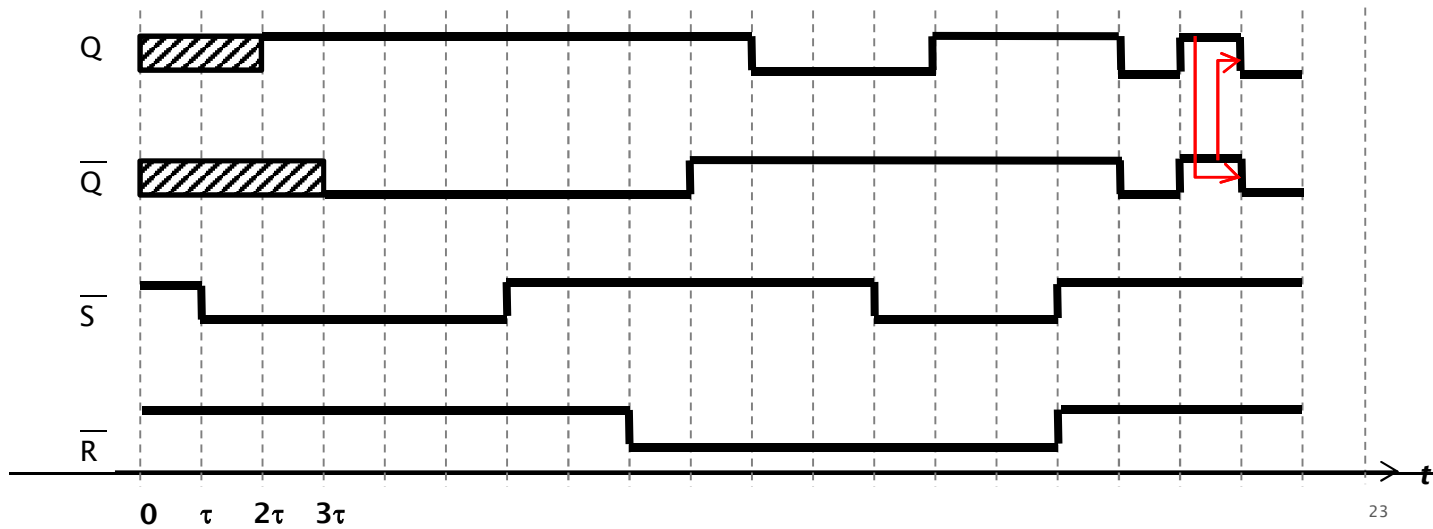
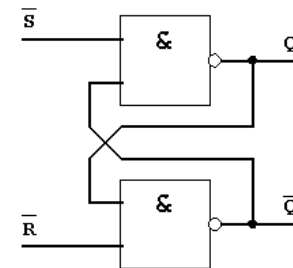
(Asynchrones) RS-Latch aus NAND-Gattern

A	B	NAND(A,B)
0	0	1
0	1	1
1	0	1
1	1	0



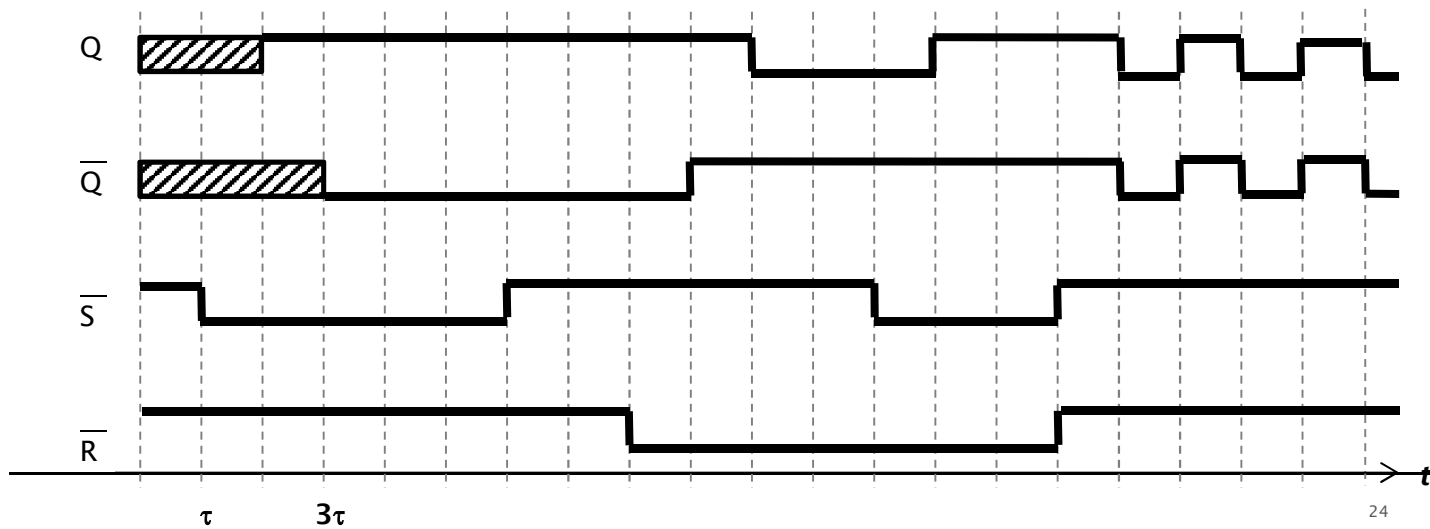
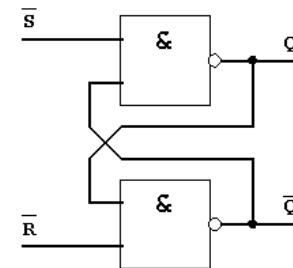
(Asynchrones) RS-Latch aus NAND-Gattern

A	B	NAND(A,B)
0	0	1
0	1	1
1	0	1
1	1	0



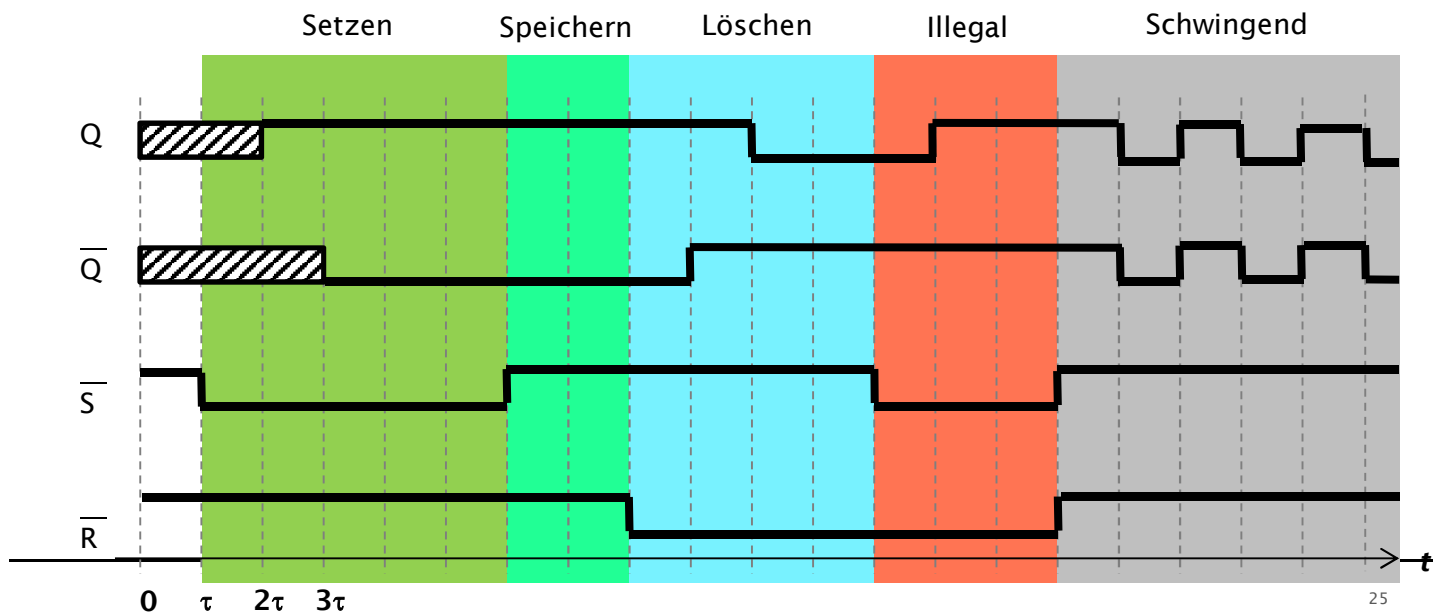
(Asynchrones) RS-Latch aus NAND-Gattern

A	B	NAND(A,B)
0	0	1
0	1	1
1	0	1
1	1	0



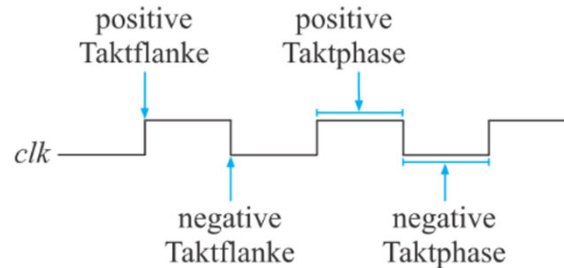
(Asynchrones) RS-Latch aus NAND-Gattern

- Setzen: $\overline{S}=0$ und $\overline{R}=1$; Speichern: $\overline{S}=1$ und $\overline{R}=1$
- Löschen: $\overline{S}=1$ und $\overline{R}=0$; illegal: $\overline{S}=0$ und $\overline{R}=0$



Synchrone Latches (Taktzustandsgeteuerte Flipflops) - Einleitung

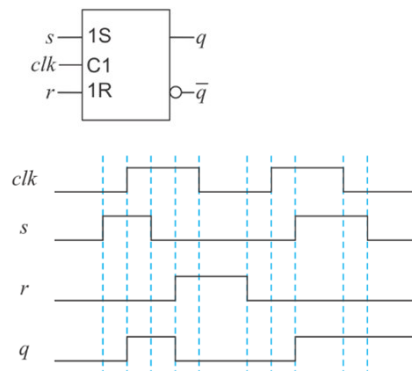
- Beim (asynchronen) RS-Latch kann eine Zustandsänderung zu jeder beliebigen Zeit erfolgen
- Im Gegensatz dazu sind Zustandsänderungen bei synchronen Schaltungen nur noch innerhalb ganz bestimmter Zeitintervalle (synchrone Latches) oder zu fest definierten Zeitpunkten möglich (Flipflops (FF))
- Die Synchronisation wird durch das Taktsignal gesteuert
- Das Taktsignal ist eine periodische Rechteckschwingung



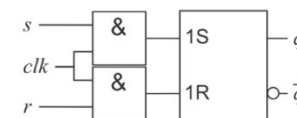
- Die Taktfrequenz f_{clk} legt fest, wie viele 1-Phasen das Taktsignal pro Sekunde erzeugt

Synchrone Latches - Synchrones RS-Latch

- Das synchrone RS-Latch reagiert während der positiven Taktphase ($clk=1$) genauso wie das asynchrone RS-Latch
- In der negativen Taktphase ($clk=0$) reagiert es nicht; der Zustand wird eingefroren; Zustandswechsel finden erst wieder bei $clk=1$ statt



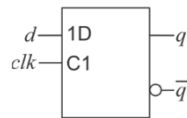
clk	r	s	q^{t+1}	
0	-	-	q^t	Speichern
1	0	0	q^t	Speichern
1	0	1	1	Setzen
1	1	0	0	Rücksetzen
1	1	1	-	Vermeiden



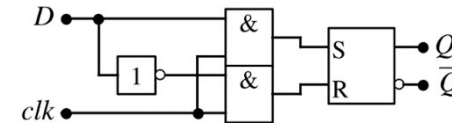
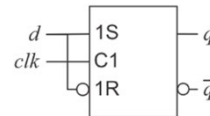
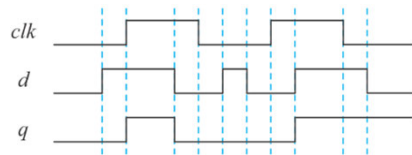
- Ein synchrone RS-Latch können wir einfach aus einem asynchronen RS-Latch aufbauen
- Die Eingänge R und S werden durch 2 UND-Gatter geführt, die den Takt clk als zweites Eingangssignal haben

Synchrone Latches - Synchrones D-Latch

- Das synchrone D-Latch (Delay-Latch) übernimmt während der positiven Taktphase ($clk=1$) das anliegende Eingangssignal d und gibt es direkt am Ausgang aus
- Während der negativen Taktphasen verharrt es im aktuellen Zustand



clk	d	q^{t+1}	
0	-	q^t	Speichern
1	0	0	Übernehmen
1	1	1	Übernehmen



- Das synchrone D-Latch lässt sich auch mit Hilfe eines synchronen RS-Latch aufbauen
- Der Eingang D wird direkt auf das Signal S gegeben und negiert auf das Signal R gegeben

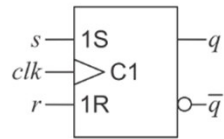
(Taktflankengesteuerte) Flipflops (FFs) - Motivation und Begriffsklärung

- Bei den synchronen Latches sind Zustandsänderungen innerhalb bestimmter Zeitintervalle möglich (z.B. positive Takt-Phase)
- Dadurch wird ein gewisser Synchronisationsgrad erreicht, der aber in der Praxis meist nicht ausreichend ist
- Wünschenswert ist, den potentiellen Zustandswechsel auf festdefinierte Zeitpunkte - die Taktflanken - zu begrenzen (Störsicherheit!)
- Das erledigen die (taktflankengesteuerten) Flipflops
- **Achtung !!!** In deutscher Literatur wird gerne der Begriff Flipflop als Oberbegriff für alle Arten von Kippschaltungen verwendet
- In der englischsprachigen Literatur wird der Begriff Latch für alles was nicht Taktflanken gesteuert ist verwendet und der Begriff Flipflop nur für taktflankengesteuerte Kippschaltungen
- Aufgrund der weiten Verbreitung wollen wir uns hier an die englischen Begriffe halten

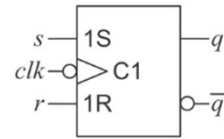
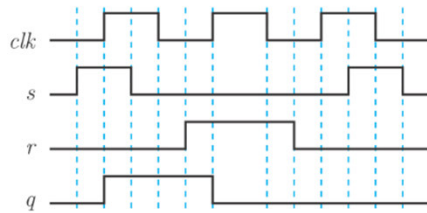
Taktflankengesteuerte Flipflops - generelle Arten

- Alle FF-Typen die im folgenden diskutiert werden, kommen in 3 Arten vor
 - Positiv flankengesteuert:
FF kann Zustand nur bei pos. Taktflanke (clk 0->1) ändern
 - Negativ flankengesteuert:
FF kann Zustand nur bei neg. Taktflanke (clk 1->0) ändern
 - Zweiflankengesteuert:
FF kann bei pos. Flanke Zustand intern wechseln, macht den internen Wechsel aber erst bei der nächsten neg. Flanke nach außen sichtbar

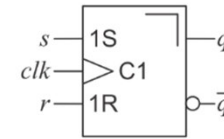
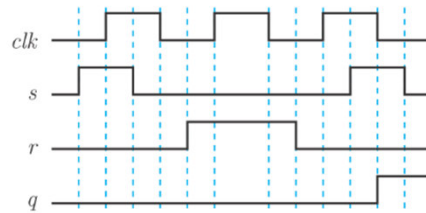
FFs - RS-Flipflop - Funktion



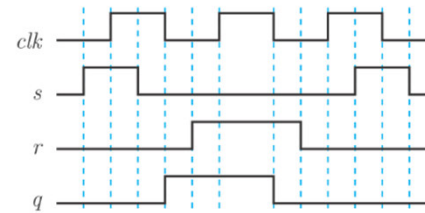
clk	s	r	q^{t+1}
0/1/↓	-	-	q^t
↑	0	0	q^t
↑	0	1	0
↑	1	0	1
↑	1	1	-



clk	s	r	q^{t+1}
0/1/↑	-	-	q^t
↓	0	0	q^t
↓	0	1	0
↓	1	0	1
↓	1	1	-

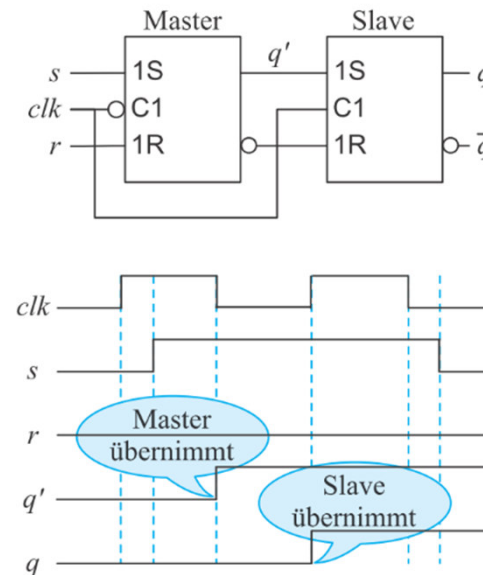


clk	s ↑	r ↑	q^{t+1}
0/1/↑	-	-	q^t
↓	0	0	q^t
↓	0	1	0
↓	1	0	1
↓	1	1	-



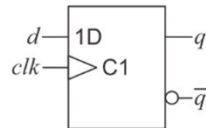
FFs - RS-Flipflop - Realisierung

- Man kann ein RS-FlipFlop realisieren, in dem man ein RS-Latch um ein weiteres RS-Latch ergänzt, das im Gegentakt arbeitet
- Da beide Latches hintereinandergeschaltet sind, blockiert während der pos. und neg. Taktphase immer eines der beiden Latches
- $S=0, R=0; clk=1 \rightarrow$ Master gesperrt; $S=1; clk=0 \rightarrow$ Master aktiv, $q'=S$; $clk=1 \rightarrow$ Slave aktiv $\rightarrow q=q'$
- Das zweiflankengesteuerte FFs erhält man, wenn man ein positiv flankengesteuertes und negativ flankengesteuertes FFs nach dem gleichen Prinzip zusammenschließt

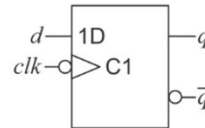
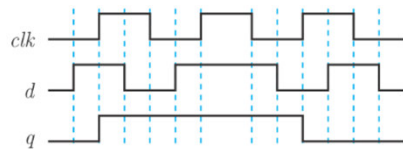


FFs - D-Flipflop

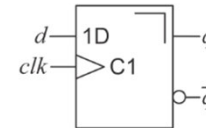
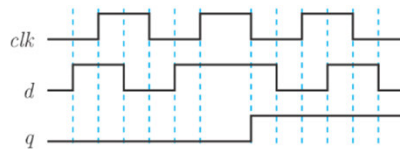
- Das D-FF speichert seinen aktuellen Zustand und führt einen Zustandswechsel immer synchron zu einer der Taktflanken aus.



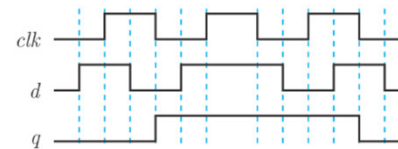
clk	d	q^{t+1}
0/1/↓	-	q^t
↑	0	0
↑	1	1



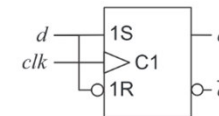
clk	d	q^{t+1}
0/1/↑	-	q^t
↓	0	0
↓	1	1



clk	d ↑	q^{t+1}
0/1/↑	-	q^t
↓	0	0
↓	1	1

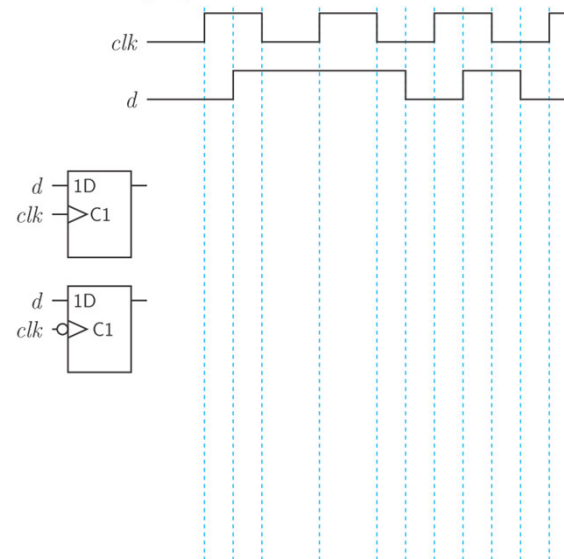


- D-FF kann auf ähnliche Weise wie das D-Latch aus einem RS-FF gebaut werden



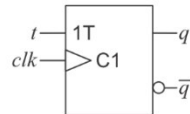
FFs - Übung

- Vervollständigen Sie den Ausgangssignalverlauf zu den gegebenen Eingangssignalen und benennen Sie den jeweiligen Latch-/FF-Typ mit seiner Art von Steuerung (async, sync [pos, neg] Taktzustand, sync [pos, neg, zwei] Taktflanke[n])

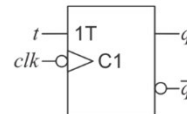
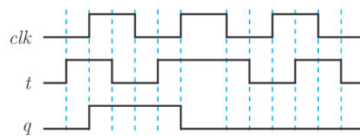


FFs - T-Flipflop

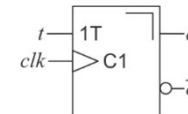
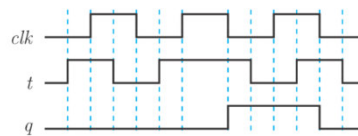
- Ähnlich wie das D-FF hat das T-FF nur einen Eingang
- Das T-FF hat seinen Namen von der „Toggle“-Funktion (engl. hin und her kippen)
- $t=1 \rightarrow q=\text{not } q$; $t=0 \rightarrow q=q$



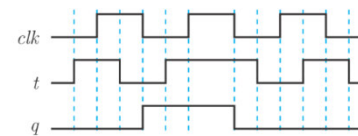
clk	t	q^{t+1}
0/1/↓	-	q^t
↑	0	q^t
↑	1	$\neg q^t$



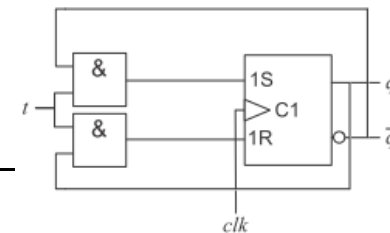
clk	t	q^{t+1}
0/1/↑	-	q^t
↓	0	q^t
↓	1	$\neg q^t$



clk	$t \uparrow$	q^{t+1}
0/1/↑	-	q^t
↓	0	q^t
↓	1	$\neg q^t$

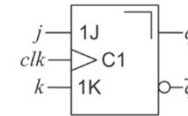
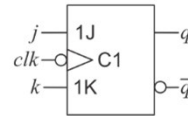
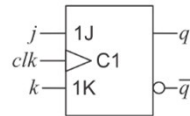


- Auch das T-FF kann durch das RS-FF realisiert werden:
 - !q wird über ein UND mit T als Eingang auf S zurückgeführt und q wird über ein UND mit T als Eingang auf R zurückgeführt



FFs - JK-Flipflop - Funktion

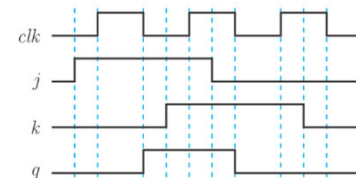
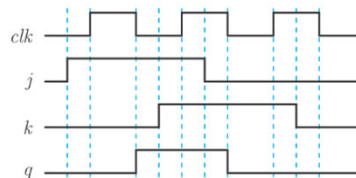
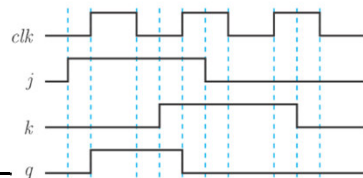
- Das JK-FF vereint die Funktion eines T-FF mit einem RS-FF
- Die Eingänge J (jump) und K (kill) haben die gleiche Bedeutung wie S und R
- Bei der für das RS-FF verbotene Kombination $SR=“11”$ wird das T-FF nachgeahmt und der Zustand invertiert



clk	j	k	q^{t+1}
0/1/↓	-	-	q^t
↑	0	0	q^t
↑	0	1	0
↑	1	0	1
↑	1	1	$\neg q^t$

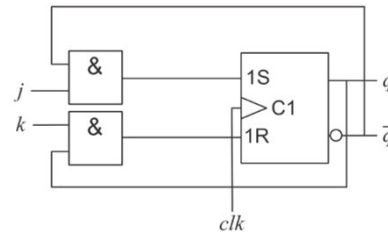
clk	j	k	q^{t+1}
0/1/↑	-	-	q^t
↓	0	0	q^t
↓	0	1	0
↓	1	0	1
↓	1	1	$\neg q^t$

clk	j ↑	k ↑	q^{t+1}
0/1	-	-	q^t
↓	0	0	q^t
↓	0	1	0
↓	1	0	1
↓	1	1	$\neg q^t$



FFs - JK-Flipflop - Realisierung

- Das JK-FF kann man aus dem RS-FF aufbauen, indem man die verbotene $SR=“1\ 1”$ Kombination durch Vorschalten zweier UND abfängt
- $\neg q$ wird über ein UND mit J als Eingang auf S zurückgeführt und q wird über ein UND mit K als Eingang auf R zurückgeführt



FFs- Bevorrechtigte Eingänge

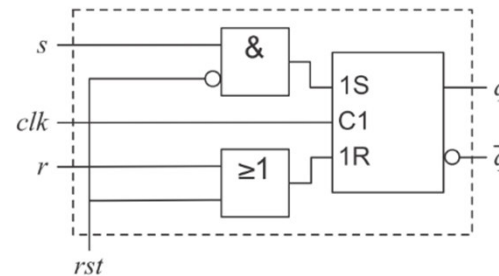
- In der Praxis haben viele Schaltelemente neben den normalen Eingängen für den Takt und Daten sog. Bevorrechtigte Eingänge
 - Diese Eingänge haben Vorrang vor den normalen Eingangssignalen
 - Eine Anwendung ist z.B. eine reset-Funktion

 - Es lassen sich zwei Arten unterscheiden
 - Synchrone bevorrechtigte Eingänge:
werden ebenso über das gleiche Taktsignal gesteuert
 - Asynchrone bevorrechtigte Eingänge:
wirken unmittelbar
-

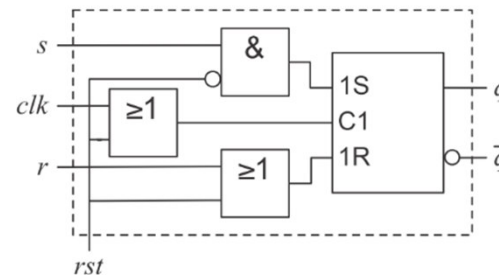
FFs- Bevorrechtigte Eingänge - synchroner und asynchroner Reset

- Bei einem synchronen Reset werden die Eingänge des RS-FFs auf $SR=“01”$ gezwungen, sobald $rst=1$
- Bei einem asynchronen Reset ist clk zusätzlich mit rst verbunden, um eine künstliche Taktphase zu erzeugen, solange $rst=1$ ist

■ Synchroner Reset-Eingang

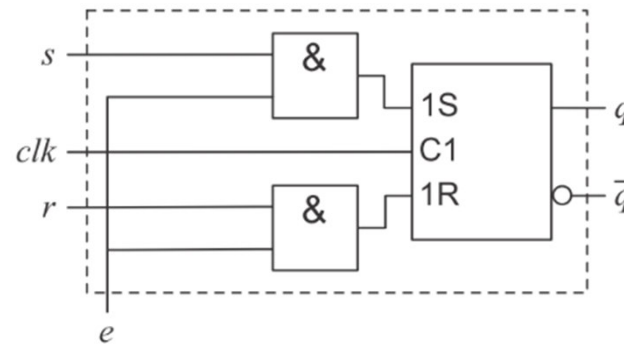


■ Asynchroner Reset-Eingang



FFs- Bevorrechtigte Eingänge - synchroner Enable

- Vor S wird ein UND-Gatter mit S und „e“ als Eingang geschaltet
- Vor R wird ein UND-Gatter mit R und „e“ als Eingang geschaltet
- Wenn „e“=0 -> egal was S oder R, das RS-FF befindet sich im Speicher-Modus (UND liefert nur 1, wenn beide Eingänge 1 sind!)



FFs - Parallelschaltung von FFs (Register)

- Die Parallelschaltung von Flipflops heißt Register
- Typische weitere Eigenschaften sind:
 - gleiche Taktleitung für alle Elemente
 - bevorrechtigte Eingänge, z.B. Set / Reset nach außen geführt
- Register werden als Speicher innerhalb des Prozessors verwendet

