# Rechnerarchitektur I

# **Schaltwerke**

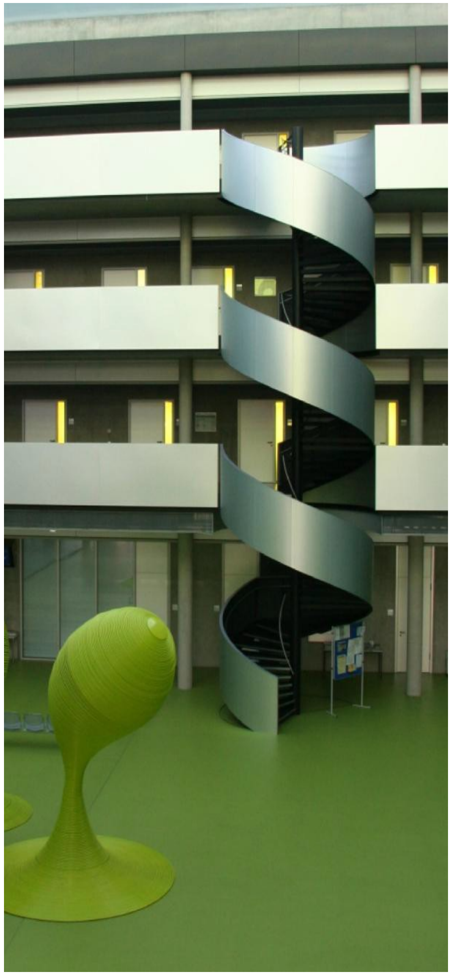
Prof. Dr. Akash Kumar

*Chair for Processor Design*

# Gliederung



- Zielstellung
- Übersicht, Begriffserklärung
- Speicherglieder (Flipflop)
- SR-Flipflop
- Synchrone Speicherglieder
- Komplexe Speicherglieder
- D-Flipflop
- Schaltwerke
- Zustandsautomaten, Automatentheorie
- Darstellung von Schaltwerken
- Analyse von Schaltwerken
- Synthese von Schaltwerken



# Zielstellung



- Kennenlernen einfacher rückgekoppelter Schaltnetze, Basis-Flipflop
- Einführung der funktionellen Bedeutung der Zeit
- Einführung des SR-Flipflop als elementares Speicherglied
- Darstellungsvarianten des SR-Flipflop im Gegensatz zu Schaltnetzen
- Übergang zu synchronen, taktgesteuerten und komplexen Speichergliedern
- Einführung des taktflankengesteuerten D-Flipflop als universelles
- Speicherglied und seiner Darstellungsvarianten
- Vermittlung des Überganges von kombinatorischen zu sequentiellen Schaltungen, von Schaltnetzen zu Schaltwerken
- Kennenlernen einer kurzen Übersicht zur Automatentheorie
- Darstellungsvarianten von Schaltwerken
- Einführung in die Analyse und Synthese von Schaltwerken

# Übersicht, Begriffserklärung



- Schaltwerke – Hauptbestandteile von Computern (v. Neumann Rechner)
- Schaltwerke enthalten Schaltnetze, Speicherglieder (Flipflops) und Signalrückführungen
- der in den Speichergliedern gespeicherte Zustand heißt „innerer Zustand“ des Schaltwerkes
- bei einem Schaltwerk hängen die Werte der Ausgangsvariablen von denen der Eingangsvariablen und vom inneren Zustand (der Vorgeschichte) ab
- einer Wertefolge der Eingangsvariablen ist damit eindeutig eine Wertefolge der Ausgangsvariablen zugeordnet (→Anfangszustand)
- charakteristisch für ein Schaltwerk ist die funktionelle Bedeutung der Zeit (diskrete Zeitpunkte werden durch ein Taktsignal realisiert)
- Flipflops sind einfache Schaltwerke

# Schaltwerk, Sequentielle Schaltung



**Schaltwerk (nach DIN44300):**

- Ein Funktionseinheit zum Verarbeiten von Schaltvariablen, bei der die Werte aller Schaltvariablen am Ausgang (Ausgangsvariablen) zu einem bestimmten Zeitpunkt $t_0$ nach Verstreichen der Lauf- und Verzögerungs-zeit $\Delta t$ nur von den Werten aller Schaltvariablen am Eingang (Eingangsvariablen) zum Zeitpunkt $t\text{-}\Delta t$ abhängen und zu endlich vielen vorangegangenen Zeitpunkten sowie ggf. vom Anfangszustand.
- (Ein Schaltwerk hat eine endlich Anzahl von inneren Zuständen und ist abstrakt gesehen ein endlicher Automat. Falls keine besonderen Vorkehrungen getroffen werden, können Schaltwerke beim Einschalten einen unbestimmten Anfangszustand annehmen.)

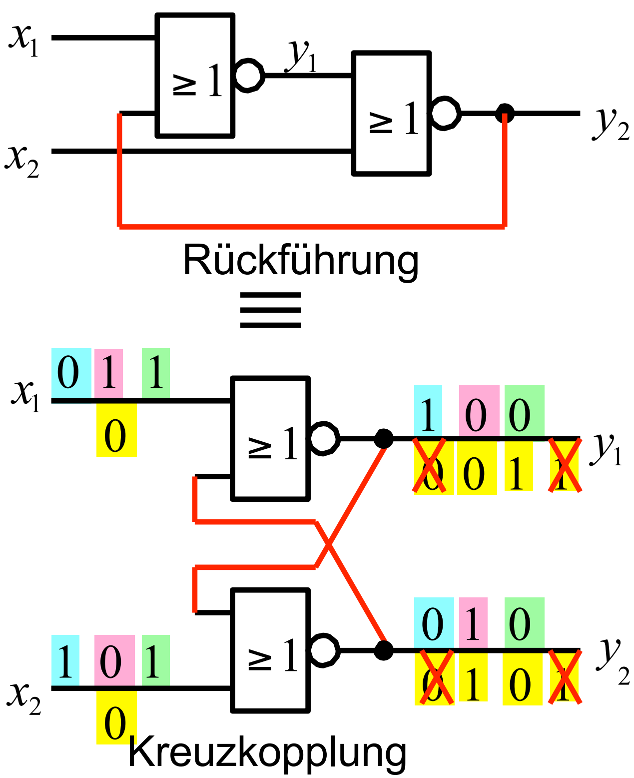
**Speicherglied (nach DIN44300) :**

- Ein Bestandteil eines Schaltwerkes, der Werte von Schaltvariablen aufnimmt, aufbewahrt und abgibt.

# Speicherglieder - Flipflops



## Schaltnetz → Speicherglied → Schaltwerk




Wertetabelle

| $x_1$ | $x_2$ | $y_1$ | $y_2$ |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| | | 1 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 0 |

Bei $x_1=x_2=0$ gibt es zwei mögliche Belegungen für $y_1$, $y_2$
➔**kein Schaltnetz**
($y_1=y_2=0$ und $y_1=y_2=1$ sind keine Lösungen
➔ Widerspruch)

Die Lösungen für $y_1$, $y_2$ bei $x_1=x_2=0$ sind von der vorherigen Belegung der Ausgänge $y_1$, $y_2$ abhängig ➔ Speicherung, Zeitverhalten

# Zeitverhalten, Zustand



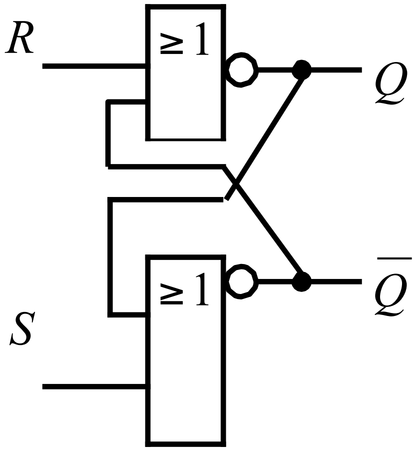
- Außer bei der Eingangsbelegung $x_1=x_2=1$ werden immer komplementäre Ausgangsbelegungen ausgegeben → bistabile Kippschaltung, Flipflop.

  $x_1 \wedge x_2 = 0 \rightarrow y_1 = \overline{y_2}$ $\quad$ $x_1 \wedge x_2 = 1 \rightarrow y_1 = y_2 = 0$

- Die bei der Eingangsbelegung $x_1=x_2=0$ eingenommenen Ausgangswerte werden als Zustand des Flipflop definiert → gespeicherter Zustand.
- Die mit den Eingangsbelegungen $x_1=0, x_2=1$ und $x_1=1, x_2=0$ eingenommene Ausgangswerte $y_1=1, y_2=0$ bzw. $y_1=0, y_2=1$ werden beim Übergang der Eingangswerte zu $x_1=x_2=0$ im Flipflop gespeichert.
- Die mit der Eingangsbelegung $x_1=x_2=1$ eingenommene Ausgangswerte $y_1=y_2=0$ können beim Übergang zu $x_1=x_2=0$ im Flipflop nicht gespeichert werden. Es erfolgt stattdessen nach einer Einschwingphase eine Speicherung von $y_1=1, y_2=0$ oder $y_1=0, y_2=1$ → bistabile Kippschaltung.
- Es handelt sich nicht um ein Schaltnetz, da keine eindeutige Abbildung der Eingangsbelegungen auf die Ausgangsbelegungen möglich ist.

# Funktionelle Bedeutung der Zeit






SR-Flipflop

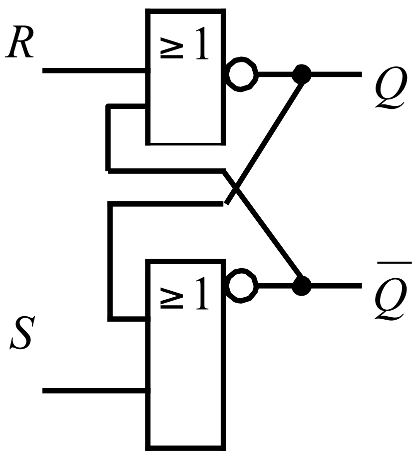
| $R(t_n)$ | $S(t_n)$ | $Q(t_{n+1})$ | $\overline{Q}(t_{n+1})$ | **Zustandsfolgetabelle** |
|---|---|---|---|---|
| 0 | 0 | $Q(t_n)$ | $\overline{Q}(t_n)$ | speichern |
| 0 | 1 | 1 | 0 | setzen |
| 1 | 0 | 0 | 1 | rücksetzen |
| 1 | 1 | – | – | nicht zulässig |

Mit Ausnahme von $R = S = 1$: Die den Eingangsbelegungen $R(t_n)$, $S(t_n)$ zum aktuellen Zeitpunkt $t_n$ entsprechenden Ausgangswerte werden zum Folgezeitpunkt $t_{n+1}$ als neuer Zustand des Flipflop $Q(t_{n+1})$, $\backslash Q(t_{n+1})$ gespeichert.

- $R(t_n) = S(t_n) = 0$ keine Zustandsänderung: $Q(t_{n+1}) = Q(t_n)$, $\backslash Q(t_{n+1}) = \backslash Q(t_n)$
- $R(t_n) = 0$, $S(t_n) = 1$ Zustandsänderung: $Q(t_{n+1}) = 1$, $\backslash Q(t_{n+1}) = 0$
- $R(t_n) = 1$, $S(t_n) = 0$ Zustandsänderung: $Q(t_{n+1}) = 0$, $\backslash Q(t_{n+1}) = 1$
- $R(t_n) = S(t_n) = 1$ nicht zulässig, da Ausgangsbelegung nicht speicherbar.

# SR-Flipflip, Zustandsfolgetabelle






SR-Flipflop
(Basis-Flipflop)

| $R$ $(t_n)$ | $S$ $(t_n)$ | $Q$ $(t_n)$ | $Q^+$ $(t_{n+1})$ | **Zustandsfolgetabelle** |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | speichern |
| 0 | 1 | 0 | 1 | setzen |
| 1 | 0 | 0 | 0 | rücksetzen |
| 1 | 1 | 0 | – | nicht zulässig |
| 0 | 0 | 1 | 1 | speichern |
| 0 | 1 | 1 | 1 | setzen |
| 1 | 0 | 1 | 0 | rücksetzen |
| 1 | 1 | 1 | – | nicht zulässig |

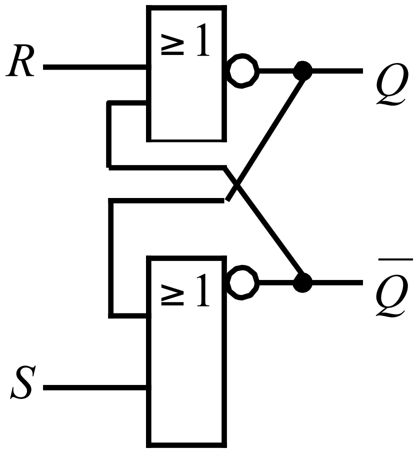
**Kurzdarstellung**

**$R(t_n) \rightarrow R$**
**$S(t_n) \rightarrow S$**
**$Q(t_n) \rightarrow Q$**
**$Q(t_{n+1}) \rightarrow Q^+$**

Abbildung aktueller Zeitpunkt → Folgezeitpunkt

# SR-Flipflip, Zustandsübergangstabelle






SR-Flipflop
(Basis-Flipflop)

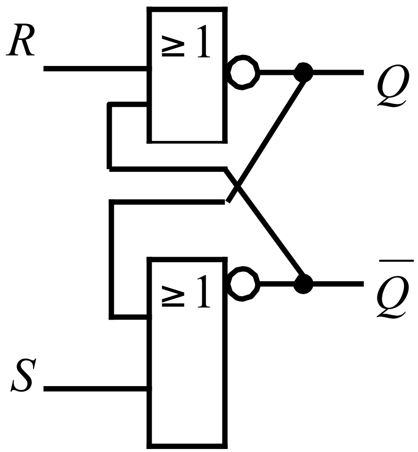
**Zustandsübergangstabelle**

| $(t_n)$ → | $(t_{n+1})$ | $(t_n)$ | | |
|---|---|---|---|---|
| $Q$ → | $Q^+$ | $R$ | $S$ | |
| 0 → | 0 | $X$ | 0 | speichern oder rücksetzen, nicht setzen |
| 0 → | 1 | 0 | 1 | setzen |
| 1 → | 0 | 1 | 0 | rücksetzen |
| 1 → | 1 | 0 | $X$ | speichern oder setzen, nicht rücksetzen |

Antwort auf die Frage: Welche Belegung der Eingänge $R$ und $S$ ist erforderlich, um vom Zustand Q in den Zustand $Q^+$ zu gelangen (→ Zustandsübergang).

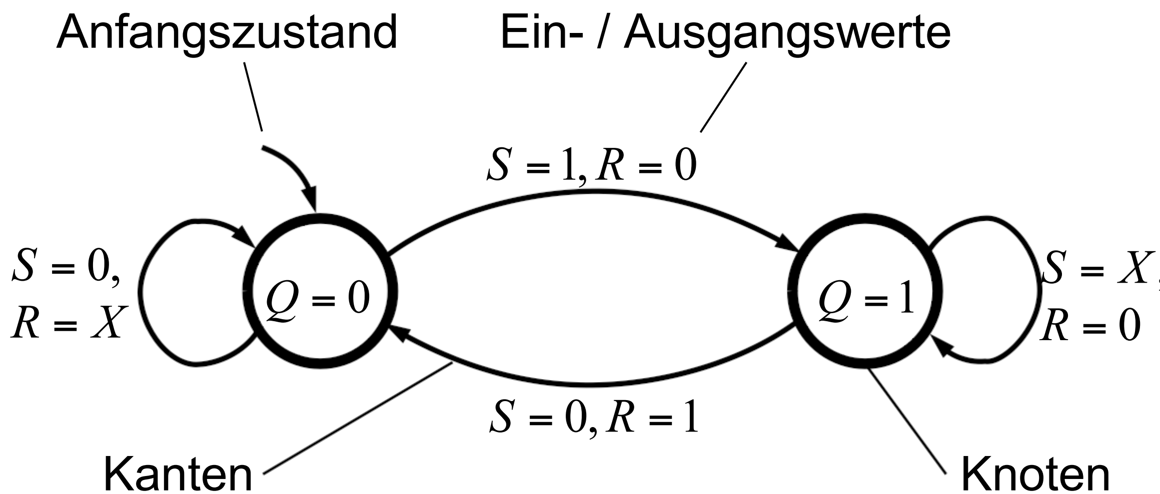
# SR-Flipflip, Zustandsgraph






SR-Flipflop
(Basis-Flipflop)

Knoten: entsprechen den Zuständen
Kanten: entsprechen den Zustandsübergängen
Anfangszustand, einseitige Kante

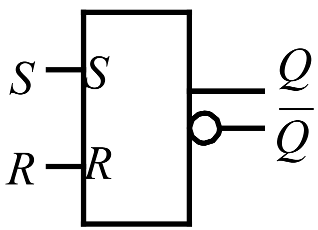



**Zustandsfolgetabelle → Zustandsübergangstabelle → Zustandsgraph**

# SR-Flipflip, Darstellungsvarianten



### Schaltsymbol




SR-Flipflop

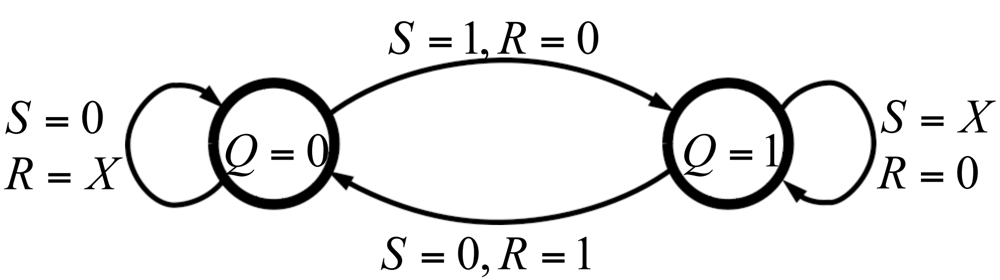
### Zustandsfolgetabelle

| $R$ | $S$ | $Q^+$ | $\overline{Q}^+$ |
|---|---|---|---|
| 0 | 0 | $Q$ | $\overline{Q}$ |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 1 | – | – |

### Zustandsübergangstabelle

| $Q$ | $\to$ | $Q^+$ | $R$ | $S$ |
|---|---|---|---|---|
| 0 | $\to$ | 0 | $X$ | 0 |
| 0 | $\to$ | 1 | 0 | 1 |
| 1 | $\to$ | 0 | 1 | 0 |
| 1 | $\to$ | 1 | 0 | $X$ |

### Zustandsgraph




### Boolesche Gleichung

$$Q^+ := \left(S \wedge \overline{R}\right) \vee \left(\overline{S} \wedge \overline{R} \wedge Q\right)$$

mit $S \wedge R = 0$

# Synchrone Speicherglieder



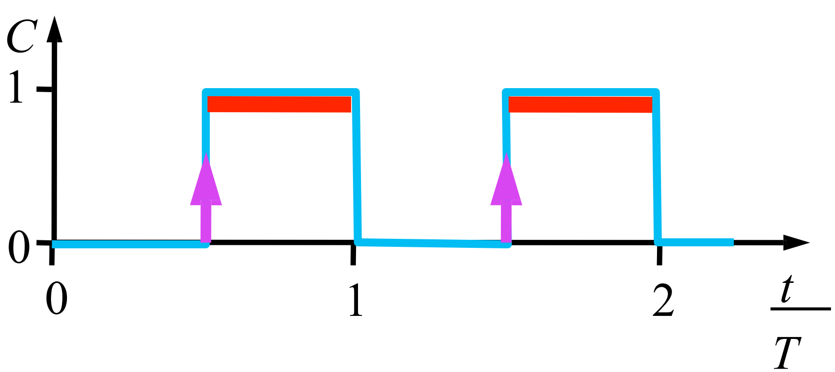
**Asynchrone Speicherglieder:**

Der Zustand und die Ausgangswerte ändern sich unmittelbar nach der Änderung der Eingangswerte → **Datensteuerung** (bisherige Betrachtung)

**Synchrone Speicherglieder:**

Der Zustand und die Ausgangswerte ändern sich synchron zu einem Taktsignal → **Taktsteuerung** (Einführung eines Taktsignales)

**Taktsignal (Clock, Synchronisationssignal):**

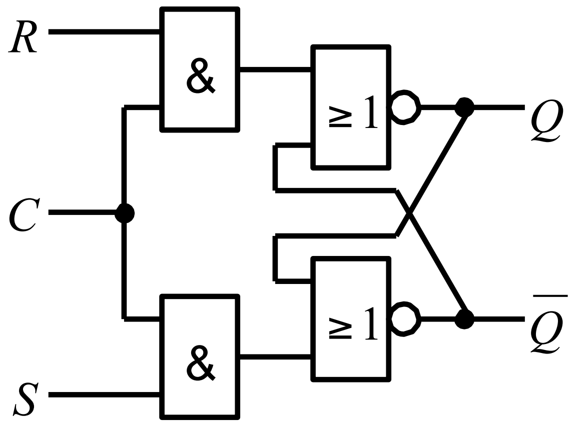


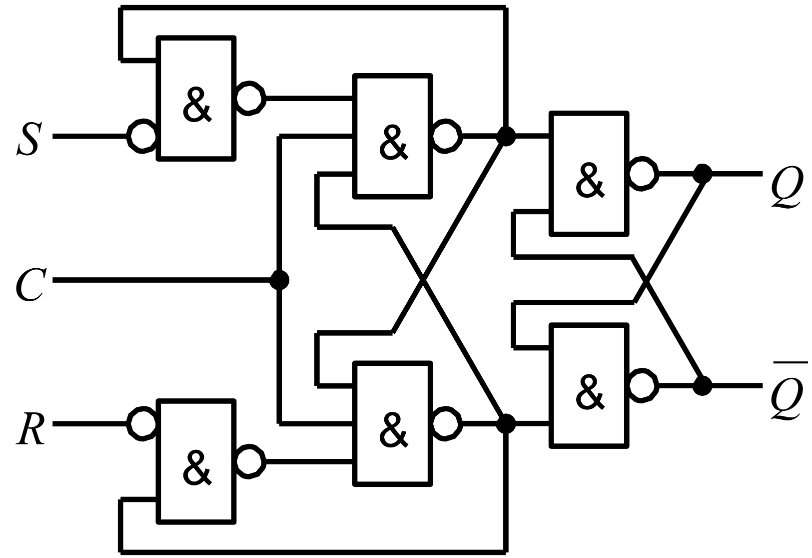

aktiver Taktzustand $C=1$
→ Taktzustandssteuerung

aktive Taktflanke $C\uparrow$ o. $C: 0 \rightarrow 1$
→ Taktflankensteuerung

# Synchrones SR-Flipflop

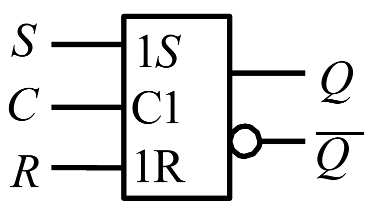




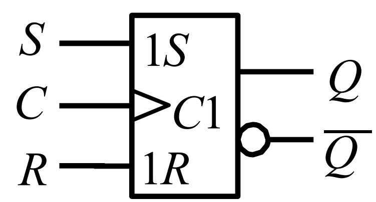

**taktzustandsgesteuert**

Schaltsymbol SR-FF TZS




**taktflankengesteuert**

Schaltsymbol SR-FF TFS



# Synchrones SR-FF, Zustandsfolgetabelle



Taktzustandsgesteuertes SR-Flipflop

| $R$ | $S$ | $C$ | $Q^+$ | |
|---|---|---|---|---|
| 0 | 0 | 1 | $Q$ | speichern |
| 0 | 1 | 1 | 1 | setzen |
| 1 | 0 | 1 | 0 | rücksetzen |
| 1 | 1 | 1 | – | nicht zulässig |
| $X$ | $X$ | 0 | $Q$ | speichern |

Taktflankengesteuertes SR-Flipflop

| $R$ | $S$ | $C$ | $Q^+$ | |
|---|---|---|---|---|
| 0 | 0 | $\uparrow$ | $Q$ | speichern |
| 0 | 1 | $\uparrow$ | 1 | setzen |
| 1 | 0 | $\uparrow$ | 0 | rücksetzen |
| 1 | 1 | $\uparrow$ | – | nicht zulässig |
| $X$ | $X$ | sonst | $Q$ | speichern |

Auf die Angabe des Taktes in der Zustandsfolgetabelle und im Zustandsgraphen kann verzichtet werden, wenn als Nebenbedingung die Art der Taktsteuerung mit angegeben wird (TZS oder TFS).

Ist der Takt nicht aktiv (Zustand oder Flanke), so speichert das Flipflop in jedem Fall den aktuellen Zustand.

# Komplexe Speicherglieder



Ansteuerschaltungen für das SR-Flipflop zur Vermeidung der nicht zulässigen Eingangsbelegung $S=R=1$ und für spezielle Ansteuervarianten und Funktionen

**Flipflop Varianten:**

| Typ | Eingänge | Funktionen |
|---|---|---|
| **T-FF** | 1 | invertieren, speichern |
| **D-FF** | 1 | setzen, rücksetzen |
| **SR-FF** | 2 | speichern, setzen, rücksetzen |
| **JK-FF** | 2 | speichern, setzen, rücksetzen, invertieren |

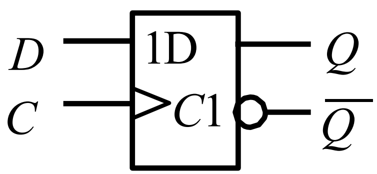
Die einzelnen Flipflop-Typen lassen sich durch Zusatzbeschaltungen ineinander überführen. Alle Flipflop-Typen sind gleichwertig anwendbar.

In der Computertechnik dominieren D-Flipflop (Delay-Flipflop) mit Taktflankensteuerung (teilweise auch mit Taktzustandssteuerung).

# D-Flipflop TFS, Darstellungsvarianten



**Schaltsymbol**




D-Flipflop TFS

**Zustandsfolgetabelle**

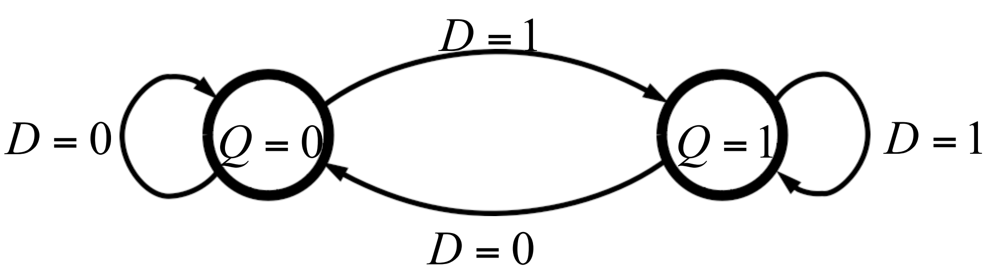
| $D$ | $Q^+$ |
|---|---|
| 0 | 0 |
| 1 | 1 |

**Zustandsübergangstabelle**

| $Q \rightarrow Q^+$ | $D$ |
|---|---|
| $0 \rightarrow 0$ | 0 |
| $0 \rightarrow 1$ | 1 |
| $1 \rightarrow 0$ | 0 |
| $1 \rightarrow 1$ | 1 |

Alle Zustandsübergänge erfolge
aktiven Taktflanke $C\uparrow$ (positive Taktflanke) !

**Zustandsgraph**

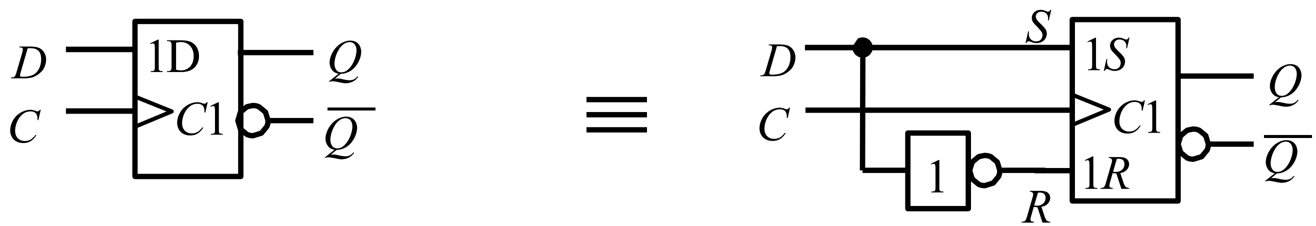



**Boolesche Gleichung**

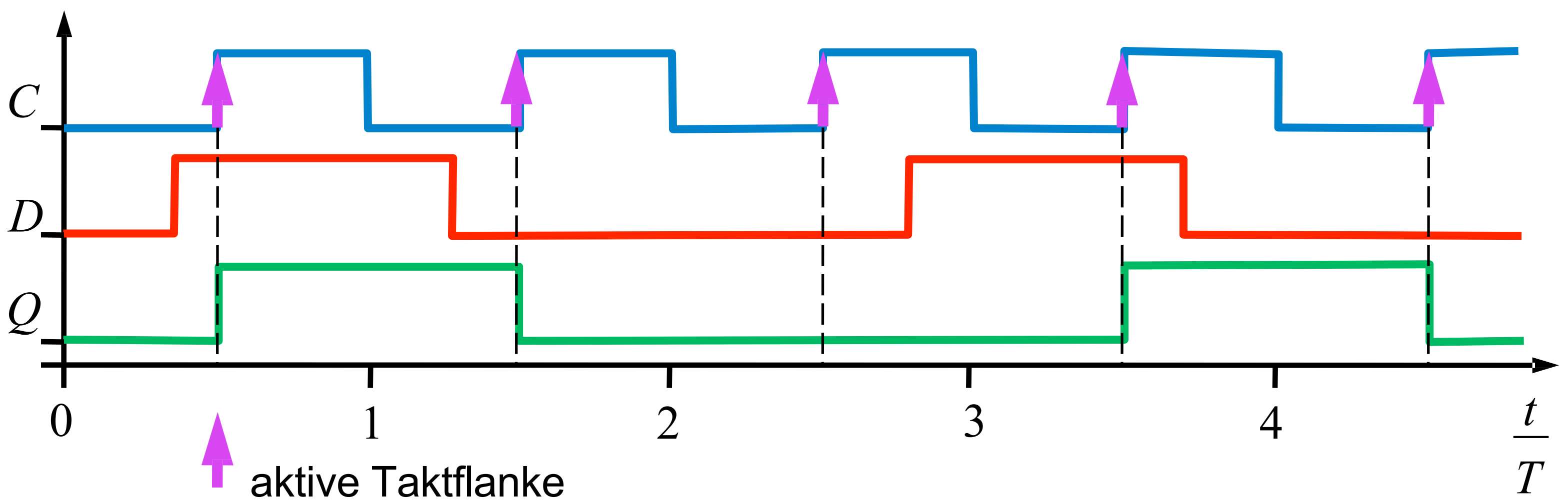
$$Q^+ := D$$

# D-Flipflop TFS, Datenübernahme






D-Flipflop TFS

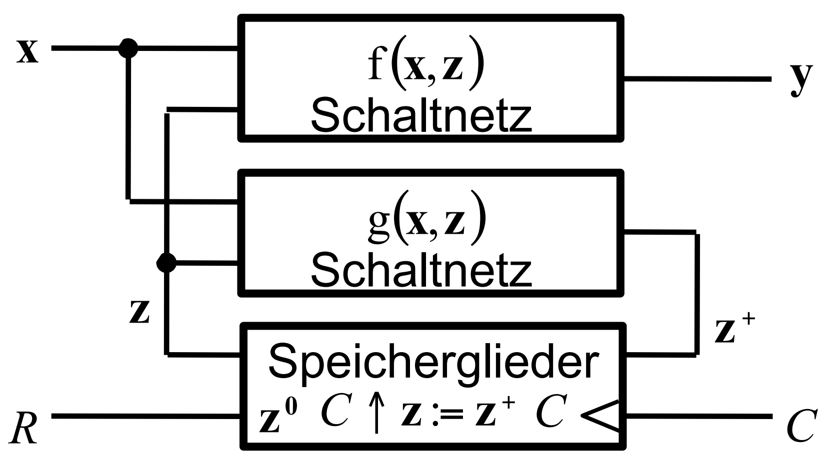


# Schaltwerke – Sequentielle Schaltungen



**Bestandteile von Schaltwerken:**
- Speicherglieder (Flipflop)
- Schaltnetze
- Verbindungen, Rückführungen
- Taktsignal (→ synchrone Schaltwerke).

**Allgemeine Darstellungsform (Huffman-Modell)**




| | |
|---|---|
| $\mathbf{x}$ : | Eingangsvektor |
| $\mathbf{y}$ : | Ausgangsvektor |
| $\mathbf{z}$ : | Zustandsvektor |
| $\mathbf{z}^0$ : | Anfangszustand |
| $\mathbf{z}^+$ : | Folgezustandsvektor |
| $\mathbf{y} = f(\mathbf{x}, \mathbf{z})$ | Ausgangsfunktion |
| $\mathbf{z}^+ = g(\mathbf{x}, \mathbf{z})$ | Übergangsfunktion |
| $C$ : | Taktsignal (clock) |
| $R$ : | Rücksetzsignal (reset) |

# Zustandsautomaten, Automatentheorie



**Automat – Modellmaschine**

- abstraktes mathematisches Modell →
- Kennzeichnung durch einen inneren Zustand → Zustandsautomat
- Automaten arbeiten sequentiell, gesteuert durch Takt oder Eingang.
- Automaten durchlaufen eine Abfolge von Zuständen, beginnend mit einem Anfangszustand, in Abhängigkeit von Eingangswerten oder Takt.
- Die Ausgabewerte der Modellmaschine sind von den aktuellen Eingabewerten und vom momentanen inneren Zustand der Maschine abhängig.

**Endliche Automaten**

Die Menge der möglichen Eingabezeichen (Eingabealphabet), der Ausgabezeichen (Ausgabealphabet) und die Zahl der möglichen inneren Zustände (Zustandsmenge) sind endlich.

**Deterministische Automaten**

Das Verhalten der Modellmaschine ist deterministisch, für eine gegebene Folge von Eingangswerten komplett vorhersagbar, determiniert.

# Zustandsautomat - Schaltwerk



**Automat – Schaltwerk**

- Automaten bilden die Grundlage für den Entwurf und die Beschreibung von sequentiellen digitalen Systemen.
- Schaltwerke stellen eine technische Realisierung der Automaten dar.
- Schaltwerke sind deterministische endliche Automaten (DEA) (auch FSM – Finite State Machine)
- Jeder DEA kann durch ein Schaltwerk realisiert werden und umgekehrt, jedes Schaltwerk kann als DEA beschrieben werden.
- Nichtdeterministische endliche Automaten (NDEA) können durch geeignete Maßnahmen in deterministische endliche Automaten (DEA) umgeformt werden.

# Deterministische Endliche Automaten (DEA)



Ein deterministischer endlicher Automat (DEA) $\mathbf{A}$ kann als 7-Tupel definiert werden:

$$\boxed{\mathbf{A} = (\mathbf{X}, \mathbf{Y}, \mathbf{Z}, \mathrm{f}, \mathrm{g}, \mathbf{z}^{\mathbf{0}}, \mathbf{E})}$$

$$g : \mathbf{X} \times \mathbf{Z} \rightarrow \mathbf{Z}$$
$$f : \mathbf{X} \times \mathbf{Z} \rightarrow \mathbf{Y}$$

| | |
|---|---|
| $\mathbf{X}:$ | Eingabealphabet |
| $\mathbf{Y}:$ | Ausgabealphabet |
| $\mathbf{Z}:$ | Zustandsmenge |
| $\mathbf{y} = \mathrm{f}(\mathbf{x}, \mathbf{z})$ | Ausgangsfunktion |
| $\mathbf{z}^{+} = \mathrm{g}(\mathbf{x}, \mathbf{z})$ | Übergangsfunktion |
| $\mathbf{z}^{\mathbf{0}} \in \mathbf{Z}$ | Anfangszustand |
| $\mathbf{E} \subseteq \mathbf{Z}$ | Finalzustandsmenge |
| $\mathbf{x} \in \mathbf{X}$ | Eingangsvektor |
| $\mathbf{y} \in \mathbf{Y}$ | Ausgangsvektor |
| $\mathbf{z} \in \mathbf{Z}$ | Zustandsvektor |
| $\mathbf{z}^{+} \in \mathbf{Z}$ | Zustandsfolgevektor |

# Deterministische Endliche Automaten (DEA)



$x \in \mathbf{U}$ Menge der Eingangsvariablen
$y \in \mathbf{V}$ Menge der Ausgangsvariablen
$z \in \mathbf{W}$ Menge der Zustandsvariablen

- Mit Hilfe der beiden Funktionen $\mathrm{f}$, $\mathrm{g}$ und dem Anfangszustand $\mathbf{z}^0$ kann das Verhalten eines Automaten ausreichend beschrieben werden.

$$\mathbf{z}^+ = \mathrm{g}(\mathbf{x}, \mathbf{z})$$
$$\mathbf{y} = \mathrm{f}(\mathbf{x}, \mathbf{z})$$
$$\mathbf{z}^0 = (\ )$$
$$\mathbf{z} := \mathbf{z}^+$$ Zustandsübergang der Speicherglieder

- Bei Schaltwerken ist zusätzlich die Beschreibung der Speicherglieder erforderlich (Typ, Steuerung).
- Die Anzahl der Zustände, die ein Automat bei Vorgabe von $\mathrm{f}$, $\mathrm{g}$ und $\mathbf{z}^0$ überhaupt einnehmen kann, kann kleiner sein als die maximal mögliche Zustandsanzahl.

# Zustandsfolge, Ausgangsfolge



| $\mathbf{x}$ | $\mathbf{z}$ | $\mathbf{y}$ |
|---|---|---|
| $\mathbf{x}^0$ | $\mathbf{z}^0 = \mathbf{z}^0$ | $\mathbf{y}^0 = \mathrm{f}\left(\mathbf{x}^0,\mathbf{z}^0\right)$ |
| $\mathbf{x}^1$ | $\mathbf{z}^1 = \mathrm{g}\left(\mathbf{x}^0,\mathbf{z}^0\right)$ | $\mathbf{y}^1 = \mathrm{f}\left(\mathbf{x}^1,\mathbf{z}^1\right) = \mathrm{f}\left(\mathbf{x}^1,\mathrm{g}\left(\mathbf{x}^0,\mathbf{z}^0\right)\right)$ |
| $\mathbf{x}^2$ | $\mathbf{z}^2 = \mathrm{g}\left(\mathbf{x}^1,\mathbf{z}^1\right) = \mathrm{g}\left(\mathbf{x}^1,\mathrm{g}\left(\mathbf{x}^0,\mathbf{z}^0\right)\right)$ | $\mathbf{y}^2 = \mathrm{f}\left(\mathbf{x}^2,\mathbf{z}^2\right) = \mathrm{f}\left(\mathbf{x}^2,\mathrm{g}\left(\mathbf{x}^1,\mathrm{g}\left(\mathbf{x}^0,\mathbf{z}^0\right)\right)\right)$ |
| $\mathbf{x}^3$ | $\mathbf{z}^3 = \mathrm{g}\left(\mathbf{x}^2,\mathbf{z}^2\right) = \mathrm{g}\left(\mathbf{x}^2,\mathrm{g}\left(\mathbf{x}^1,\mathrm{g}\left(\mathbf{x}^0,\mathbf{z}^0\right)\right)\right)$ | $\mathbf{y}^3 = \mathrm{f}\left(\mathbf{x}^3,\mathbf{z}^3\right) = \mathrm{f}\left(\mathbf{x}^3,\mathrm{g}\left(\mathbf{x}^2,\mathrm{g}\left(\mathbf{x}^1,\mathrm{g}\left(\mathbf{x}^0,\mathbf{z}^0\right)\right)\right)\right)$ |
| $\mathbf{x}^n$ | $\mathbf{z}^n = \mathrm{g}\left(\mathbf{x}^{n-1},\mathbf{z}^{n-1}\right)$ | $\mathbf{y}^n = \mathrm{f}\left(\mathbf{x}^n,\mathbf{z}^n\right)$ |

**Zustandsfolge**

$$\mathbf{z}^n = \mathrm{g}\left(\mathbf{x}^{n-1},\mathbf{z}^{n-1}\right) = \mathrm{g}\left(\mathbf{x}^{n-1},\mathrm{g}\left(\mathbf{x}^{n-2},\ldots\mathrm{g}\left(\mathbf{x}^1,\mathrm{g}\left(\mathbf{x}^0,\mathbf{z}^0\right)\right)\right)\right)$$

**Ausgangsfolge**

$$\mathbf{y}^n = \mathrm{f}\left(\mathbf{x}^n,\mathbf{z}^n\right) = \mathrm{f}\left(\mathbf{x}^n,\mathrm{g}\left(\mathbf{x}^{n-1},\ldots\mathrm{g}\left(\mathbf{x}^1,\mathrm{g}\left(\mathbf{x}^0,\mathbf{z}^0\right)\right)\right)\right)$$

# Zustandsbestimmung



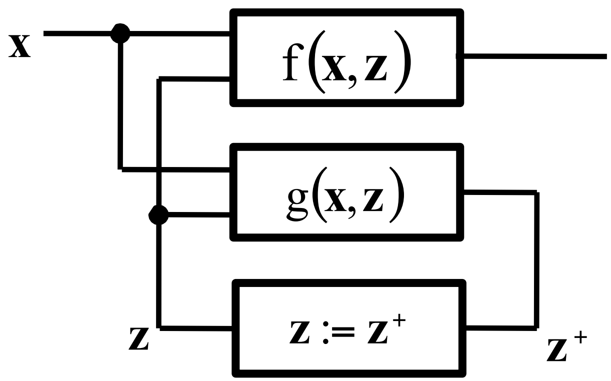
- Der Zustand $z^n$ eines deterministischen endlichen Automaten (DEA) zum Zeitpunkt $t_n$ bestimmt sich entweder aus dem Zustand $z^{n-1}$ und dem Eingangswert $x^{n-1}$ zum vorherigen Zeitpunkt $t_{n-1}$ oder aus dem Anfangszustand $z^0$ und der gesamten Folge der Eingangswerte $x^0, \ldots, x^{n-1}$ von $t_0$ beginnend, bis zum vorherigen Zeitpunkt $t_{n-1}$.
- Im momentanen inneren Zustand des Automaten sind implizit alle seine vorhergehenden Zustände enthalten.
- Beginnend mit einem Anfangszustand $z^0$ bestimmt eine Folge von Eingangswerten $x^0, \ldots, x^n$ eine Folge von Zuständen $z^1, \ldots, z^n$ und entsprechend eine Folge von Ausgangswerten $y^0, \ldots, y^n$.

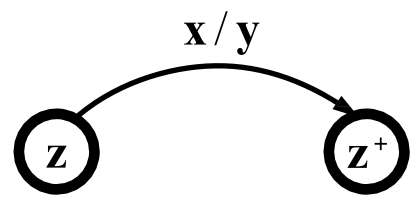
➔ Der Anfangszustand $z^0$ ist dabei von entscheidender Bedeutung.

➔ Hat der Zustandsvektor $z$ eine Dimension von $i$ (Zustandsvariablen), dann sind maximal $2^i$ verschieden Zustände möglich → Zustandskodierung.

# Mealy-Automat (allgemeinste Form)






Zustandsgraph

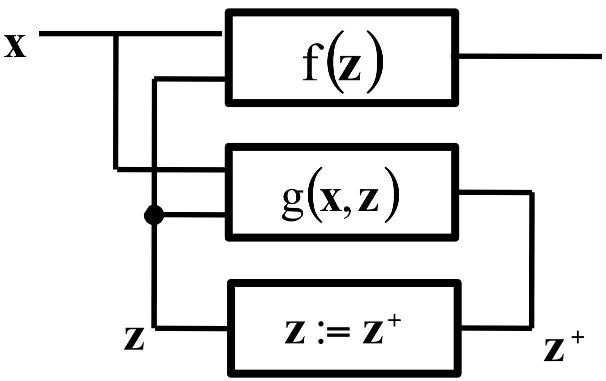
$$
\begin{aligned}
z^{+} &= g(x,z) \\
y &= f(x,z) \\
z^{0} &= (\ ) \\
y^{+} &= f(x^{+}, z^{+}) = f(x^{+}, g(x,z))
\end{aligned}
$$

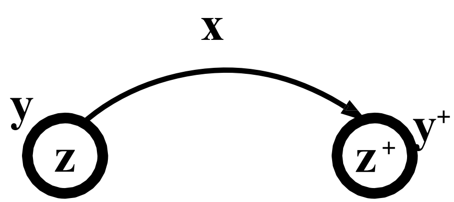
Mealy-Automaten sind **übergangsorientiert**.

Änderungen des Einganges beeinflussen sofort den Ausgang. Sie stellen die allgemeinste Form der deterministischen endlichen Automaten dar.

# Moore-Automat





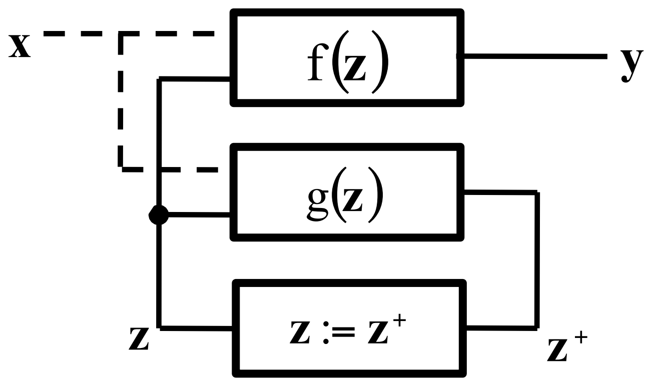

Zustandsgraph

$$z^{+} = g(x, z)$$
$$y = f(z)$$
$$z^{0} = (\quad)$$
$$y^{+} = f(z^{+}) = f(g(x, z))$$

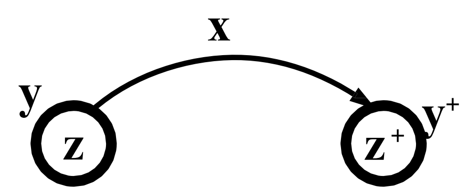
Moore-Automaten sind **zustandsorientiert**.
Änderungen des Einganges beeinflussen den Ausgang erst zum Folgezustand. Sie stellen einen Sonderfall des Mealy-Automaten.

# Autonomer Automat






$$z^{+} = g(z)$$
$$y = f(z)$$
$$z^{0} = (\ )$$
$$y^{+} = f(z^{+}) = f(g(z))$$




Zustandsgraph
(nur eine abgehende Kante pro Knoten)

Autonome Automaten sind nicht eingangsgesteuert.
Änderungen des Einganges beeinflussen somit weder den Ausgang noch den Folgezustand. Sie stellen einen Sonderfall des Moore-Automaten und verfügen nicht über einen Eingang.

# Darstellung von Schaltwerken

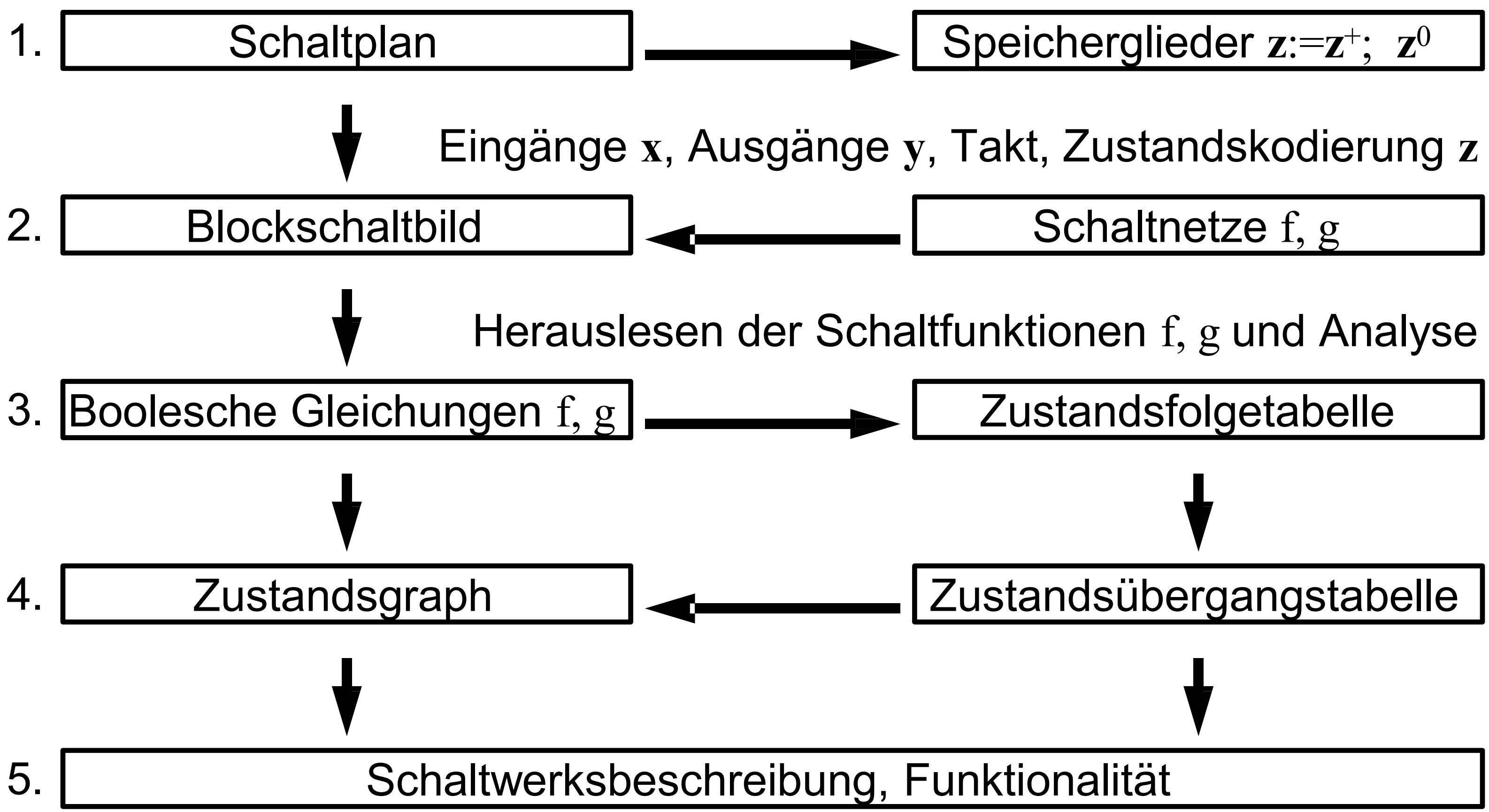


Es gibt verschiedene mögliche Formen der Schaltwerksdarstellung. Alle Varianten sind inhaltlich gleichwertig und können ineinander überführt werden. Bezüglich der Anschaulichkeit und Nutzbarkeit gibt es Unterschiede.

- Boolesche Gleichungen (Automat)
- Zustandsfolgetabelle (Automatentabelle)
- Zustandsübergangstabelle
- Zustandsgraph
- Impulsfolgediagramm
- Schaltplan (Logikplan)
- Schaltsymbol (DIN-Norm)
- Hardwarebeschreibungssprachen (VHDL, Verilog, SystemC, …)
- Programmiersprachen
- Binäre Entscheidungsgraphen (BDD), Speicherelemente

# Analyse von Schaltwerken

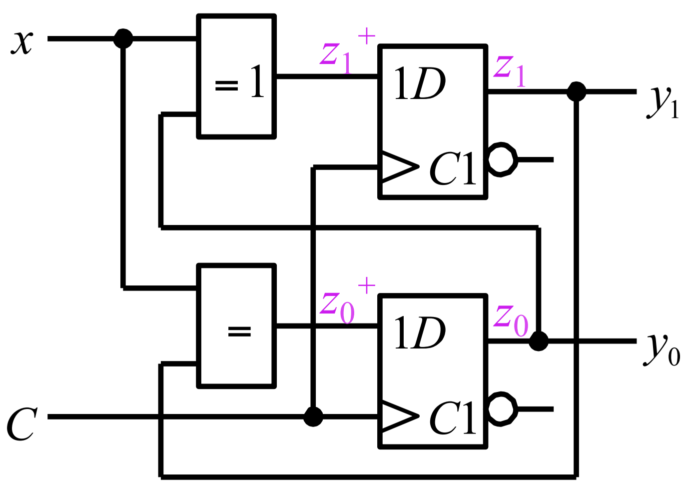




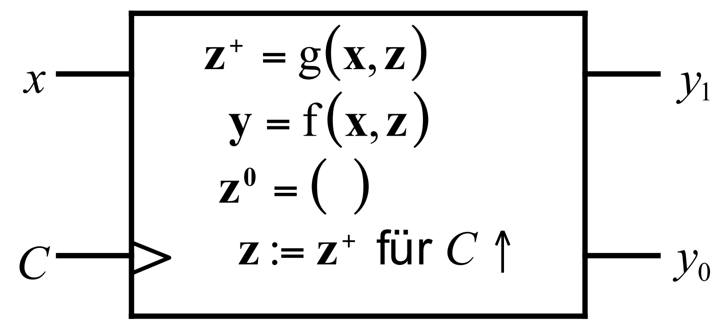
# Analysebeispiel: Ausgangspunkt Schaltplan



Ausgangspunkt Schaltplan




Blockschaltbild




Boolesche Gleichungen

Zustandskodierung:

$$\mathbf{z} = (z_1, z_0)$$

$$\mathbf{z}^+ = (z_1^+, z_0^+)$$

$$\mathbf{x} = (x)$$

$$\mathbf{y} = (y_1, y_0)$$

Moore-Automat: $\mathbf{y} = \mathrm{f}(\mathbf{z})$

$$z_1^+ = x\bar{z}_0 + \bar{x}z_0 \; ; \quad z_1^0 = 0$$

$$z_0^+ = xz_1 + \bar{x}\bar{z}_1 \; ; \quad z_0^0 = 0$$

$$y_1 = z_1$$

$$y_0 = z_0$$

# Zustandsfolgetabelle (Automatentabelle)



| $x$ | $\mathbf{x}$ | $z_1$ | $z_0$ | $\mathbf{z}$ | $\mathbf{z}^+$ | $z_1^+$ | $z_0^+$ | $y_1$ | $y_0$ | $\mathbf{y}$ |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | $\mathbf{x_0}$ | 0 | 0 | $\mathbf{z_0}$ | $\mathbf{z}_1$ | 0 | 1 | 0 | 0 | $\mathbf{y_0}$ |
| 0 | $\mathbf{x_0}$ | 0 | 1 | $\mathbf{z_1}$ | $\mathbf{z}_3$ | 1 | 1 | 0 | 1 | $\mathbf{y}_1$ |
| 0 | $\mathbf{x_0}$ | 1 | 0 | $\mathbf{z}_2$ | $\mathbf{z}_0$ | 0 | 0 | 1 | 0 | $\mathbf{y}_2$ |
| 0 | $\mathbf{x_0}$ | 1 | 1 | $\mathbf{z}_3$ | $\mathbf{z}_2$ | 1 | 0 | 1 | 1 | $\mathbf{y}_3$ |
| 1 | $\mathbf{x_1}$ | 0 | 0 | $\mathbf{z}_0$ | $\mathbf{z}_2$ | 1 | 0 | 0 | 0 | $\mathbf{y_0}$ |
| 1 | $\mathbf{x_1}$ | 0 | 1 | $\mathbf{z}_1$ | $\mathbf{z}_0$ | 0 | 0 | 0 | 1 | $\mathbf{y}_1$ |
| 1 | $\mathbf{x_1}$ | 1 | 0 | $\mathbf{z}_2$ | $\mathbf{z}_3$ | 1 | 1 | 1 | 0 | $\mathbf{y}_2$ |
| 1 | $\mathbf{x_1}$ | 1 | 1 | $\mathbf{z}_3$ | $\mathbf{z}_1$ | 0 | 1 | 1 | 1 | $\mathbf{y}_3$ |

$$\mathbf{x}_0 = (0)$$
$$\mathbf{x}_1 = (1)$$
$$\mathbf{y}_0 = (0,0)$$
$$\mathbf{y}_1 = (0,1)$$
$$\mathbf{y}_2 = (1,0)$$
$$\mathbf{y}_3 = (1,1)$$
$$\mathbf{z}_0 = (0,0)$$
$$\mathbf{z}_1 = (0,1)$$
$$\mathbf{z}_2 = (1,0)$$
$$\mathbf{z}_3 = (1,1)$$
$$\mathbf{z}^0 = \mathbf{z}_0$$

Anfangszustand.

Angabe von Vektoren oder und Komponenten, je nach Übersichtlichkeit.

# Zustandsübergangstabelle

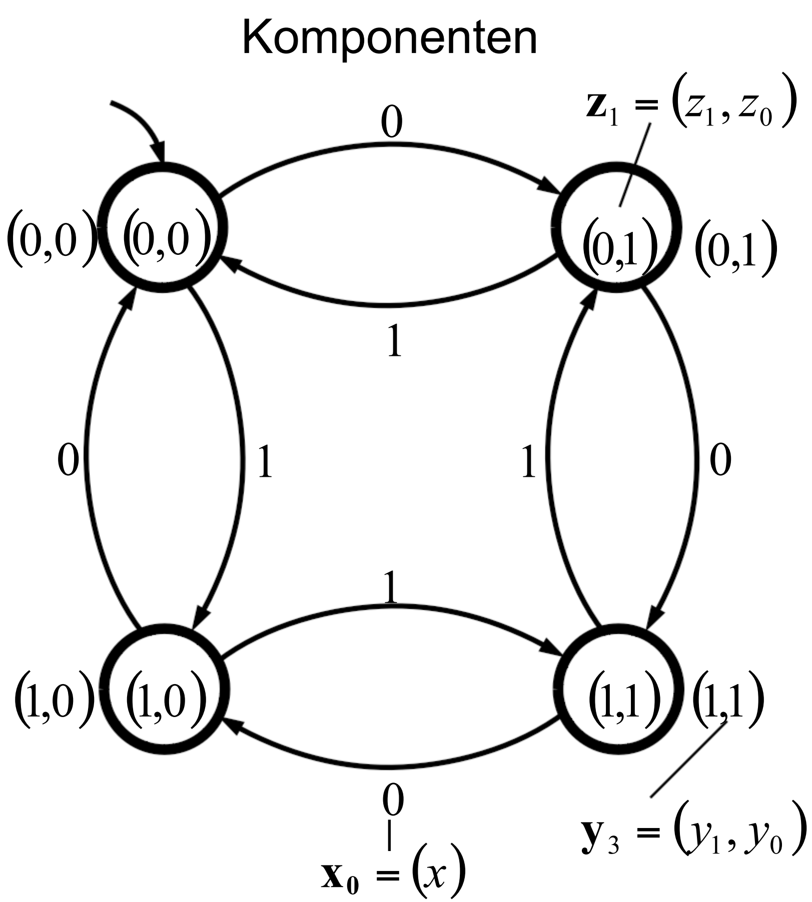


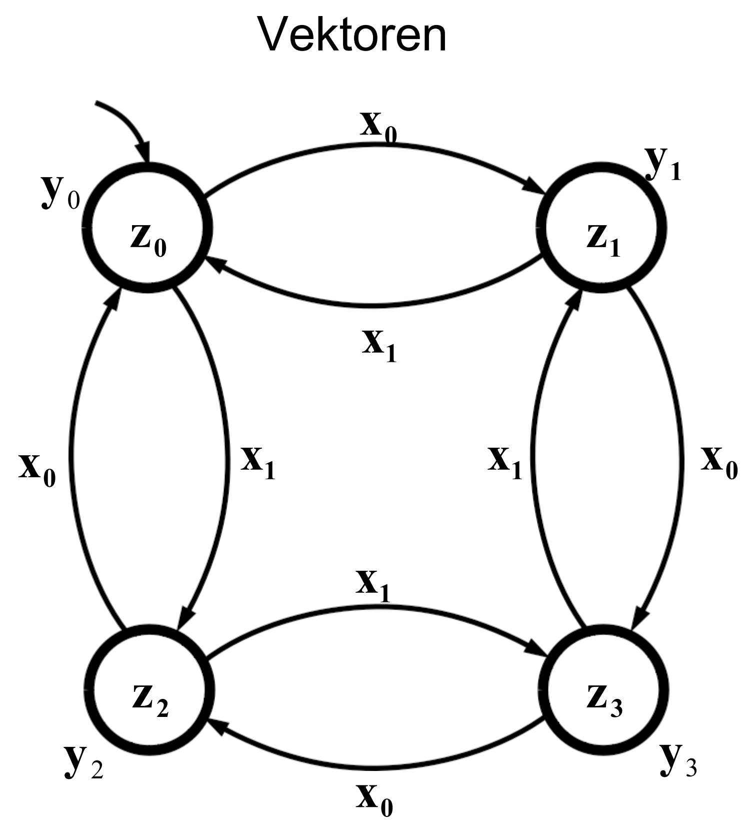
| | | | $\mathbf{x_0}$ | $x=$ | $0$ | $\mathbf{x_1}$ | $x=$ | $1$ | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $z_1$ | $z_0$ | $\mathbf{z}$ | $\mathbf{z}^+$ | $z_1^+$ | $z_0^+$ | $\mathbf{z}^+$ | $z_1^+$ | $z_0^+$ | $y_1$ | $y_0$ | $\mathbf{y}$ |
| 0 | 0 | $\mathbf{z}_0$ | $\mathbf{z}_1$ | 0 | 1 | $\mathbf{z_2}$ | 1 | 0 | 0 | 0 | $\mathbf{y_0}$ |
| 0 | 1 | $\mathbf{z}_1$ | $\mathbf{z}_3$ | 1 | 1 | $\mathbf{z}_0$ | 0 | 0 | 0 | 1 | $\mathbf{y}_1$ |
| 1 | 0 | $\mathbf{z}_2$ | $\mathbf{z}_0$ | 0 | 0 | $\mathbf{z}_3$ | 1 | 1 | 1 | 0 | $\mathbf{y}_2$ |
| 1 | 1 | $\mathbf{z}_3$ | $\mathbf{z_2}$ | 1 | 0 | $\mathbf{z}_1$ | 0 | 1 | 1 | 1 | $\mathbf{y}_3$ |

Vektoren und oder Komponenten
Bei einem Mealy-Automaten ist der Ausgangsvektor pro Eingangsvektor anzugeben, parallel zu den Folgezuständen.
Beim Moore-Automaten reicht eine Angabe parallel zum Zustand aus.

# Zustandsgraph





# Schaltwerksbeschreibung



Das im Schaltplan gegebene Schaltwerk realisiert folgende Eigenschaften:

- zwei Speicherglieder realisieren vier verschiedene Zustände
- Moore-Automat
- folgende Zustandsfolge wird in Abhängigkeit vom Eingang realisiert:

  $x = 0: \ (0,0) \to (0,1) \to (1,1) \to (1,0) \to (0,0) \to (0,1) \to \ldots$

  $x = 1: \ (0,0) \to (1,0) \to (1,1) \to (0,1) \to (0,0) \to (1,0) \to \ldots$
- es handelt sich um einen 2-bit Gray-Code zyklischen Synchronzähler mit Vor- und Rückwärtssteuerung (Zustandskodierung Gray-Code)

  $x = 0:$ vorwärts zählen

  $x = 1:$ rückwärts zählen
- Zählrichtung kann in jedem beliebigen Zustand umgekehrt werden
- ausgegeben wird direkt der Zählerzustand (Gray-Code)
- Startzustand ist $\mathbf{z}^{\mathbf{0}} = \mathbf{z}_{\mathbf{0}} = (0,0)$.

# Synthese von Schaltwerken