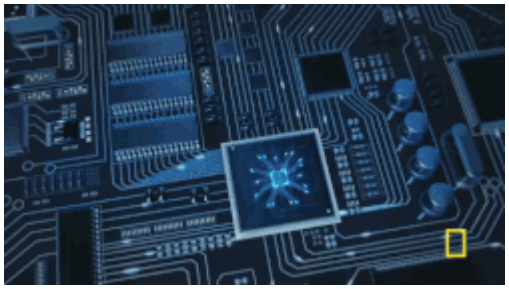
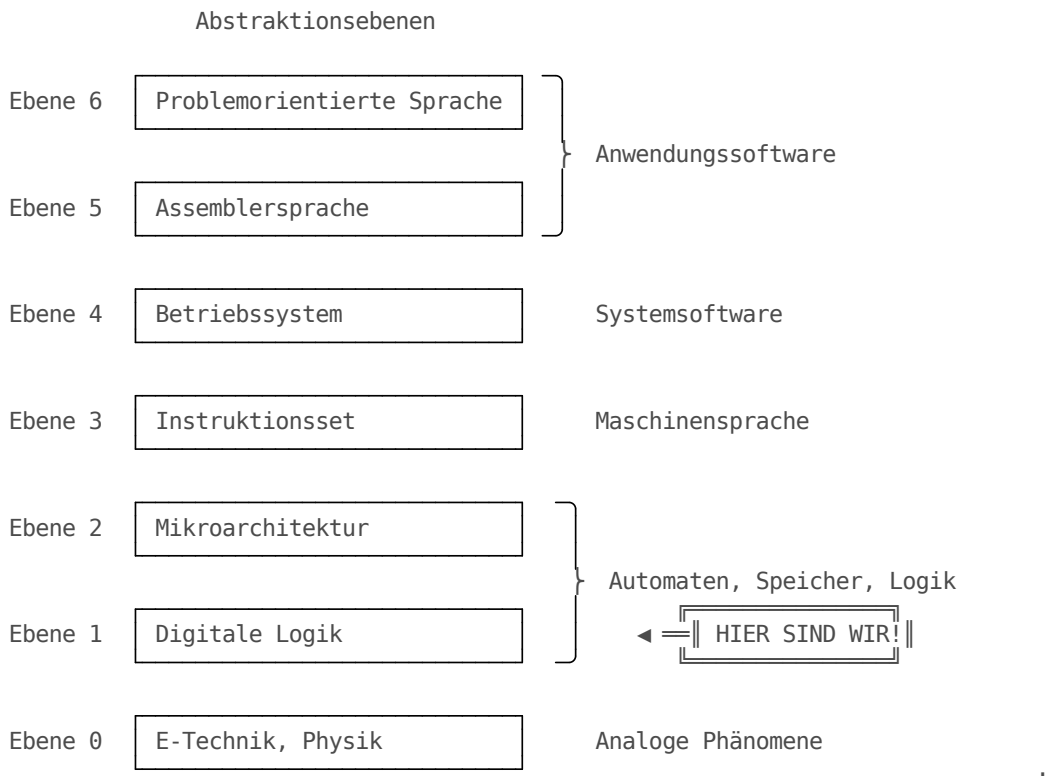
FlipFlops

Parameter	Kursinformationen
Veranstaltung:	`Digitale Systeme / Eingebettete Systeme`
Semester:	`Wintersemester 2025/26`
Hochschule:	`Technische Universität Freiberg`
Inhalte:	`FlipFlop-Schaltungen, sequenzielle Logik, SR-, D-, JK- und T-FlipFlops`
Link auf GitHub:	https://github.com/TUBAF-IfI-LiaScript/VL_EingebetteteSysteme/blob/master/06_FlipFlops.md
Autoren:	Sebastian Zug & André Dietrich & Fabian Bär



Fragen an die Veranstaltung

- Erläutern Sie die Notwendigkeit von Speicherelementen für die Umsetzung eines Rechners.
- Worin besteht der Unterschied zwischen Schaltfunktionen/Schaltnetzen und Schaltwerken?
- Beschreiben Sie die Eingangsbelegungen und korrespondierenden Zustandsänderungen am RS-Flip-Flop.
- Welche anderen Flip-Flop-Typen kennen sie, wo liegen die Vorteile gegenüber dem RS-Flip-Flop?
- Erläutern Sie die Begriffe „zustandsgesteuert“ und „flankengesteuert“.
- Welche Kernelemente hat ein Schaltwerk? Worin unterscheiden sich die Varianten von Mealy und Moore?
- Welches Vorgehen ist für die Umsetzung eines Schaltwerkes notwendig?
- An welcher Stelle ist die invertierte Wahrheitstabelle eines Flip-Flops wichtig?



Einführung sequenzieller Logik

Kombinatorische Logik wird durch Schaltnetze repräsentiert, die durch zyklenfreie Graphen dargestellt werden können, sprich es gibt keine Eigenschaften:

1. Zustandslos: Ausgabe ist nur von der Eingabe abhängig;
2. One-Way: keine Rückkopplungen im Schaltnetz;
3. 0-Verzögerung: keine Berücksichtigung der Gatterlaufzeit.

Frage: Können wir die bisherigen Konzepte und Techniken der logischen Schaltungen einsetzen, um wesentliche Elemente eines Rechners zu beschreiben wie z.B. :

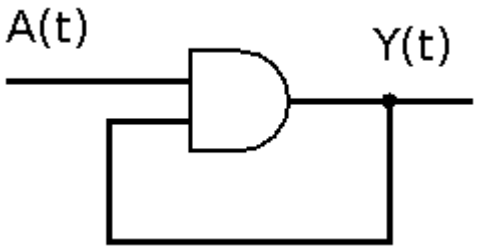
- Ablaufsteuerungen
- Speicherelemente
- Takterzeuger

> Merke: Es fehlt ein Speicher!

Was geschieht in einer digitalen Schaltung bei der Rückkopplung eines Gatterausganges?

Erster Ansatz

Verwendung eines rückgekoppelten AND-Bausteins.



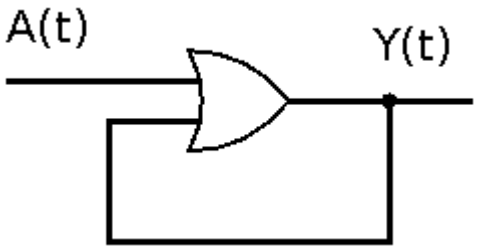
$A(t)$	$Y(t)$	$Y(t+\Delta t)$	$Y(t+2\Delta t)$	$Y(t+3\Delta t)$
1	1	1	1	1
0	1	0	0	0
1	0	0	0	0
0	0	0	0	0

$A(t)$	$Y(t+\Delta t)$	$Y(t+2\Delta t)$	$Y(t+3\Delta t)$
1	$Y(t)$	$Y(t)$	$Y(t)$
0	0	0	0

Offenbar behalten wir bei der Eingabe von 1 den aktuellen Zustand bei. Aus der Eingabe einer 0 folgt das "Löschen" des Zustandes.

Zweiter Ansatz

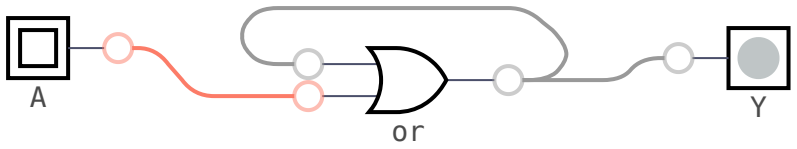
Verwendung eines rückgekoppelten OR-Bausteins.



$A(t)$	$Y(t)$	$Y(t+\Delta t)$	$Y(t+2\Delta t)$	$Y(t+3\Delta t)$
1	1	1	1	1
0	1	1	1	1
1	0	1	1	1
0	0	0	0	0

$A(t)$	$Y(t+\Delta t)$	$Y(t+2\Delta t)$	$Y(t+3\Delta t)$
1	1	1	1
0	$Y(t)$	$Y(t)$	$Y(t)$

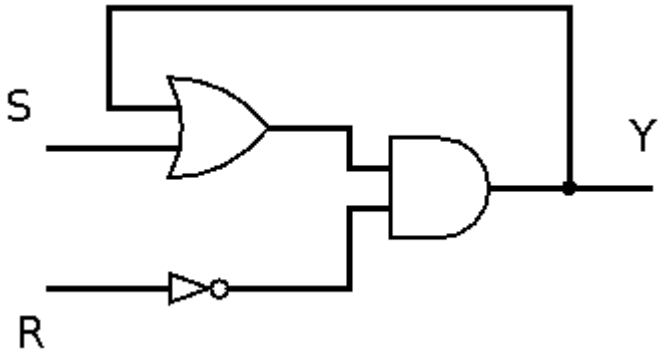
...



Dritter Ansatz (SR-Latch)

Gesteuerte Rückkopplung über OR-Baustein.

Wir fügen eine weitere Eingangsgröße hinzu, die die Logik des Speichers auf der Basis des rückgekoppelten OR vervollständigt. Durch die Einbettung des AND-Gatters können wir die Rückkopplung kontrollieren! Die Rückkopplung einer 1 erfolgt nur dann, wenn unser `R` Eingang nicht selbst eine 1 abbildet. Sobald wir an S eine 1 anlegen, wird diese auf Y sichtbar und gespeichert.

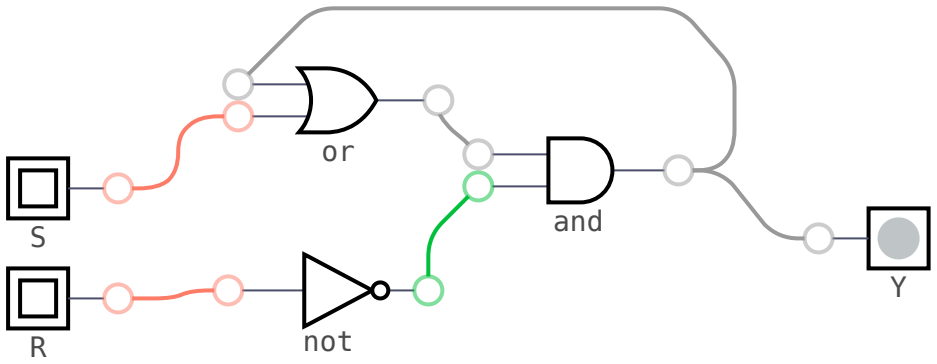


$R(t)$	$S(t)$	$Y(t)$	$Y(t+2\Delta t)$	**Bedeutung**
0	0	0	0	Speichern
0	0	1	1	Speichern
0	1	0	1	Setzen
0	1	1	1	Setzen*
1	0	0	0	Rücksetzen *
1	0	1	0	Rücksetzen
1	1	0	0	Vermeiden*
1	1	1	0	Vermeiden

Die mit * markierten Einträge werden jeweils aus dem benachbarten Zustand erreicht. Wenn wir zum Beispiel von einem Zustand $Y = 0$ beim Setzen ausgehen, generieren wir mit $S == 1$ ein $Y(t + 2\Delta t)$ einen 1-Pegel. Dies entspricht der Zeile 4 der Wahrheitstabelle.

Der Zeitversatz von $2\Delta t$ ergibt sich aus dem Schaltverhalten der beiden Bauteile. In obigem Schaubild sind noch drei Typen (OR, AND und NOT), im folgenden wird diese Konfiguration auf NAND- und NOR-Gatter vereinheitlicht. Entsprechend kann dann von einem einheitlichen Δt ausgegangen werden.

Werden beide Eingänge auf 1-Pegel gesetzt, führen beide Ausgänge 1-Pegel. Dieser Zustand kann nicht gespeichert werden. In der Literatur wird dieser Zustand als "unbestimmt" oder "verboten" bezeichnet. Doch die Unbestimmtheit tritt nur in dem Fall ein, wenn beide Eingänge nach diesem Zustand gleichzeitig 0-Pegel erhalten. Dieser Folgezustand ist "unbestimmt", weil nicht klar ist, welcher Ausgang 1-Pegel führt.



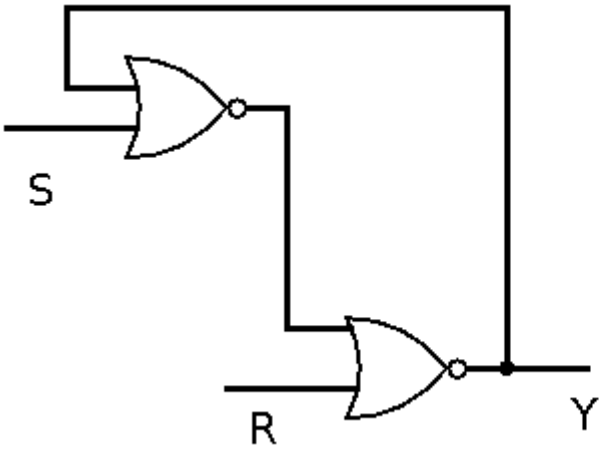
Daraus folgt die Wertetabelle eines RS-Flip-Flops. Anhand dieser Darstellung wird klar, dass die Eingangsgrößen R und S ihre Kürzel aus gutem Grund tragen: `Reset` und `Set`.

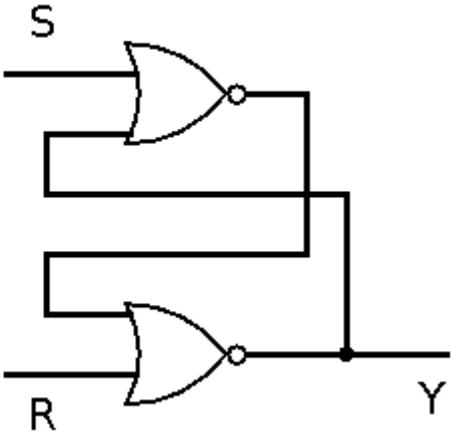
$R(t)$	$S(t)$	$Y'(t) = Y(t + 2\Delta t)$
0	0	$Y(t)$
0	1	1
1	0	0
1	1	Vermeiden

Nun können wir mit dem Theorem von de Morgan einige Anpassungen vornehmen:

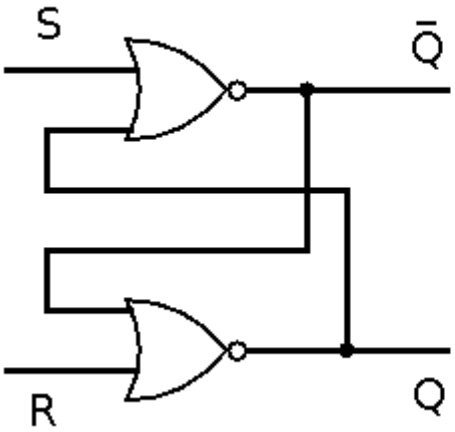
$$\begin{aligned} Y' &= (Y + S) \cdot \overline{R} \\ &= \overline{\overline{Y + S} + R} \end{aligned}$$

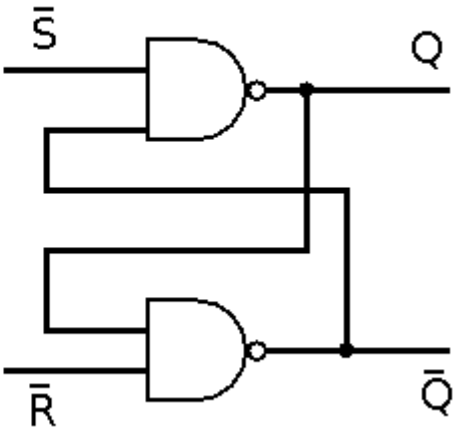
Damit sind wir in der Lage, unseren bistabilen Speicher mit zwei Gattern gleichen Typs zu realisieren.

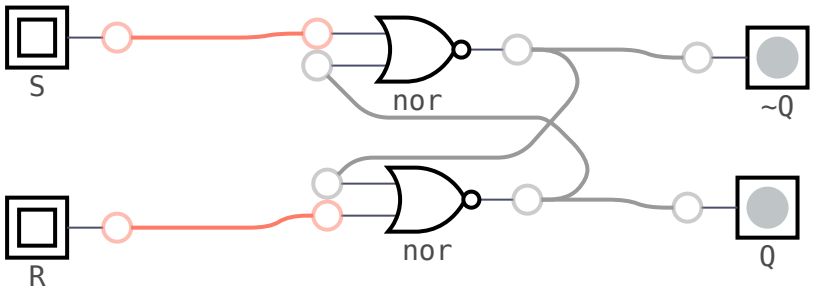


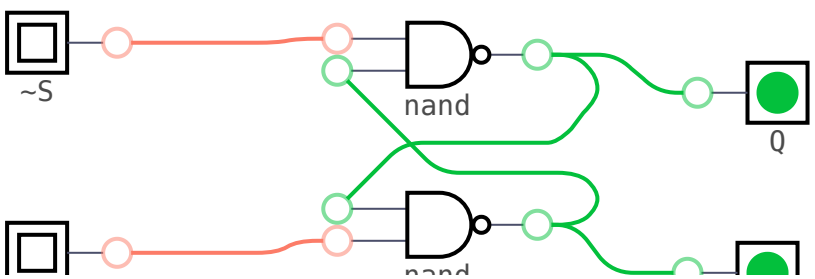


Die Darstellung des Flip-Flops kann auch mit Hilfe von NAND-Bausteinen erfolgen. Beachten Sie dabei die entsprechende Anpassung der Eingangsgrößen.

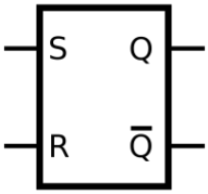








R-S Flip-Flops werden in einem eigenen Symbol abstrahiert. Dabei wird unser Ausgang Y nun mit Q bezeichnet.



D-Latch

Der D-Latch macht sich die Tatsache zunutze, dass in den beiden aktiven Eingangskombinationen (01 und 10) eines gattergesteuerten SR-Latch R das Komplement von S ist.

RS-Latch

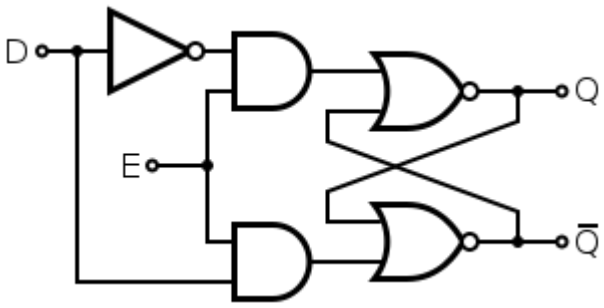
$R(t)$	$S(t)$	$Y'(t) = Y(t + 2\Delta t)$
0	0	$Y(t)$
0	1	1
1	0	0
1	1	nicht erlaubt

D-Latch

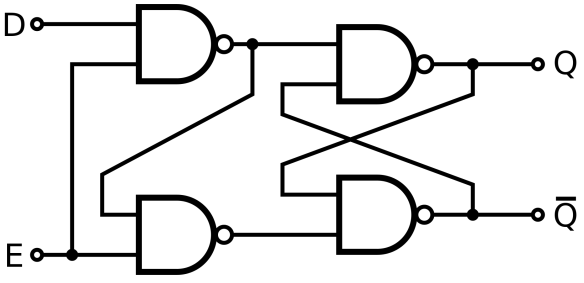
$E(t)$	$D(t)$	Y Y $2.$
0	0	Y
0	1	Y
1	0	0
1	1	1

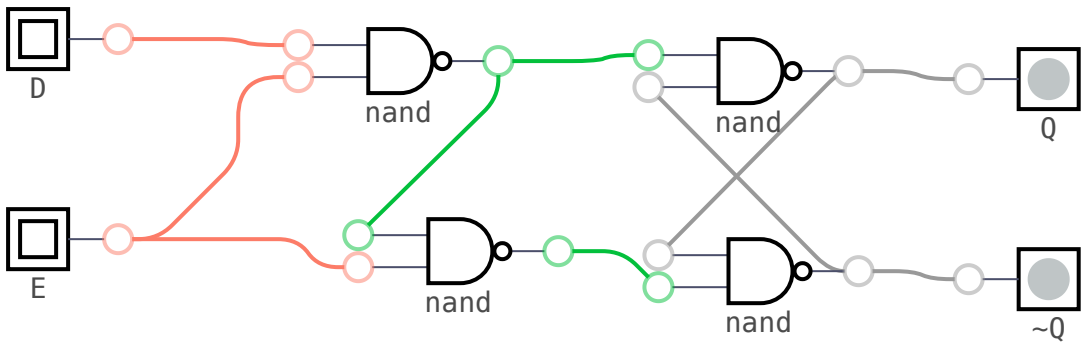
Die Eingangs-NAND-Stufe wandelt die beiden D-Eingangszustände (0 und 1) durch Invertieren des Dateneingangssignals in diese beiden Eingangskombinationen für den nächsten SR-Latch um.

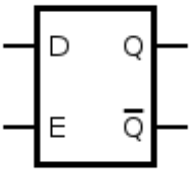
D-Latch auf der Basis eines NOR-RS-Gatters



D-Latch auf der Basis eines NAND-RS-Gatters



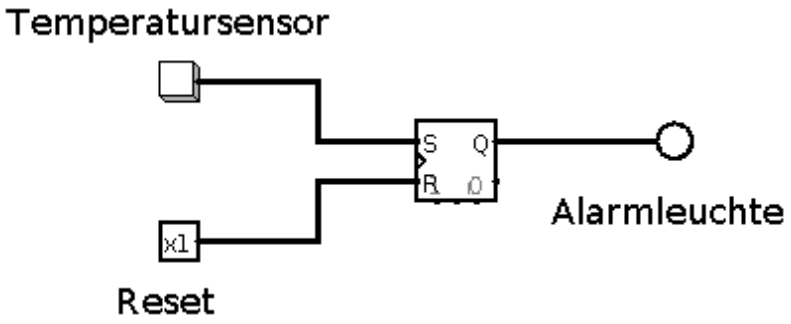




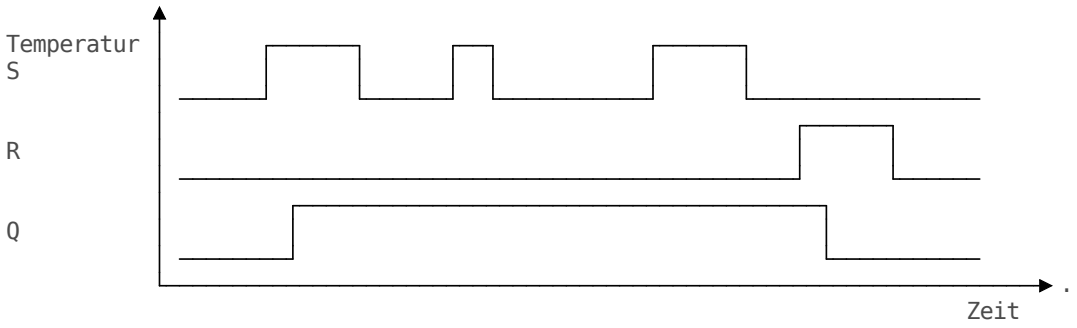
Der D-Latch kann auch als synchrones Flip-Flop verstanden werden. In diesem Fall würde `E` als Takt betrachtet werden.

Anwendungsbeispiele

Anwendungsbeispiel 1

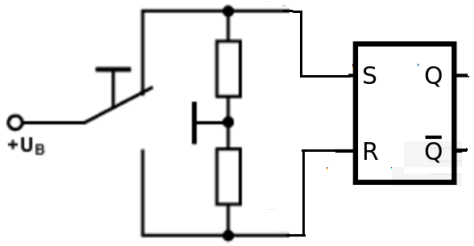


Der Temperatursensor hat einen digitalen Ausgang, der mit dem Überschreiten einen High-Pegel annimmt. Dieser wird in unserem Flip-Flop gespeichert `Q`, bis der Zustand mit `R` wieder auf einen Low-Pegel geführt wird.



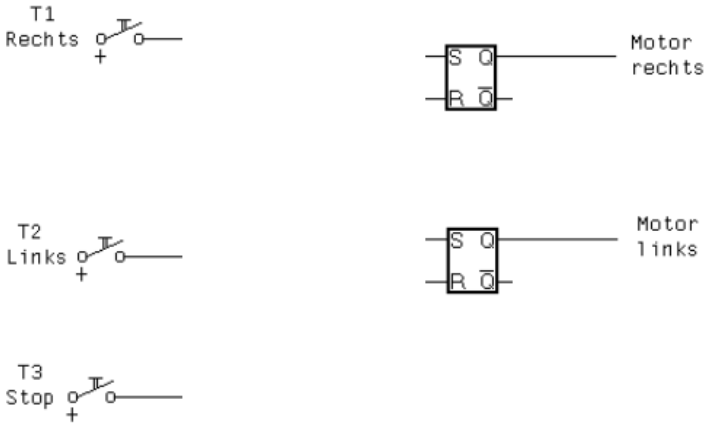
Anwendungsbeispiel 2

Ein Taster schaltet bei der Aktivierung nicht zu einem definierten Zeitpunkt, sondern ist durch eine Übergangsphase gekennzeichnet.



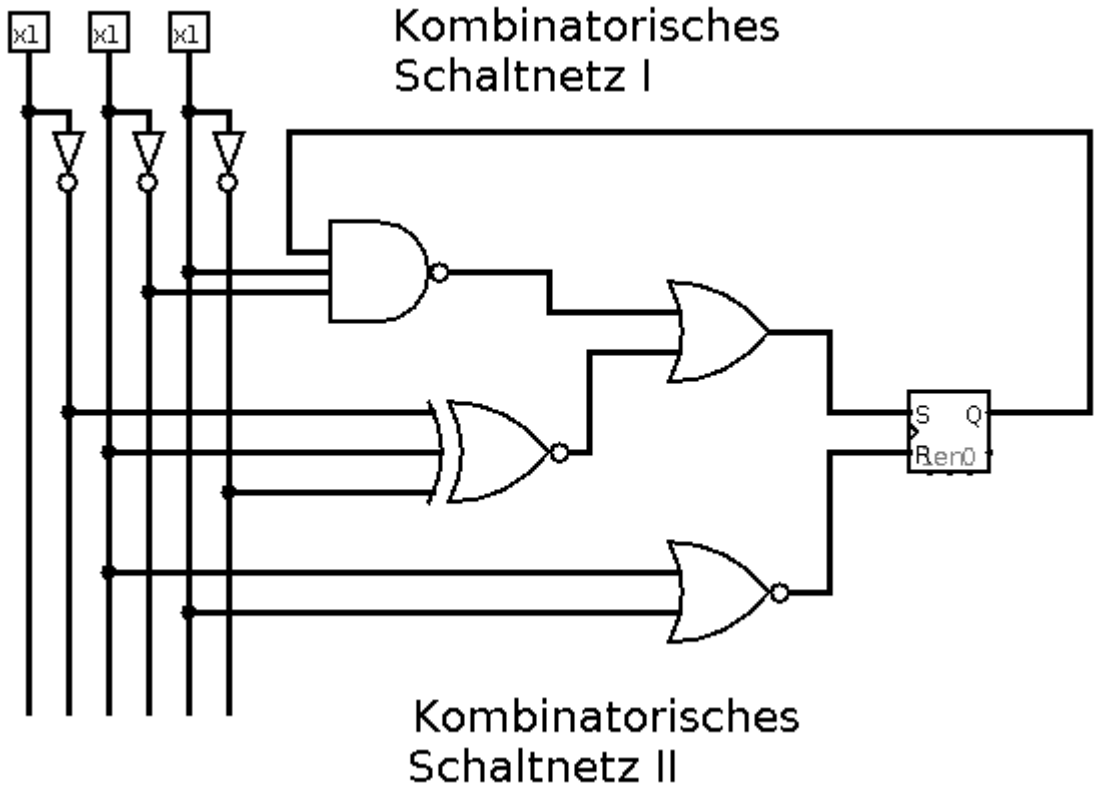
Anwendungsbeispiel 3

Wir wollen eine Schaltung entwerfen, die für einen Motor den abrupten Wechsel von einem Rechts- auf einen Links-Lauf vermeidet. Wie könnte dies umgesetzt werden?



Der Motor soll nur aus einem Stop-Mode heraus in den Links-Lauf oder den Rechts-Lauf starten.

Taktung von Schaltwerken



Wann sind die Werte am Ausgang Q gültig? Die Laufzeit der Schaltnetze bestimmt das Verhalten des Speicherelements!

> Zielstellung: Spezifische Zeiträume zur Übernahme von Eingabe und Ausgabewerten

asynchrone Schaltwerke

- gesteuert durch Veränderung der Eingangssignale
- Zeitpunkt, an dem wieder stabile Ausgangssignale vorliegen, ist nur durch Gatterlaufzeit festgelegt
- aufwendiger Entwurf (Zeit ist „Echtzeit“)
- sehr schnelle Schaltwerke möglich

synchrone Schaltwerke

- gesteuert durch zentralen Takt
- Übernahme der Änderung eines Eingangssignals erfolgt nur zu festen Zeiten (Zeiträumen oder Zeitpunkten)
- einfacher, systematischer Entwurf (Zeit ist „Taktzeit“)
- langsamste Komponente bestimmt maximale Taktfrequenz

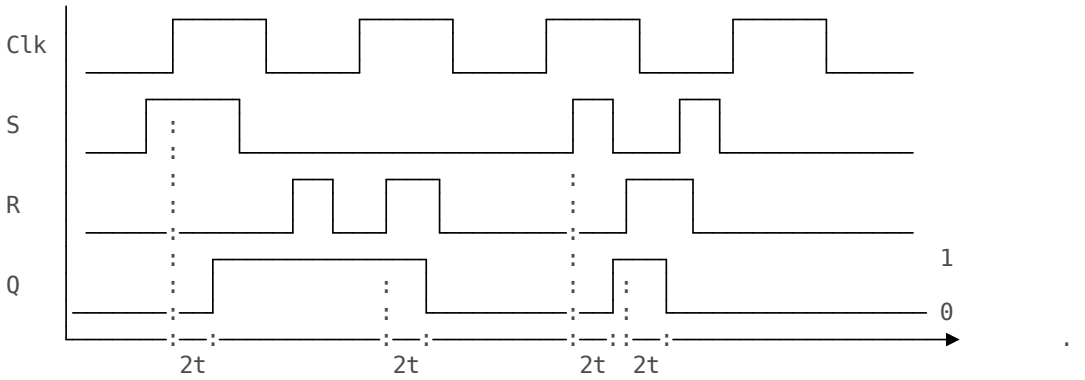
Synchrones SR-Latch

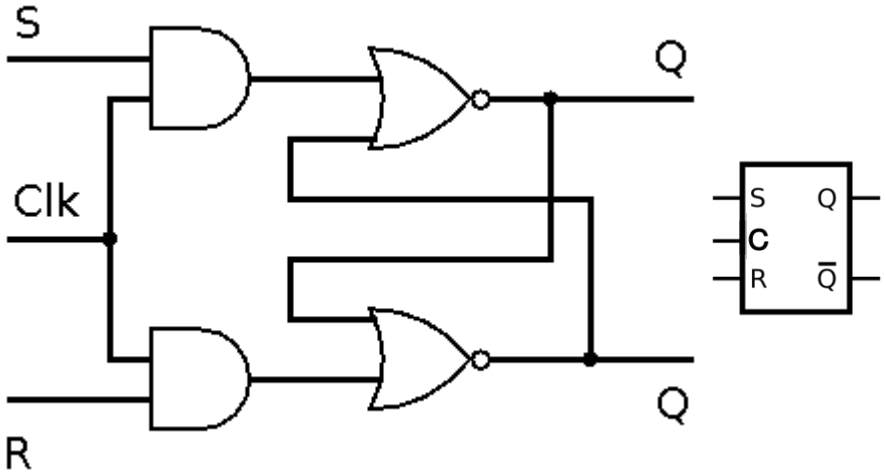
Für die Realisierung eines synchronisierten Zustandswechsels wird der Eingang unseres Flip-Flops um einen Takt `Clk` erweitert. Die Signale an R und S werden nur übernommen, wenn Taktsignal Clk aktiv ist:

- bei Clk = 0 sind R und S irrelevant (d = „don't care“)
- bei Clk = 1 stellt sich der neue Folgezustand Q´ ein

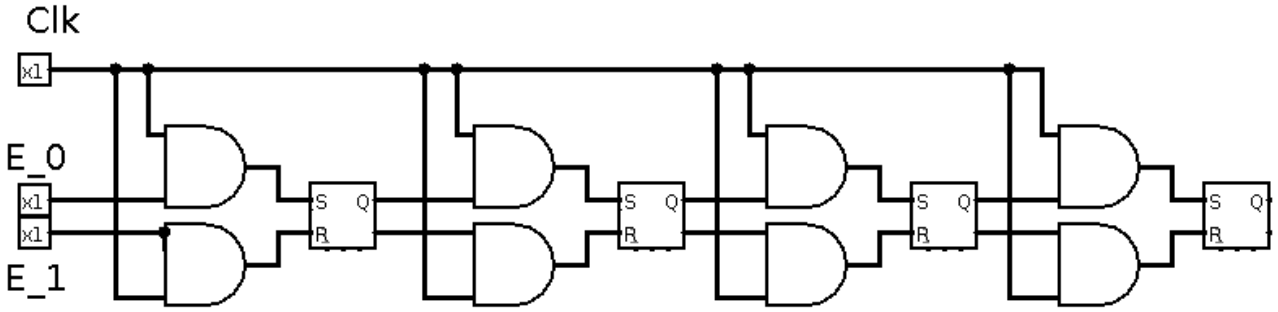
$R(t)$	$S(t)$	$Clk(t)$	$Q'(t)$
d	d	0	Q
0	0	1	Q
0	1	1	1
1	0	1	0
1	1	1	nicht erlaubt

Beachten Sie, dass sich mit dem d-Zustand die Wertetabelle deutlich verkürzt.





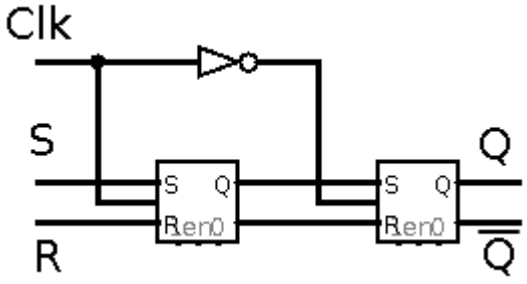
Merke: Durch die Möglichkeit mehrerer Zustandsänderungen in einer Taktphase ist das zustandsgesteuerte RS-Flip-Flop für viele Anwendungen ungeeignet. Welche Idee hat die nachfolgende Schaltung und warum wird sie nicht funktionieren?



Gewünscht: Flip-Flop-Variante, die Änderungen nur zu einem definierten Zeitpunkt zulässt

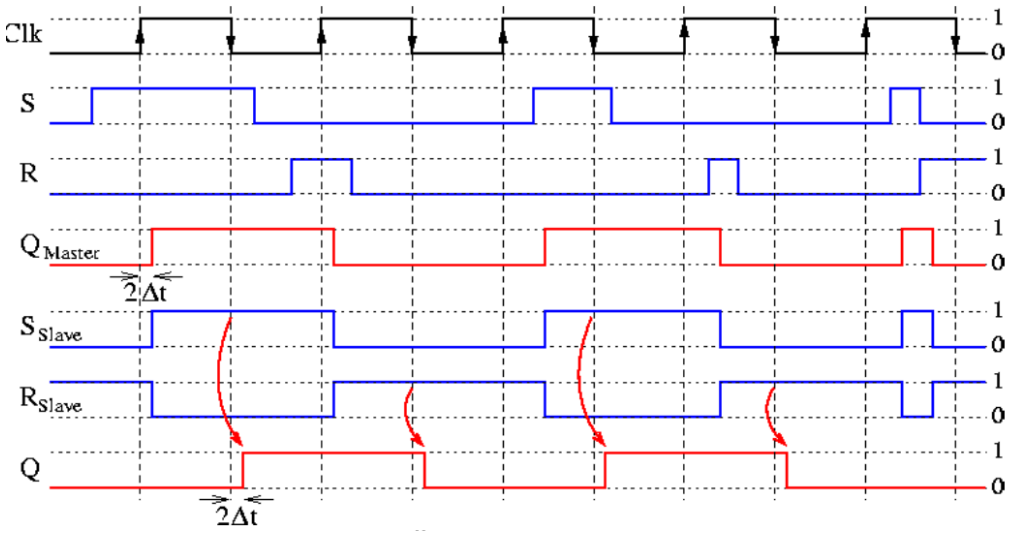
Master-Slave Flip-Flops

Ein Master-Slave RS-Flip-Flop besteht aus 2 hintereinander-geschalteten zustandsgesteuerten RS-Flip-Flops (als „Master“ und als „Slave“ bezeichnet); zusätzlicher Inverter negiert Taktsignal für „Slave“.



Ablauf einer Periode:

1. „Master“ übernimmt Eingangswerte bei Clk = 1 („Slave“ ändert sich nicht)
2. „Slave“ übernimmt Werte vom „Master“ bei Clk = 0 („Master“ ändert sich nicht)

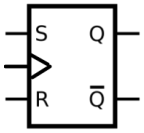


Frage: Welche maximale Verzögerung für ein eingehendes Signal ergibt sich also?

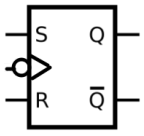
Flankengesteuerte Flip-Flops

Durch eine spezielle Schaltungstechnik kann erreicht werden, dass auch die Eingangsleitungen nur

- bei steigender Flanke (positive Flanke),

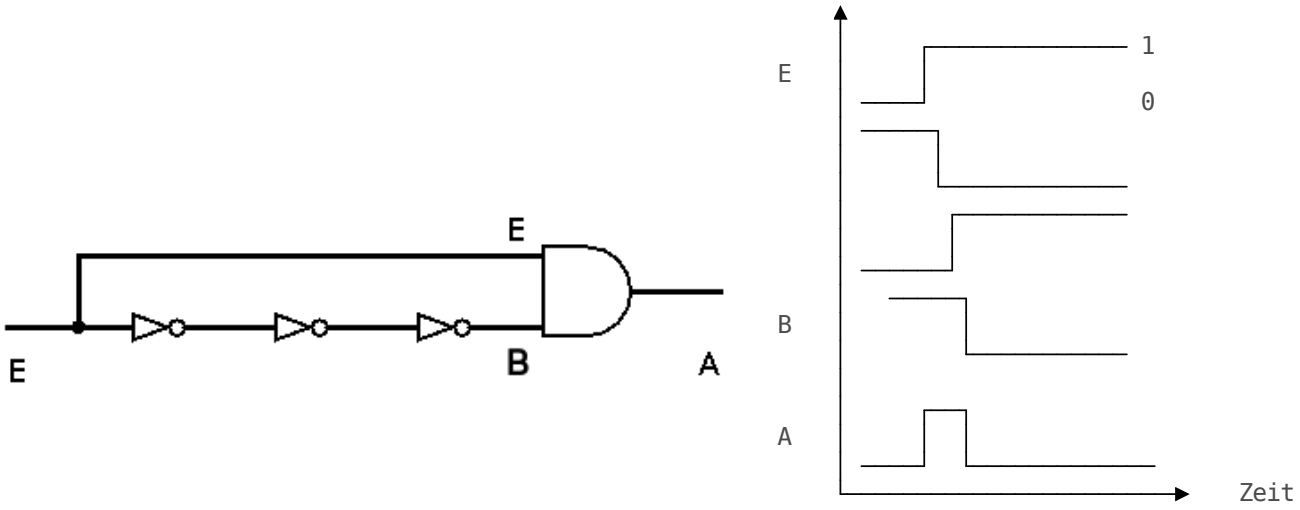


- bei fallender Flanke (negative Flanke),



- in beiden Fällen berücksichtigt werden!

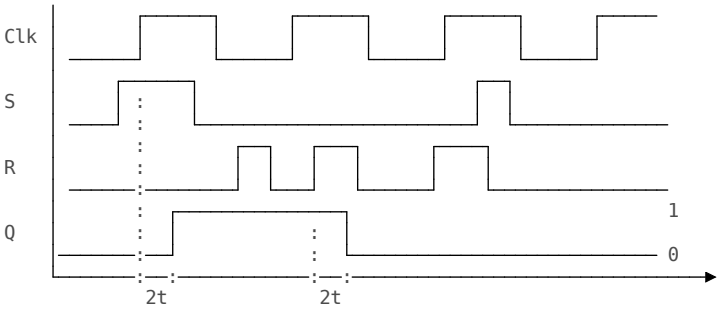
Wie lässt sich ein Flankendetektor umsetzen? Wir machen uns das Laufzeitverhalten unserer Bauteile zu Nutze.



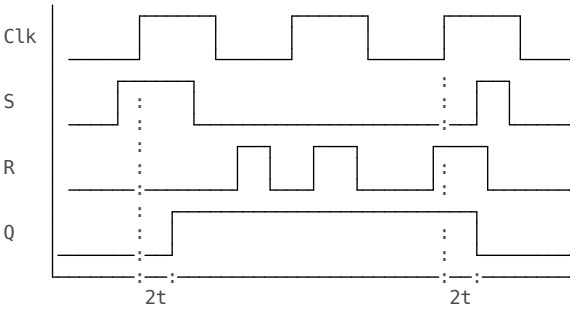
Die Signallaufzeiten sind für die ansteigende und fallende Signalflanke unterschiedlich und variieren mit den Schaltkreisfamilien (3 - 30ns).

Welches Verhalten ergibt sich für ein (positiv) flankengetriggertes RS-FlipFlop daraus?

Zustandsgetrieben



Positiv Flankengetrieben

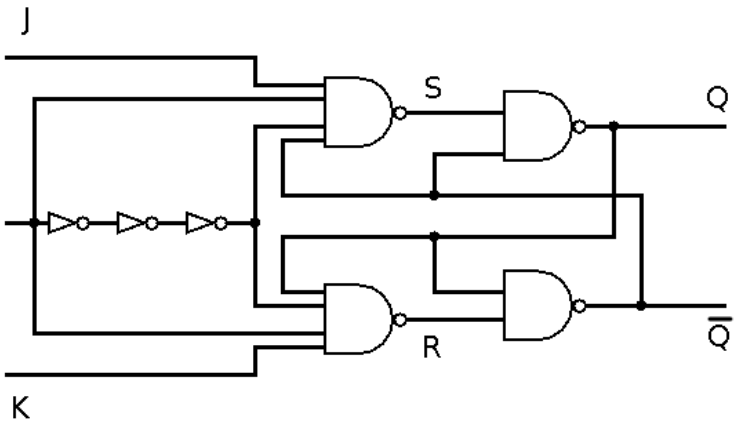


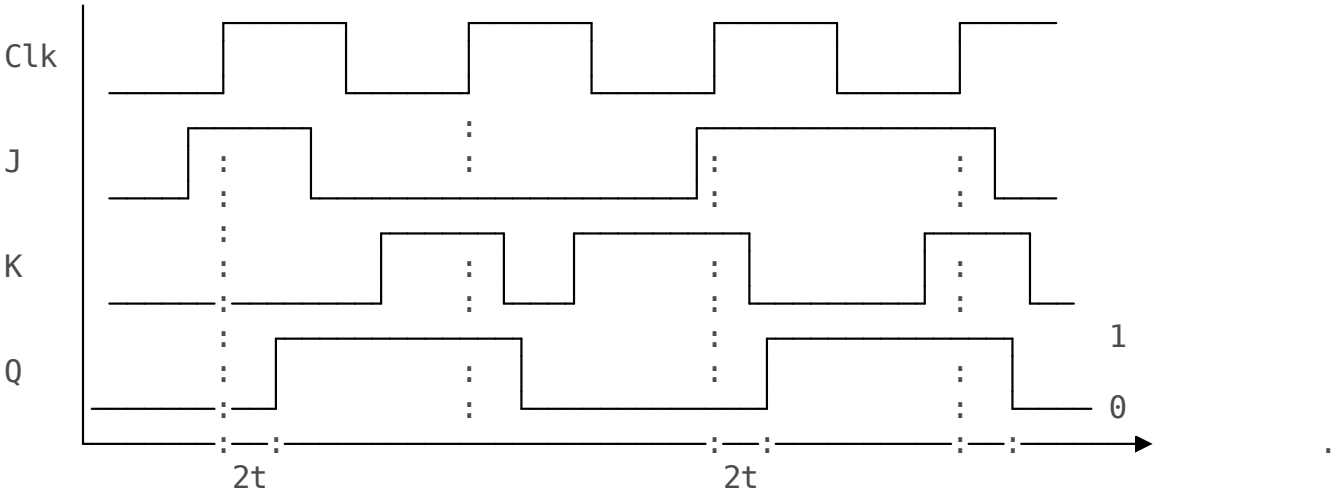
Weitere Flip-Flop-Konfigurationen

JK-Flip-Flop

JK-Flipflops wurden wahrscheinlich nach Jack Kilby benannt. Sie basieren auf dem asynchronen RS-Flipflop, sind aber flankengesteuert oder als Master-Slave-Flipflop ausgeführt. Mit dem Taktsignal und der Eingangsbelegung J = 1 und K = 0 wird am Ausgang eine 1 erzeugt und gespeichert, alternativ bei K = 1 und J = 0 eine 0. Der Zustand J = K = 1 ist erlaubt; in diesem Fall wechselt der Ausgangspegel mit jeder wirksamen Flanke des Taktsignals. Dieses Verhalten lässt die Bezeichnung als Toggle-Flipflop zu.

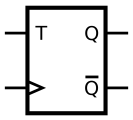
$J(t)$	$K(t)$	$Q'(t)$ **bei Flankenwechsel**
0	0	Q
0	1	0
1	0	1
1	1	$\overline{Q}$





T-Flip-Flop

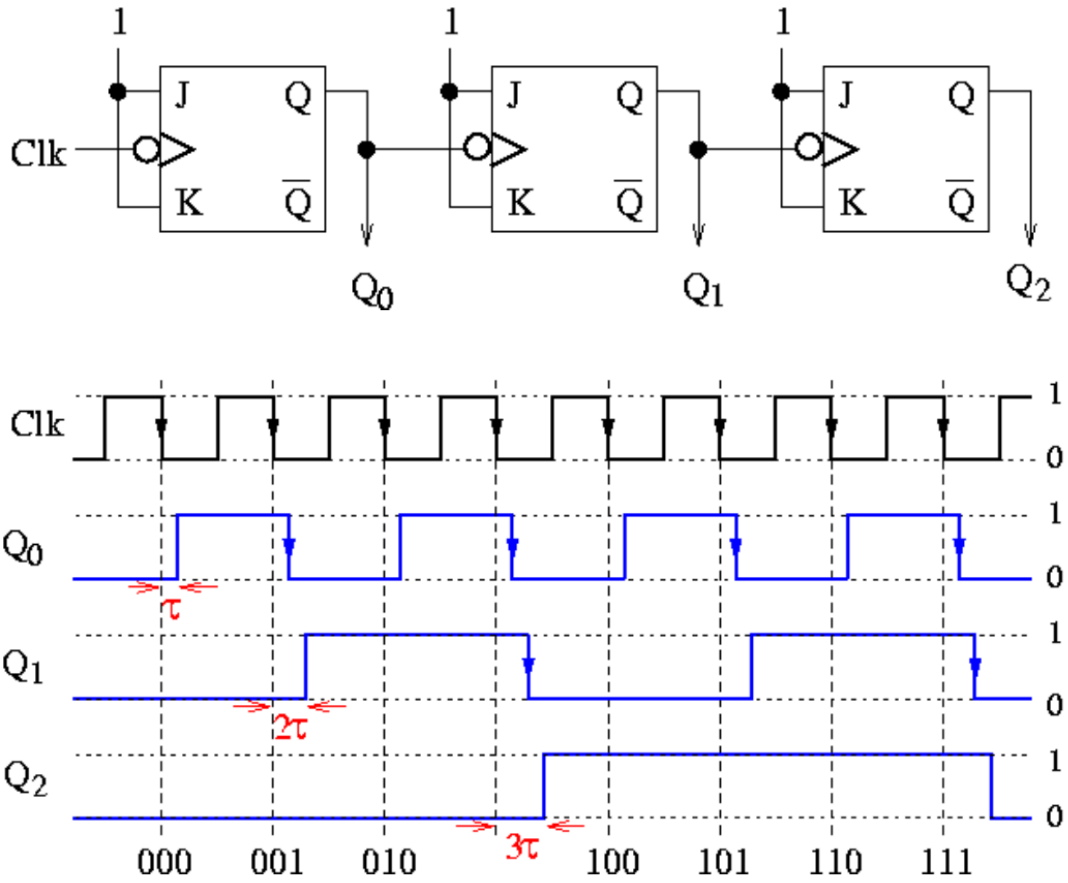
Das synchrone T-Flipflop besitzt neben dem Clk-Takteingang einen T-Eingang. T steht dabei für toggle – hin- und herschalten. Es zeigt ein Wechselverhalten synchron zur aktiven Flanke immer dann und nur dann, wenn T = 1 ist. Es kann aus einem flankengesteuerten JK-Flipflop gebildet werden, indem J- und K-Eingang verbunden werden und gemeinsam als T-Eingang fungieren.



$T(t)$	$Q'(t)$ **bei Flankenwechsel**
0	Q
1	$\overline{Q}$

Anwendungen

Asynchroner 3-Bit-Zähler / Frequenzteiler



Beachten Sie die Verzögerung der flanken-getriggerten JK-Flipflops - diese bestimmt die maximale Taktfrequenz.

4 Bit-Speicher Array

DM74LS194A 4-Bit Bidirectional Universal Shift Register

http://www.datasheet-pdf.com/PDF/DM74LS194A-Datasheet-FairchildSemiconductor-518503

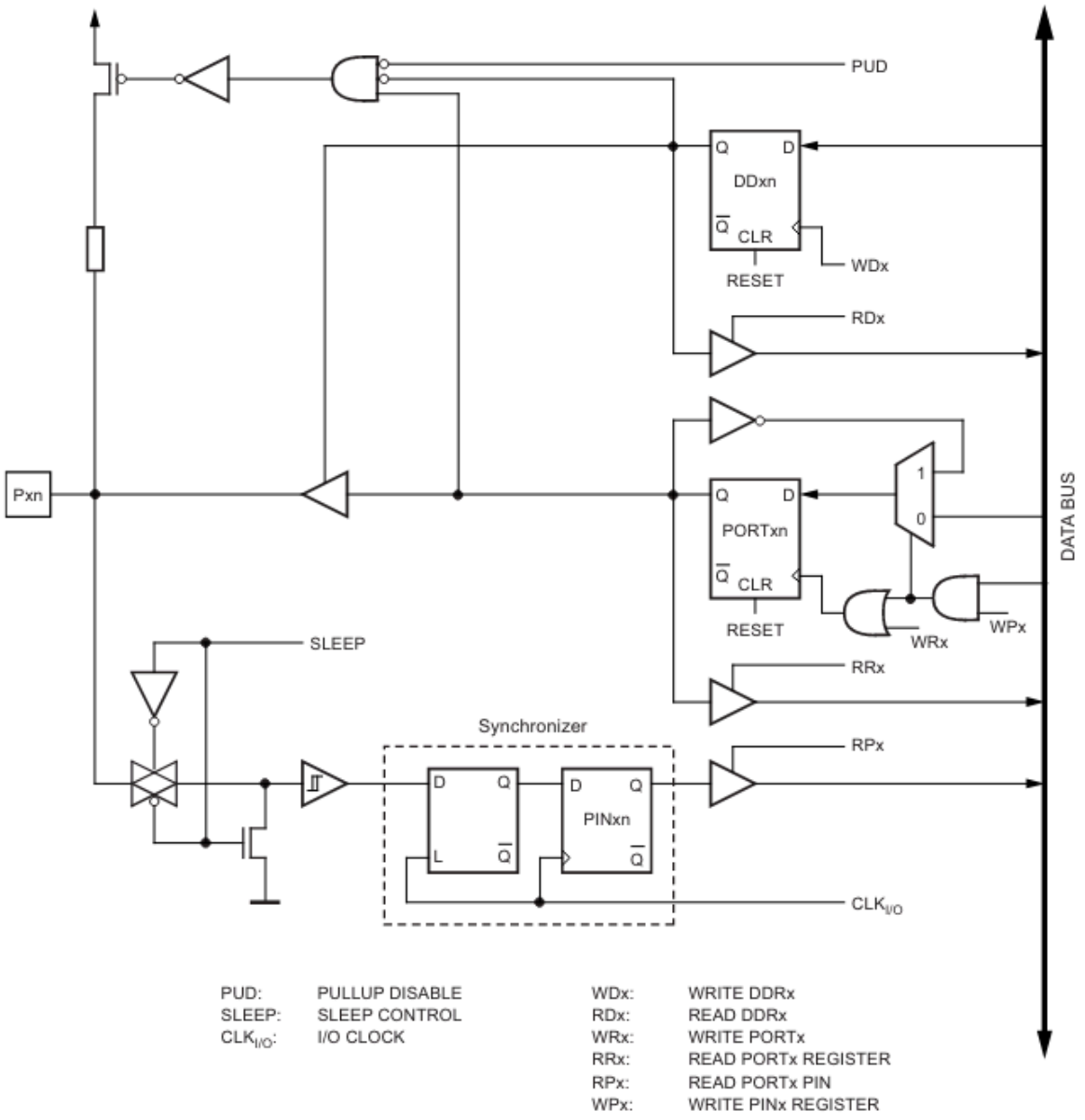
Zusammenfassung

Parameter	Zustandsgesteuertes FF (Latch)	Flankengesteuertes FF
Eingangssignal	`Enable`	`Clock`
Trigger	Pegelgetrieben	Flankengesteuert
Reaktionszeit `R`	kleiner oder gleich dem Takt	entspricht Implementierung der Periodendauer
Komplexität	einfach	aufwändigere Flankenerkennung
Robustheit gegenüber Rauschen	kann Probleme generieren	
Anwendungen	Speichern einzelner Bytes	Register, Counter, Frequenzteiler Register

Einsatzbeispiel

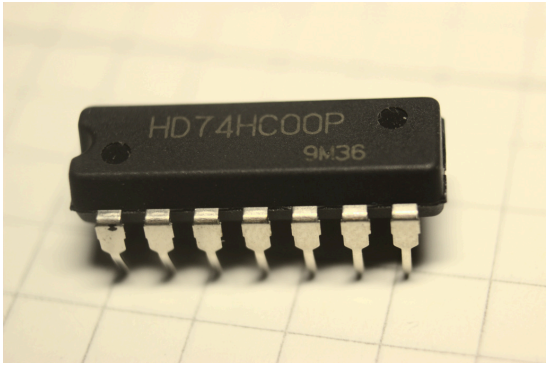
Konfiguration eines einzelnen IO-Pins eines AVR-Mikrocontrollers mit Hilfe von 4 Flip-Flops.

Figure 13-2. General Digital I/O(1)

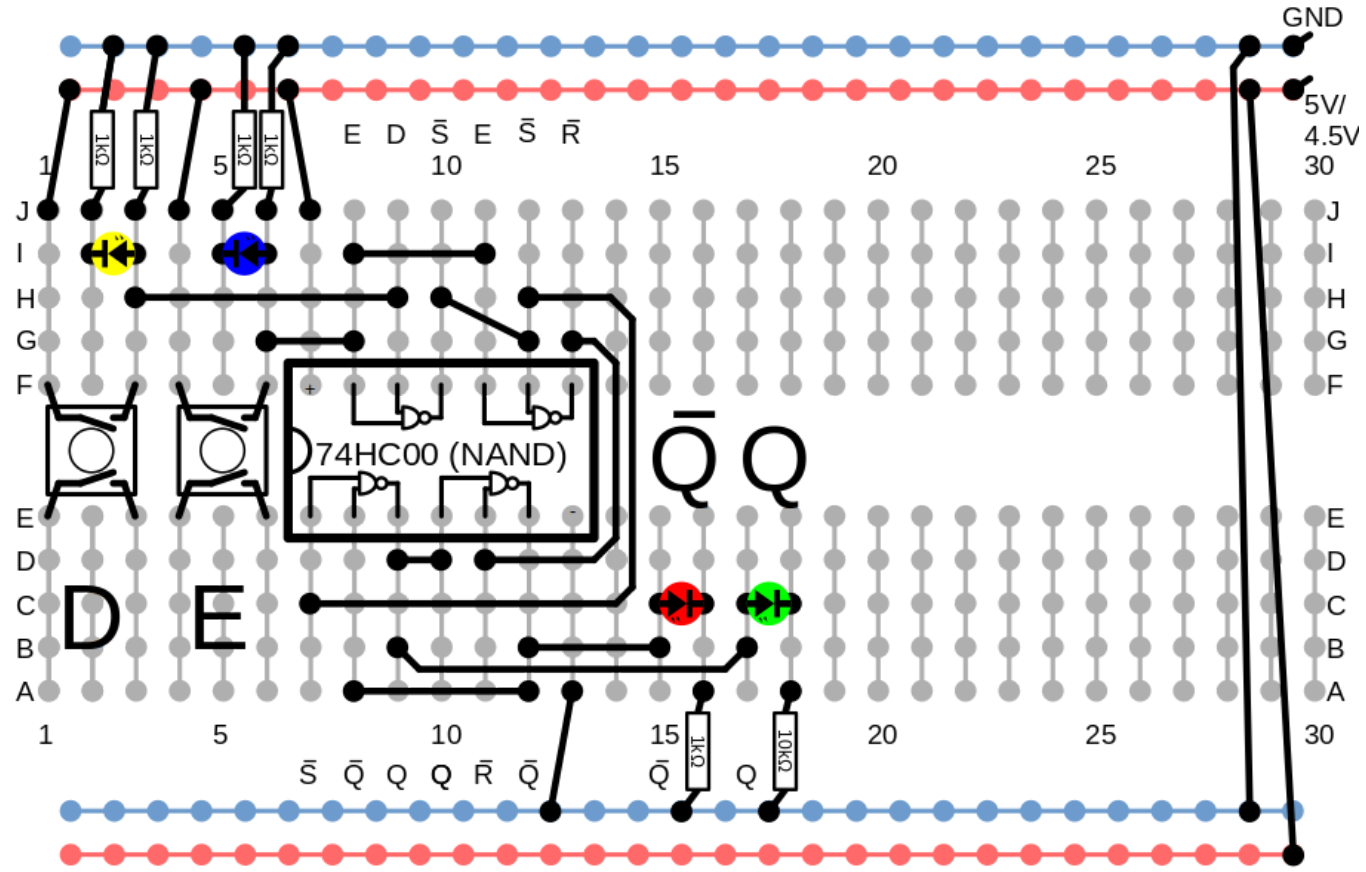


Realisierungsbeispiel

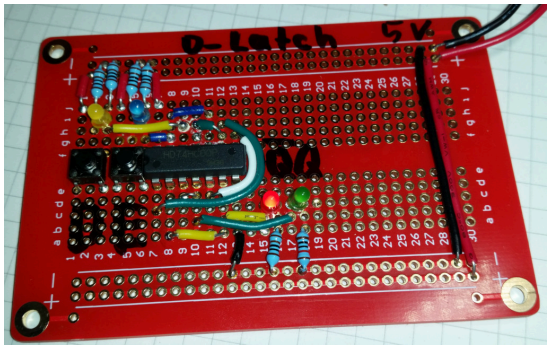
Schaltung basierend auf 74HC00 IC (4x NAND)



Schaltung planen:



Aufbauen:



Übungsaufgaben

- Weisen Sie nach, dass der zweite Ausgang P am RS-Flip-Flop den invertierten Wert von Q realisiert.
- Experimentieren Sie mit der Elektroniksimulation TinkerCAD und einem vorgefertigen Beispiel für ein SR-Flip-Flop. Link