

# Technische Informatik

## Sequentielle Schaltungen

Thorsten Thormählen

07. Dezember 2021

Teil 7, Kapitel 3

Dies ist die Druck-Ansicht.

**Aktiviere Präsentationsansicht**



## Steuerungstasten

- nächste Folie (auch **Enter** oder **Spacebar**).
- ← vorherige Folie
- d** schaltet das Zeichnen auf Folien ein/aus
- p** wechselt zwischen Druck- und Präsentationsansicht
- CTRL** **+** vergrößert die Folien
- CTRL** **-** verkleinert die Folien
- CTRL** **0** setzt die Größenänderung zurück



# Notation

Typ	Schriftart	Beispiele
Variablen (Skalare)	kursiv	$a, b, x, y$
Funktionen	aufrecht	$f, g(x), \max(x)$
Vektoren	fett, Elemente zeilenweise	$\mathbf{a}, \mathbf{b} = \begin{pmatrix} x \\ y \end{pmatrix} = (x, y)^\top,$ $\mathbf{B} = (x, y, z)^\top$
Matrizen	Schreibmaschine	$\mathbf{A}, \mathbf{B} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$
Mengen	kalligrafisch	$\mathcal{A}, \mathcal{B} = \{a, b\}, b \in \mathcal{B}$
Zahlenbereiche, Koordinatenräume	doppelt gestrichen	$\mathbb{N}, \mathbb{Z}, \mathbb{R}^2, \mathbb{R}^3$



# Inhalt

## Sequentielle Schaltungen

Flipflops

Register

Schieberegister

Zähler



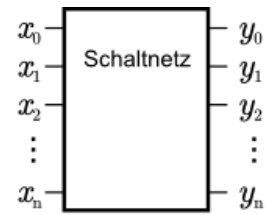
# Sequentielle Schaltungen

## Ohne Rückkopplung (Schaltnetz)

Werte an den Ausgängen sind nur abhängig von den Eingängen

Solche Schaltungen verhalten sich immer gleich (sind zustandslos) und sind durch ihre Schaltfunktion eindeutig beschrieben

Es ist jedoch nicht, möglich etwas zu speichern



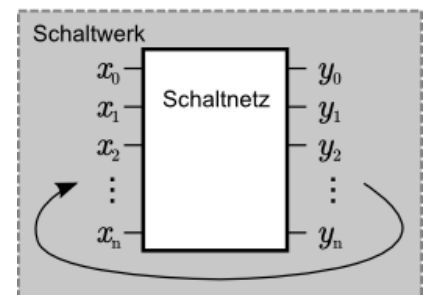
## Mit Rückkopplung (Schaltwerk)

Werte an den Ausgängen sind abhängig von den Eingängen und den vorherigen Ausgangswerten

Das Zeitverhalten muss genau betrachtet werden

Die vorherigen Ausgangswerte können als Zustand der Gatter interpretiert werden. Abhängig vom Zustand verhalten sich die Gatter anders (zustandsabhängige Schaltfunktion)

Es wird möglich, Zustände zu speichern

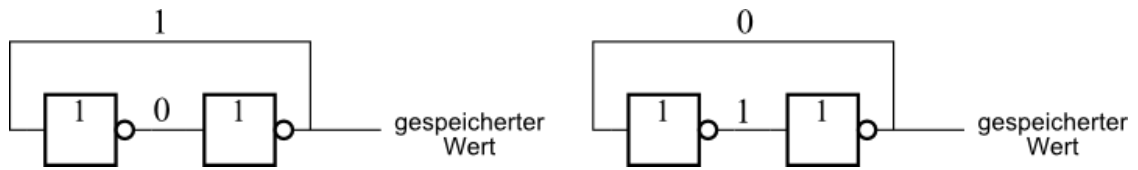




# Speicher mit rückgekoppelten Gattern

Zwei Inverter bilden eine statische Speicherzelle

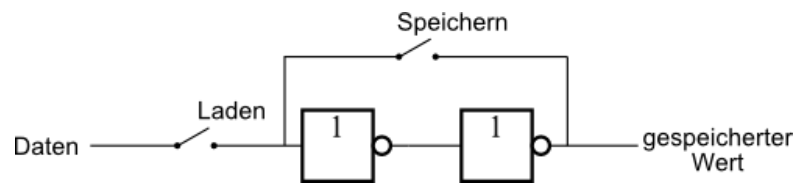
Der Wert bleibt erhalten bis die Versorgungsspannung abgeschaltet wird



Wie kann ein neuer Wert gespeichert werden?

Öffnen des Rückkopplungspfads

Laden des neuen Datenwertes



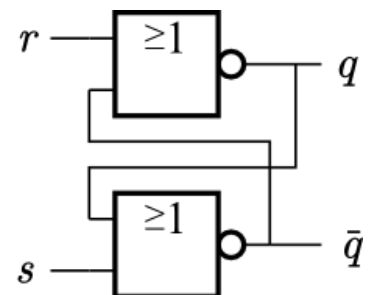
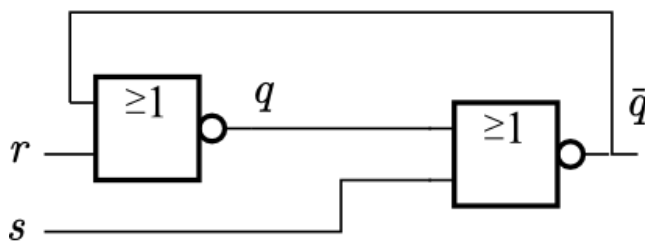


# Speicher mit rückgekoppelten Gattern

## Rückgekoppelte NOR-Gatter

Bei  $r=0$  und  $s=0$  bleibt der aktuelle Wert gespeichert (entspricht der Inverter-Schaltung auf der vorherigen Folie)

Der gespeicherte Wert  $q$  kann mit Reset  $r=1$  auf  $q=0$  und mit Set  $s=1$  auf  $q=1$  gesetzt werden



Thorsten Thormählen 7 / 42

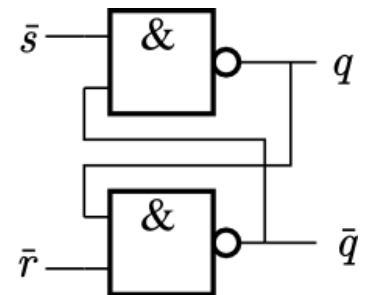
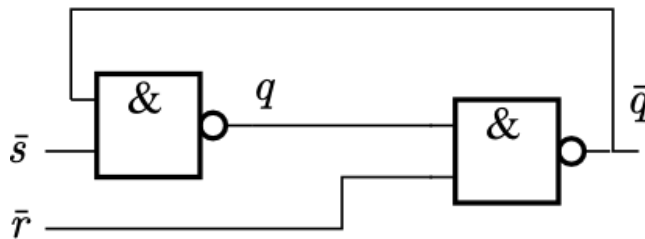


# Speicher mit rückgekoppelten Gattern

## Rückgekoppelte NAND-Gatter

Bei  $\bar{r}=1$  und  $\bar{s}=1$  bleibt der aktuelle Wert gespeichert

Der gespeicherte Wert  $q$  kann mit Reset  $\bar{r}=0$  auf  $q=0$  und mit Set  $\bar{s}=0$  auf  $q=1$  gesetzt werden



Thorsten Thormählen 8 / 42

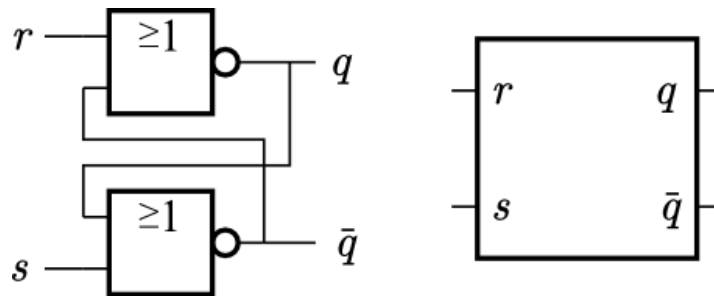


## Asynchrones RS-Flipflop

Diese Schaltung wird asynchrones RS-Flipflop genannt

Das RS-Flipflop kann 1 Bit speichern

Für ein asynchrones RS-Flipflop wird auch folgendes Ersatzschaltbild verwendet



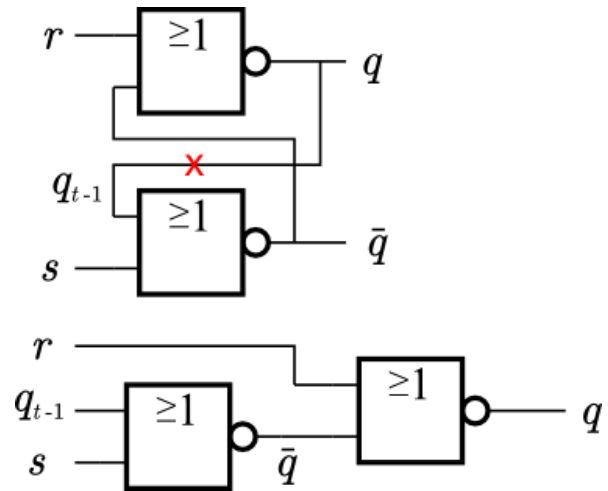


# Asynchrones RS-Flipflop

Wird der vorherige Ausgangswert von  $q$  mit  $q_{t-1}$  bezeichnet, kann folgende Wahrheitstafel aufgestellt werden

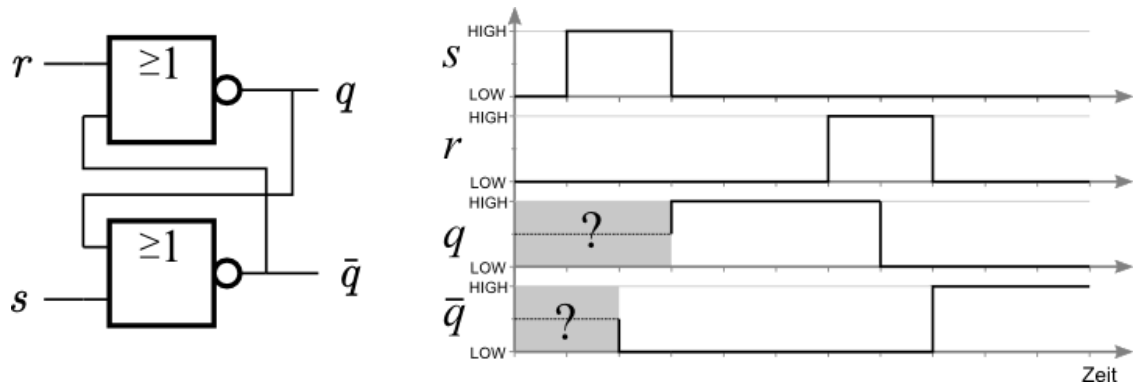
$r$	$s$	$q_{t-1}$	$q$	Funktion
0	0	0	0	halten
0	0	1	1	halten
0	1	0	1	setzen
0	1	1	1	setzen
1	0	0	0	rücksetzen
1	0	1	0	rücksetzen
1	1	0	x	illegal
1	1	1	x	illegal

D.h. der Zustand des vorherigen Ausgangswertes  $q_{t-1}$  hat bei der Schaltung keinen Einfluss auf die Funktion





## Zeitverhalten eines asynchronen RS-Flipflops



Die Eingangsbelegung  $r=1$  und  $s=1$  sollte vermieden werden. Beim gleichzeitigen Wechsel auf 0 fängt das System sonst an zu schwingen (siehe [Simulation in Amilosim](#) )



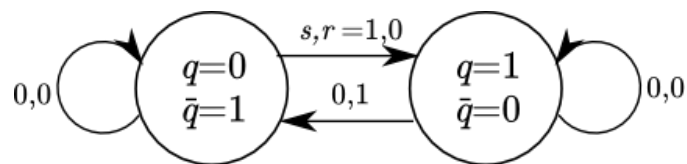
## Zustandsdiagramm eines asynchronen RS-Flipflops

Wird ein RS-Flipflop ohne die illegale Eingangsbelegung  $r=1$  und  $s=1$  betrieben, gibt es zwei Zustände in dem sich das System befinden kann:

Zustand 1:  $q=0$  und  $\bar{q}=1$

Zustand 2:  $q=1$  und  $\bar{q}=0$

In dem Zustandsdiagramm (siehe unten) sind die Zustände mit Kreisen markiert  
Abhängig von den Eingangsvariablen  $r$  und  $s$  können Zustandswechsel auftreten  
(dargestellt durch Pfeile im Zustandsdiagramm)



Das Verhalten des Flipflops an den Ausgängen ist nicht abhängig vom Zustand



# Zustandsdiagramm eines asynchronen RS-Flipflops

Wird ein RS-Flipflop mit der illegalen Eingangsbelegung  $r=1$  und  $s=1$  betrieben, gibt es vier Zustände in dem sich das System befinden kann:

Zustand 1:  $q=0$  und  $\bar{q}=1$

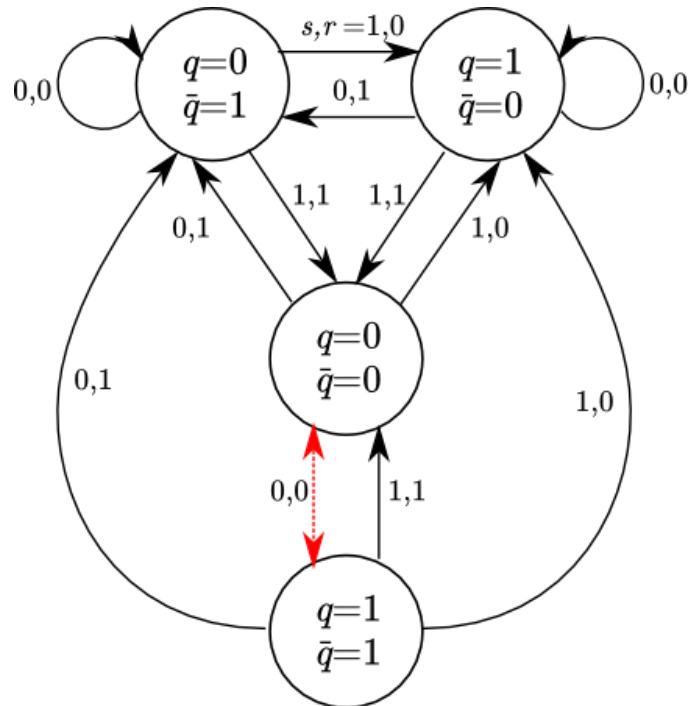
Zustand 2:  $q=1$  und  $\bar{q}=0$

Zustand 3:  $q=0$  und  $\bar{q}=0$

Zustand 4:  $q=1$  und  $\bar{q}=1$

Das Verhalten des Flipflops an den Ausgängen ist jetzt abhängig vom Zustand

Beim gleichzeitigem Wechsel auf  $r=0, s=0$  im Zustand  $q=0, \bar{q}=0$  fängt das System an zu schwingen



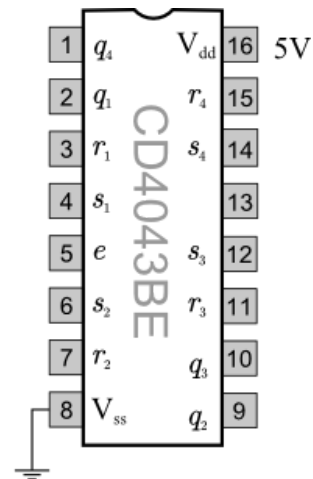
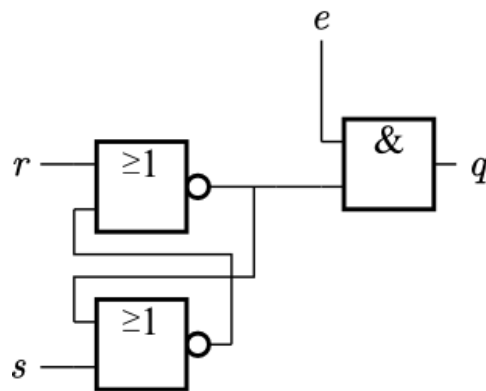


## Praxisbeispiel: Flipflop-Schaltung

Nun soll mit Hilfe des CMOS-IC CD4043BE eine reale Flipflop-Schaltung aufgebaut werden

Der CD4043BE enthält 4 asynchrone RS-Flipflops und kann somit 4 Bits speichern

Der Enable-Pin  $e$  muss auf 1 gesetzt werden, damit die Ausgänge aktiv sind



Thorsten Thormählen 14 / 42



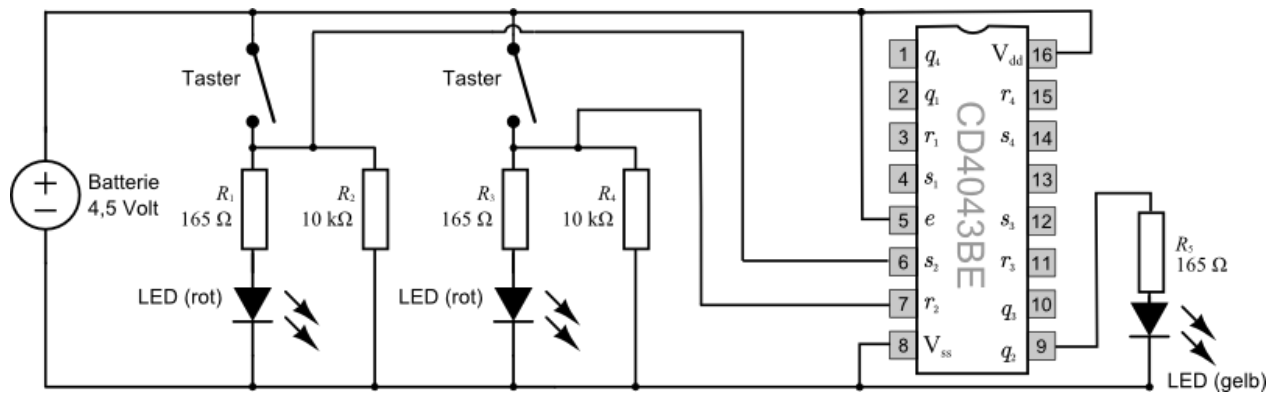
## Praxisbeispiel: Flipflop-Schaltung

Die Pull-Down Widerstände sorgen dafür, dass bei geöffneten Tastern  $r_2=0$  und  $s_2=0$  ist, d.h. das Flipflop hält seinen Ausgangswert  $q_2$

Werden die Taster gedrückt, gilt  $s_2=1$  bzw.  $r_2=1$ .

Bei  $s_2=1$  und  $r_2=0$  wird die LED am Ausgang  $q_2$  eingeschaltet

Bei  $s_2=0$  und  $r_2=1$  wird die LED am Ausgang  $q_2$  ausgeschaltet

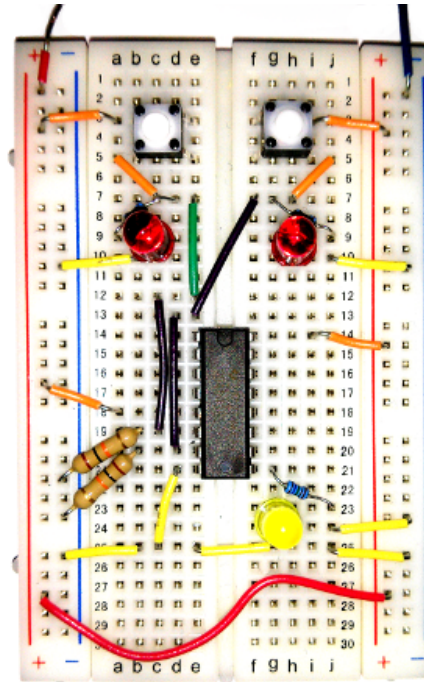


Thorsten Thormählen 15 / 42



## Praxisbeispiel: Flipflop-Schaltung

Dieses Photo zeigt den Aufbau der Schaltung aus der vorangegangenen Folie auf eine Steckplatine



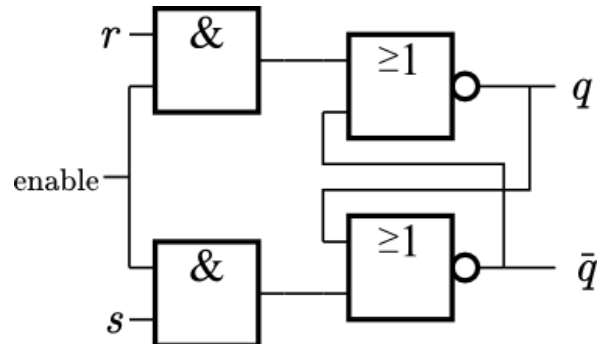
Thorsten Thormählen 16 / 42



## Gesteuertes RS-Flipflops

Beim gesteuerten RS-Flipflop legt der "Enable"-Eingang fest, ob  $r$  und  $s$  eine Rolle spielen

Änderungen am Eingang werden nur noch für  $\text{enable}=1$  übernommen



Diese Maßnahme schützt das Flipflop vor eventuellen Störungen am Eingang solange  $\text{enable}=0$

Das Problem, dass die Schaltung in Schwingung geraten kann, löst die Maßnahme jedoch nicht



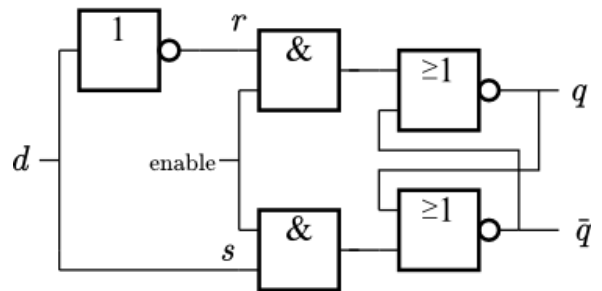
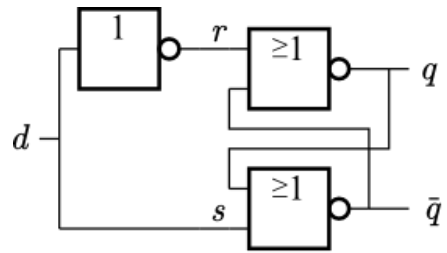
## Gesteuertes D-Flipflop

Die einfachste Lösung, die illegale Eingangsbelegung  $r=1$  und  $s=1$  zu verhindern ist, nur ein Eingangssignal  $d$  zu verwenden (rechts oben)

Allerdings kann die Eingangsbelegung  $s=0$  und  $r=0$ , die das Speichern des Wertes bewirkt, nicht mehr erzeugt werden

Wird die Schaltung mit der "Enable"-Schaltung kombiniert (rechts unten), ist Speichern wieder möglich (enable=0).

Diese Schaltung wird auch D-Flipflop genannt





# Synchrone Flipflops

Durch einen Takt (engl. Clock) kann eine synchrone Schaltung entworfen werden

Der Takt gibt dem System eine gemeinsame Zeitbasis

Zweck bei Flipflops:

Solange warten, bis die Eingänge  $r$  und  $s$  stabil sind

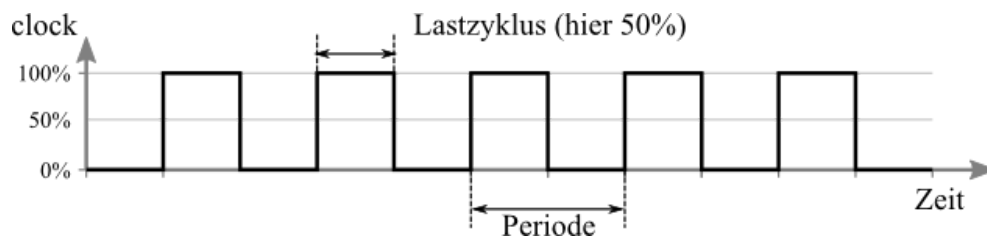
Dann die beabsichtigten Änderungen zulassen ( $\text{enable}=1$ )

Dies kann erreicht werden, indem das Taktsignal clock auf den enable-Eingang gelegt wird

Taktsignale sind periodische Signale

Periodendauer: Zeit zwischen zwei gleichen Flanken

Lastzyklus: hier 50%, da Zeit für 1 gleich lang, wie für 0

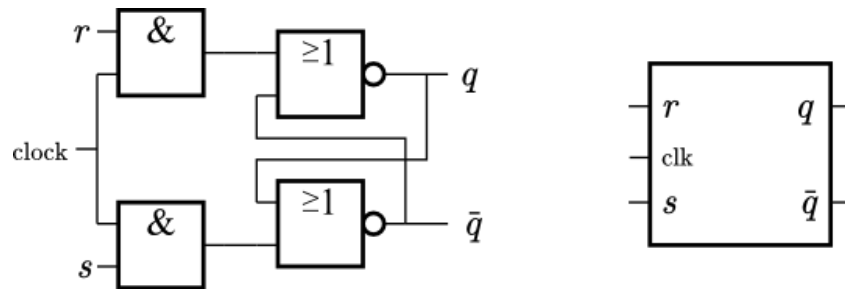


Thorsten Thormählen 19 / 42



# Taktzustandgesteuerte Flipflops

Synchrones RS-Flipflop ([Öffnen in Amilosim](#) )

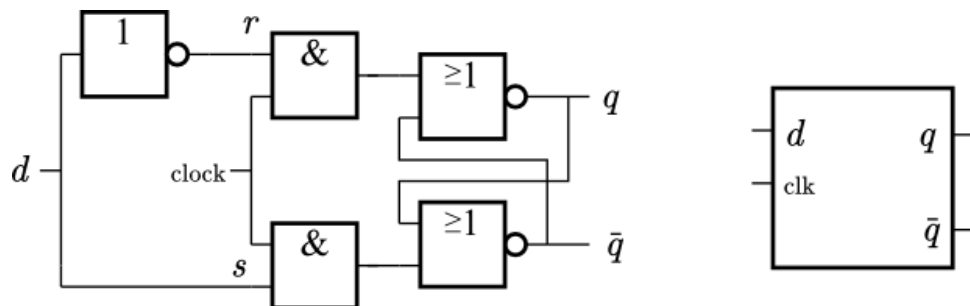


Änderungen am Eingang werden nur während des Lastzyklus übernommen (d.h. wenn  $clock=1$ )



# Taktzustandgesteuerte Flipflops

Synchrones D-Flipflop ([Öffnen in Amilosim](#) )



Änderungen am Eingang werden nur während des Lastzyklus übernommen (d.h. wenn  $clock=1$ )

Ist während des Lastzyklus  $d=1$ , wird  $q=1$  gesetzt und für den Rest der Periode gehalten

Erst beim nächsten Lastzyklus kann sich  $q$  wieder ändern

Das D-Flipflop realisiert damit eine Verzögerung (engl. "Delay"), daher der Name "D"-Flipflop

Thorsten Thormählen 21 / 42



## Taktflankengesteuerte Flipflops

Bisher waren die Flipflops taktzustandsgesteuert

Häufig sinnvoller: taktflankengesteuerte Flipflops

Positiv taktflankengesteuert

Eingänge werden bei der steigenden Flanke abgetastet

Ausgänge ändern sich nach der steigenden Flanke

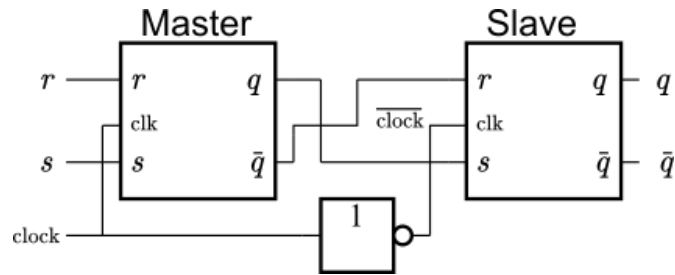
Negativ taktflankengesteuert

Eingänge werden bei der fallenden Flanke abgetastet

Ausgänge ändern sich nach der fallenden Flanke



# Taktflankengesteuertes Flipflop: Master-Slave Flipflop



Durch hintereinander Schalten von zwei taktzustandsgesteuerten RS-Flipflops entsteht ein taktflankengesteuertes Flipflop, auch "Master-Slave Flipflop" genannt:

$r$  und  $s$  werden bei steigender Flanke des  $clock$ -Signals vom Master eingelesen

Während sich die Ausgänge des Masters ändern, passiert beim Slave nichts, da dort das Takt Signal invertiert wurde (Eingang  $\overline{clock}=0$ ) und das Slave Flipflop damit deaktiviert ist

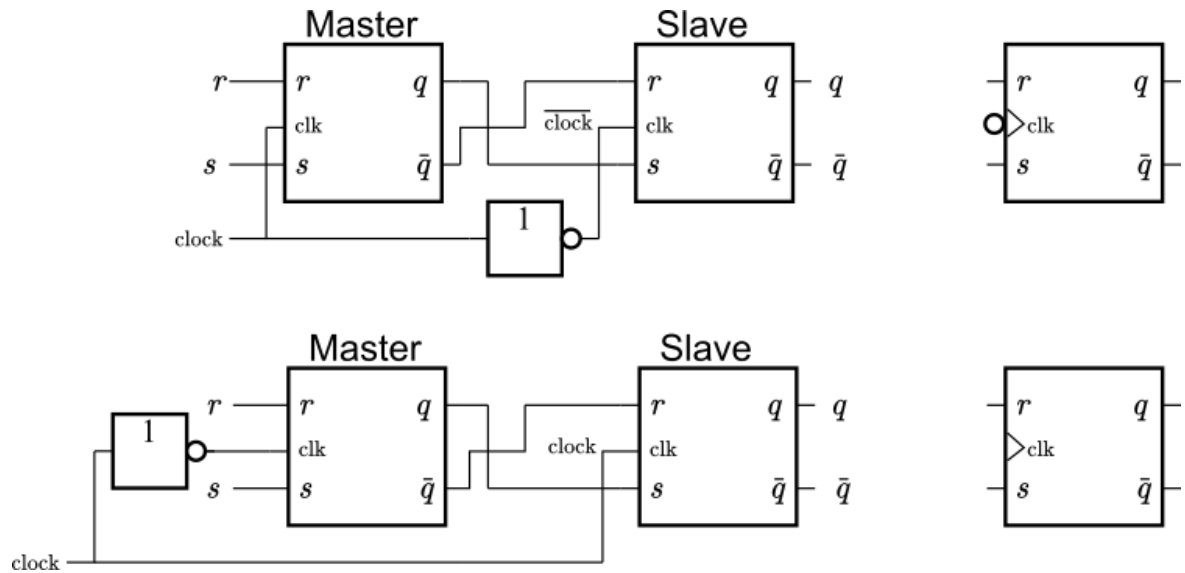
Bei fallender Flanke werden die Ausgänge des Masters vom Slave eingelesen, die Ausgänge des Slaves ändern sich

Eine Änderung der Ausgänge ist beim Master-Slave Flipflop damit nur bei einer fallenden Flanke möglich (egal zu welchem Zeitpunkt sich die Eingangssignale ändern) ([Öffnen in Amilosim](#) )



# Taktflankengesteuertes RS-Flipflop

Es werden die folgenden zwei Ersatzschaltbilder für das taktflankengesteuerte RS-Flipflop verwendet, je nachdem, ob sich die Ausgänge bei fallender (Abb. oben) oder steigender (Abb. unten) Flanke ändern



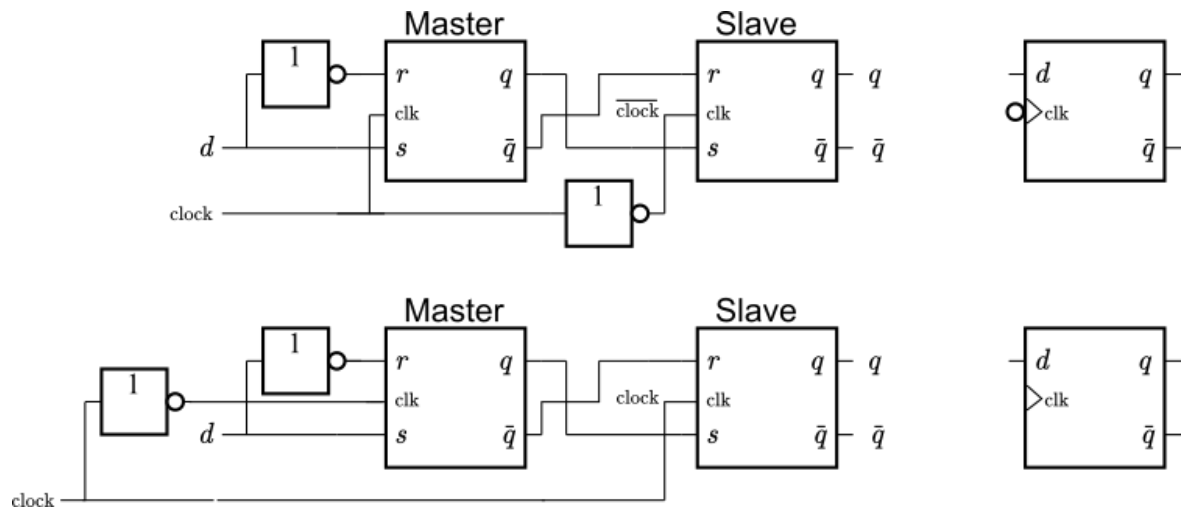
Thorsten Thormählen 24 / 42



# Taktflankengesteuertes D-Flipflop

Ein taktflankengesteuertes RS-Flipflop kann leicht in ein taktflankengesteuertes D-Flipflop umgewandelt werden

Auch für das taktflankengesteuerte D-Flipflop werden Ersatzschaltbilder eingeführt



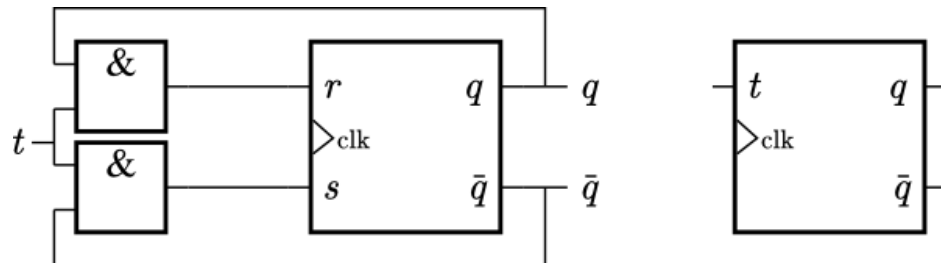
Thorsten Thormählen 25 / 42



# T-Flipflop

Neben RS- und D-Flipflops gibt es auch T-Flipflops

Diese können aus einem RS-Flipflop durch das Vorschalten zweier UND-Gatter erzeugt werden



Wie das D-Flipflop hat das T-Flipflop nur einen Eingang  $t$

Ist  $t=1$  ändert, sich der Ausgang  $q$  von 0 nach 1 bzw. von 1 nach 0. Der Ausgang wird also jeweils umgeschaltet (engl. "toggle"), daher der Name T-Flipflop.

Ist das T-Flipflop einmal in einem stabilen Zustand, bleibt es stabil, da die beiden Ausgänge  $q$  und  $\bar{q}$  nicht gleichzeitig 1 sein können



# JK-Flipflop

Eine weitere Variante, die universell einsetzbar ist, ist das JK-Flipflop

Dieses kann aus einem RS-Flipflop durch das Vorschalten zweier UND-Gatter erzeugt werden

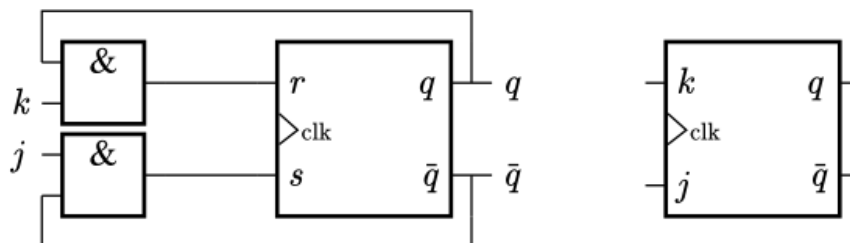
Ist  $j=1$  und  $k=1$ , entspricht das Verhalten dem eines T-Flipflop

Ansonsten dem Verhalten eines RS-Flipflops

D.h. die beim RS-Flipflop illegale

Eingangsbelegung  $j=1$  und  $k=1$  bekommt eine eigene Funktionalität zugewiesen

$k$	$j$	$q_{t-1}$	$q$	Funktion
0	0	0	0	halten
0	0	1	1	halten
0	1	0	1	setzen
0	1	1	1	setzen
1	0	0	0	rücksetzen
1	0	1	0	rücksetzen
1	1	0	1	umschalten
1	1	1	0	umschalten



Thorsten Thormählen 27 / 42



## Register

Ein Register besteht aus mehreren Speicherelementen, die parallel gelesen bzw. geschrieben werden können

Eine wichtige Kenngröße ist die Anzahl  $n$  der Speicherelemente, z.B. 8-, 16-, 32-, 64-, 128-Bit Register

Im Folgenden werden drei verschiedene Registertypen vorgestellt

Auffangregister

Schieberegister

Universalregister

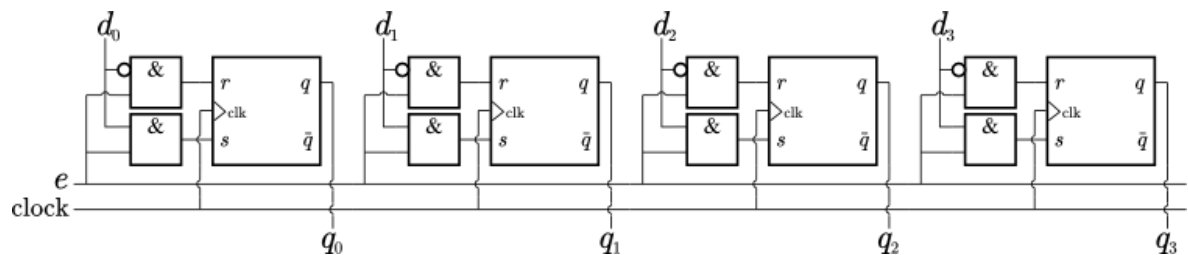


# Auffangregister

Das Auffangregister dient zur Zwischenspeicherung von Datenworten

Ein  $n$ -Bit Auffangregister kann aus einer parallelen Anordnung von  $n$  Flipflops bestehen, die jeweils 1-Bit speichern

Hier wird die Realisierung eines 4-Bit Registers mit taktflankengesteuerten RS-Flipflops gezeigt, die mit einem gemeinsamen Takt  $\text{clock}$  und gemeinsamem Enable-Signal  $e$  angesteuert werden



Wenn  $e=1$  ist, werden die Datenbits  $d_i$  parallel mit steigender Flanke in das Register übernommen

An den Ausgängen  $q_i$  kann das Datenwort ausgelesen werden

Wenn  $e=0$  ist, werden die Ausgänge konstant gehalten, d.h. das Datenwort ist gespeichert

[Bildquelle: frei nach: D. W. Hoffmann: Grundlagen der Technischen Informatik, 2. Auflage, Hanser 2009, Abb. 9.2, S. 310]

Thorsten Thormählen 29 / 42



# Schieberegister

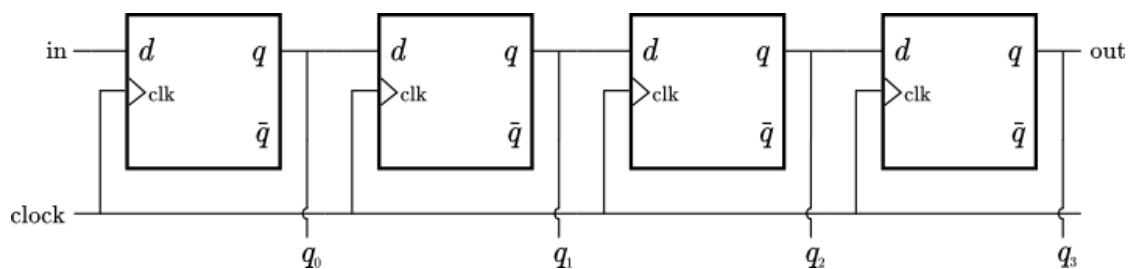
Ein  $n$ -bit Schieberegister hat nur einen Eingang  $in$  aber  $n$  parallele Ausgänge

Mit jedem Takt wird der Eingang  $in$  abgefragt und der Wert im ersten Flipflop gespeichert

Der Ausgang eines Flipflops ist jeweils mit dem Eingang des nächsten Flipflops verbunden

Mit jedem Takt wird die Information ein Register weitergeschoben (daher der Name "Schieberegister")

Hier ist die Realisierung eines 4-Bit Registers mit taktflankengesteuerten D-Flipflops gezeigt

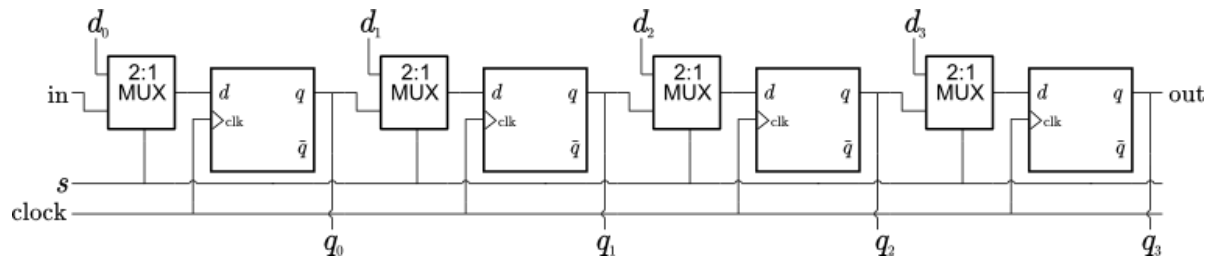


Eine wichtige Anwendung ist das Umsetzen eines seriellen Datenstroms in Datenwörter der Länge  $n$

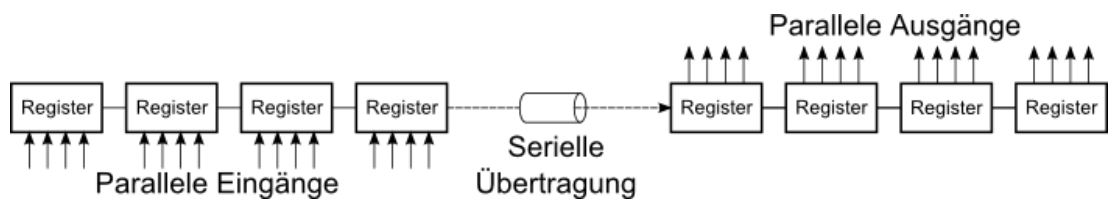


# Schieberegister

Durch Vorschalten eines 2-zu-1 Multiplexers kann das Schieberegister so erweitert werden, dass entweder eine Schiebeoperation ausgeführt wird (Steuerleitung  $s=1$ ) oder Daten parallel geladen werden (Steuerleitung  $s=0$ )



Anwendung: Serielle Übertragung





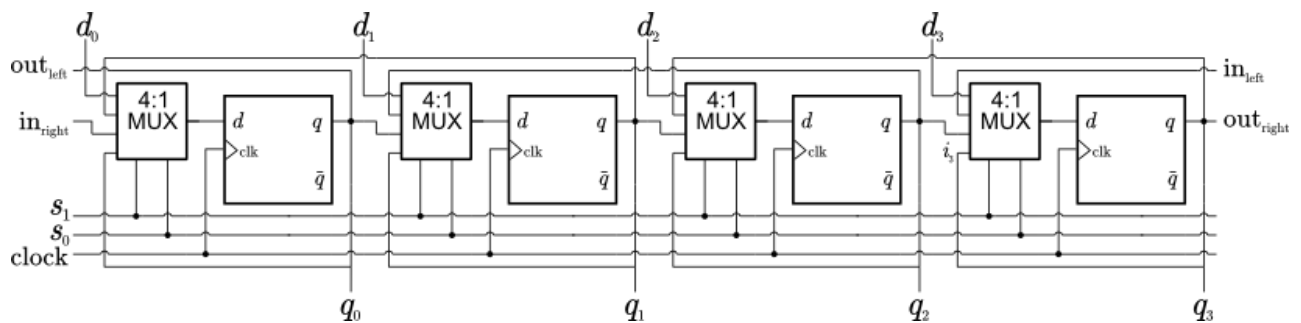
# Universalregister

Das Universalregister vereint die Funktionalität des Auffang- und des Schieberegisters

Das hier gezeigte 4-Bit Universalregister besteht aus 4 taktflankengesteuerten D-Flipflops mit jeweils einem vorgeschalteten 4-zu-1 Multiplexer

Mit den gemeinsamen Steuerleitungen  $s_0$  und  $s_1$  der Multiplexer kann die gewünscht Funktion ausgewählt werden (siehe Tabelle)

$s_1$	$s_0$	Funktion
0	0	laden
0	1	links schieben
1	0	rechts schieben
1	1	halten



Thorsten Thormählen 32 / 42



# Zähler

Mit Registern lassen sich leicht Zähler aufbauen

Im Folgenden werden drei Beispiele gezeigt

Ringzähler

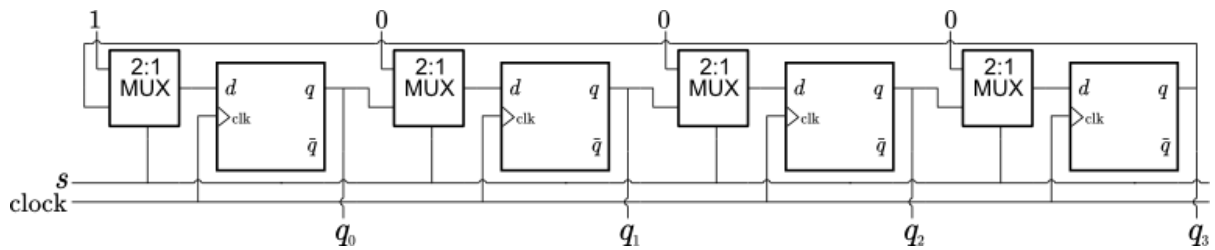
Johnson-Zähler

Binärzähler



# Ringzähler

Zur Realisierung eines Ringzählers kann ein rückgekoppeltes Schieberegister verwendet werden



Ein 4-Bit Ringzähler erzeugt z.B. als Ausgabe die Folge: 1000, 0100, 0010, 0001, 1000, 0100, ...

Es ist also jeweils nur ein Ausgang 1, die anderen 0

Das Schieberegister wird mit 1000 initialisiert (Steuerleitung  $s=0$ )

Anschließend wird die 1 nach rechts durch die Flipflops geschoben (Steuerleitung  $s=1$ )

Ein Ringzähler kann u.a. eingesetzt werden, um Aktionen nacheinander auszuführen. Dazu kann jeder Ausgang  $q_i$  mit einer Aktion verknüpft werden, die ausgeführt wird, sobald der Ausgang 1 ist



# Ringzähler

Ein Ringzähler kann als Frequenzteiler verwendet werden

Wird ein 4-Bit Ringzähler mit 1010 initialisiert, wird die Taktrate halbiert

Initialisierung mit 1100, resultiert in einem 4-fach langsameren Takt

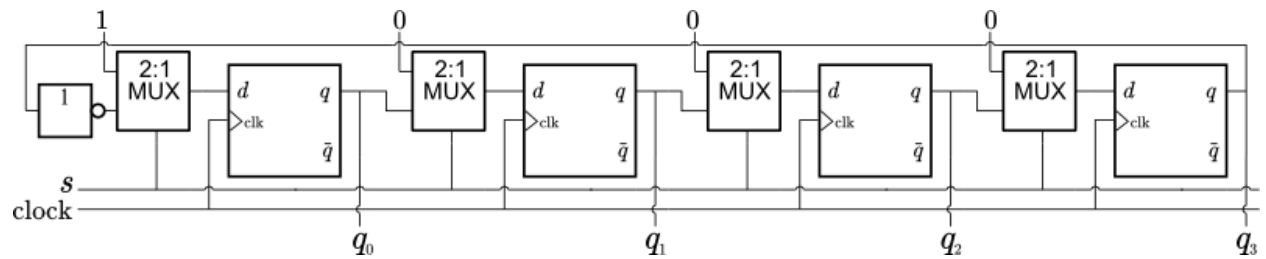
1. Steigende Flanke: 1100
2. Steigende Flanke: 0110
3. Steigende Flanke: 0011
4. Steigende Flanke: 1001 (Zyklus beginnt von vorn)

Der Eingangstakt hat 4 steigende Flanken, das Signal an einem Ausgang nur eine, daher 4-fach verlangsamer Takt



# Johnson-Zähler

Beim Johnson-Zähler (auch Möbius-Zähler genannt) wird der rückgekoppelte Wert invertiert



Dadurch ergibt sich bei Initialisierung mit 1000 die Folge: 1000, 1100, 1110, 1111, 0111, 0011, 0001, 0000, 1000, ...



# Binärzähler

Bei einem Binärzähler werden aufsteigende Binärzahlen erzeugt (hier niederwertiges Bit links)

Es soll also die Folge entstehen:

0000, 1000, 0100, 1100, 0010, 1010, 0110, 1110

0001, 1001, 0101, 1101, 0011, 1011, 0111, 1111

0000, ...

Ein Binärzähler kann ebenfalls mit Flipflops realisiert werden, allerdings werden einige weitere Logikbausteine benötigt

Das niederwertigste Bit  $q_0$  wechselt mit jedem Takt, dies kann mit einem Inverter realisiert werden

$$q_0^{t+1} = \neg q_0^t$$

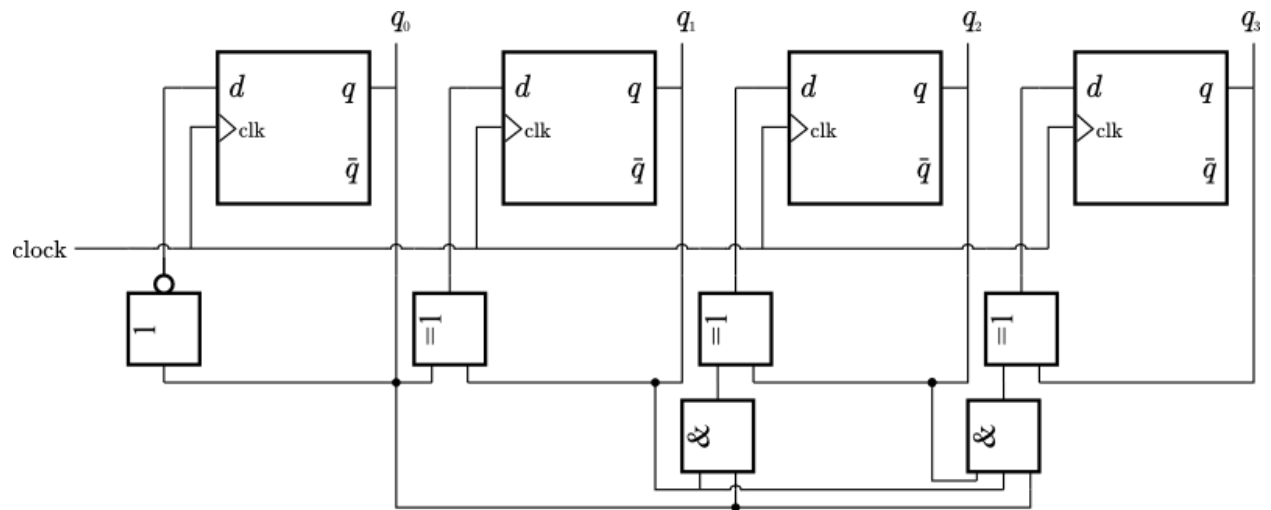
$$\text{Für } q_1 \text{ gilt: } q_1^{t+1} = q_1^t \leftrightarrow q_0^t$$

$$\text{Für } q_2 \text{ gilt: } q_2^{t+1} = q_2^t \leftrightarrow (q_1^t \wedge q_0^t)$$

$$\text{Für } q_3 \text{ gilt: } q_3^{t+1} = q_3^t \leftrightarrow (q_2^t \wedge q_1^t \wedge q_0^t)$$



# Binärzähler



$$q_0^{t+1} = \neg q_0^t$$

$$q_1^{t+1} = q_1^t \leftrightarrow q_0^t$$

$$q_2^{t+1} = q_2^t \leftrightarrow (q_1^t \wedge q_0^t)$$

$$q_3^{t+1} = q_3^t \leftrightarrow (q_2^t \wedge q_1^t \wedge q_0^t)$$



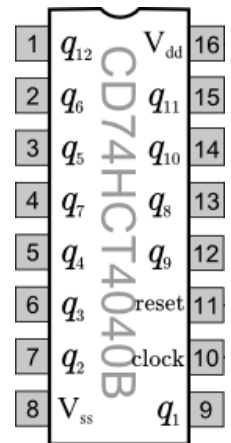
## Praxisbeispiel: Binärzähler

Der CMOS IC CD74HCT4040B ist ein 12-Bit Binärzähler mit den Ausgängen

$q_1, q_2, \dots, q_{12}$

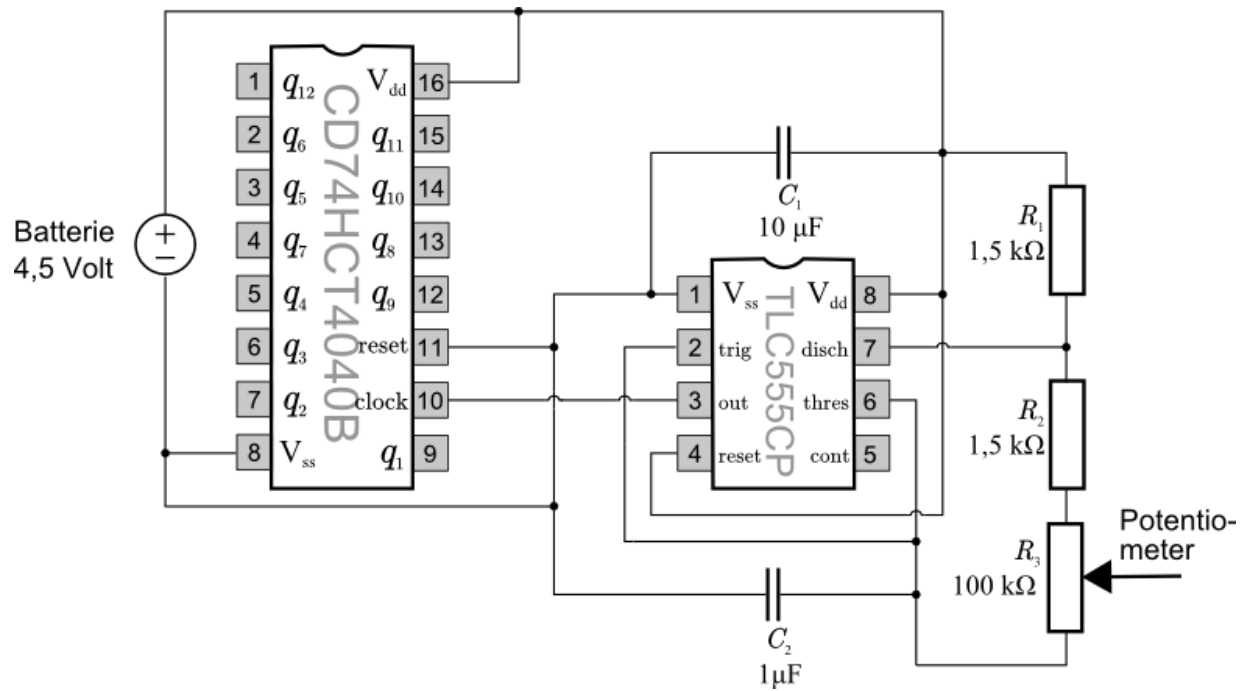
Ist  $\text{reset} = 0$  und liegt am Pin **clock** ein Takt an, zählt der Baustein bei jeder fallenden Flanke um einen Schritt hoch

Der Takt kann z.B. mit dem Timer-Baustein TLC555CP erzeugt werden (nähere Informationen im [Datenblatt](#) )





## Praxisbeispiel: Binärzähler



Thorsten Thormählen 40 / 42



## Praxisbeispiel: Binärzähler

Dieses Video zeigt den Aufbau der Schaltung aus der vorangegangenen Folie auf eine Steckplatine

0:00 / 0:10



Thorsten Thormählen 41 / 42



## Gibt es Fragen?



Anregungen oder Verbesserungsvorschläge können auch gerne per E-mail an mich gesendet werden: [Kontakt](#)

[Weitere Vorlesungsfolien](#)

[\[Impressum\]](#) [\[Datenschutz\]](#)

Thorsten Thormählen 42 / 42

