Grundlagen der Informatik 2
Grundlagen der Digitaltechnik

6. Automaten und Schaltwerke

Prof. Dr.-Ing. Jürgen Teich
Dr.-Ing. Christian Haubelt
Lehrstuhl für Hardware-Software-Co-Design

Automaten - Motivation

- **<u>Schaltnetze:</u> Bilden Ausgangsgrößen unmittelbar** aus (als Funktion von) **Eingangsgrößen**
- **Nachteil: Keine Zustände darstellbar**

- **<u>Notwendig:</u>** ein **neues Konzept**, welches die Möglichkeit hat, **Zustände** zu speichern und die Ausgabe abhängig vom Zustand des Systems zu beschreiben

- Dieses Konzept basiert auf dem Modell von **Automaten**, deren Realisierungen wir als **Schaltwerk** bezeichnen.

Beispiel: Getränkeautomat

- Ein *dümmlicher* Getränkeautomat erwartet, dass man im ersten Schritt einen Euro einwirft und dann im zweiten Schritt eines von zwei möglichen Getränken auswählt.
- Verhält man sich anders, zeigt der Automat keine Reaktion.
- Man kann aber auf den Rückgabeknopf drücken.
- Dann gibt der Automat das Geld zurück, für das er keine Ware geliefert hat.

Automaten - Allgemein

- Gegeben:
  - **endliches Eingabealphabet**: $\mathbf{E = \{ E_1, E_2, \dots , E_g, \dots , E_u \}}$
  - **endliches Ausgabealphabet**: $\mathbf{A = \{ A_1, A_2, \dots , A_h, \dots , A_v \}}$
- Ein **Automat** ist eine **Einheit AT**, bei der eine **zeitliche Folge** von **Elementen** des **Eingabealphabets** F_E in eine **zeitliche Folge** von **Elementen** des **Ausgabealphabets** F_A abgebildet wird:

$$F_E \Rightarrow \boxed{\text{AT}} \Rightarrow F_A$$

- Zur **Unterscheidung zeitlicher Reihung:** Einführung eines **Ordnungsindizes** ν**:**

$$F_E = E_g^{\nu} \quad E_g^{\nu-1} \quad E_g^{\nu-2} \quad \dots \qquad\qquad F_A = A_h^{\nu} \; A_h^{\nu-1} \quad A_h^{\nu-2} \quad \dots$$

Beispiel: Getränkeautomat

- Eingabealphabet:
  E = { m (Euro einwerfen),
  w_1 (Sorte 1 wählen),
  w_2 (Sorte 2 wählen),
  r (Geldrückgabe anfordern)}
- Ausgabealphabet:
  A = { g (Geld zurück),
  k (keine Reaktion),
  a_1 (Sorte 1 ausgeben),
  a_2 (Sorte 2 ausgeben)}

Automaten - Allgemein

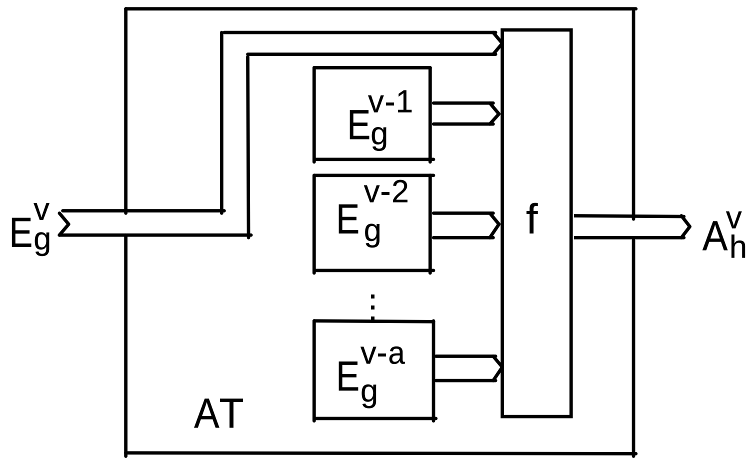
- Für die neue **Einheit AT** muss gelten:

$$\mathbf{A_h^{\nu} = f\,(\,E_g^{\nu},\, E_g^{\nu-1},\, E_g^{\nu-2},\, \ldots\,,\, E_g^{\nu-\alpha}\,)}$$

  - insgesamt werden α+1 Elemente des Eingabealphabets beobachtet
  - α ist die **zeitliche Tiefe** der **Verarbeitung**

Automaten - Allgemein

- **Nachteil dieses Konzeptes:** **Speicherung** von α **Eingabeelementen** erforderlich, um A_h^ν **zu berechnen**:



- **Erhalt eines neuen Elements** -> ältestes Element $E_g^{\nu-\alpha}$ entfällt aus Folge mit Länge (α+1)

Automaten - Allgemein

- Bei **Normierung** der **Indizes** dem **Alter nach** erhält man:

ν: $E_g^{\nu} E_g^{\nu-1} E_g^{\nu-2} ..... E_g^{\nu-\alpha+1} E_g^{\nu-\alpha}$

$\nu+1$: $E_g^{\nu+1} E_g^{\nu} E_g^{\nu-1} E_g^{\nu-2} ..... E_g^{\nu-\alpha+1} E_g^{\nu-\alpha}$

$\nu+1$: $E_g^{\nu+1} E_g^{\nu} E_g^{\nu-1} E_g^{\nu-2} ..... E_g^{\nu-\alpha+1} \cancel{E_g^{\nu-\alpha}}$ **reduziert**

$\nu+1 \Rightarrow \nu$: $E_g^{\nu} E_g^{\nu-1} E_g^{\nu-2} ..... E_g^{\nu-\alpha+1} E_g^{\nu-\alpha}$ **normiert**

Automaten - Allgemein

- **Weiterhin:** **Abbildung** der **Folge** $E_g^{\nu-1}$ ... $E_g^{\nu-\alpha}$ auf ein **neues Alphabet** $S = \{ S_1, \dots , S_k, \dots , S_w \}$ mit **geringerer Mächtigkeit**:

$$(E_g^{\nu-1} \dots E_g^{\nu-\alpha}) \rightarrow S, \quad \text{mit } |E|^{\alpha} \geq |S|$$

- Das **Element** S_k wird **Zustand** des Automaten **AT** genannt
- Unter **Beachtung** der **zeitlichen Reihung** erhält man nun:

$$A_h^{\nu} = \lambda (E_g^{\nu}, S_k^{\nu})$$

mit λ als ***Ausgabefunktion*** des Automaten AT

Automaten - Allgemein

- Bei Eingabe eines **neuen Eingabeelements E_g^{ν}** ergibt sich unter Berücksichtigung des **aktuellen Zustands** (sogenante “Historie der Eingaben“):

$$S_k^{\nu+1} = \delta (E_g^{\nu}, S_k^{\nu})$$

mit δ **als Überführungsfunktion** der Zustände

- Kennzeichnung der **Einheit AT als Quintupel**

$$AT = (E, A, S, \delta, \lambda)$$

mit **drei endlichen Mengen** und **zwei Abbildungen** zwischen diesen Mengen

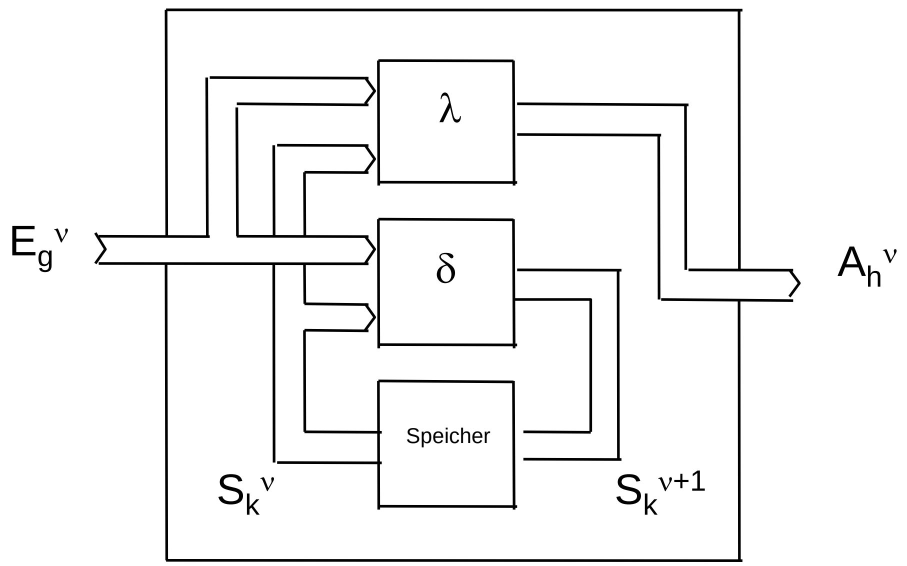
Beispiel: Getränkeautomat

- Zustandsmenge:
  $S = \{s_1, s_2\}$
- Darstellung der Überführungs- und Ausgabefunktion durch eine Tabelle:

δ / λ	m	r	w_1	w_2
s_1	s_2 / k	s_1 / k	s_1 / k	s_1 / k
s_2	s_2 / k	s_1 / g	s_1 / a_1	s_1 / a_2

Automaten - Allgemein

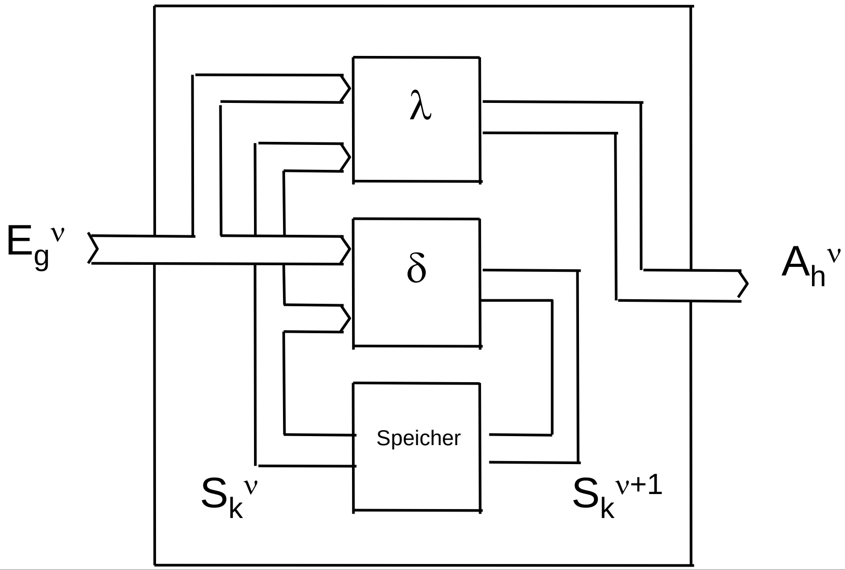
- Darstellung eines **Automaten** durch folgende **Struktur**:



  - **rekursive Anordnung** für die **Bestimmung** des **neuen Zustandes**
  - der **Speicher** nimmt den momentanen **Zustand S_k^ν** auf
  - der **neue Zustand $S_k^{\nu+1}$** muss zum richtigen **Zeitpunkt übernommen** werden

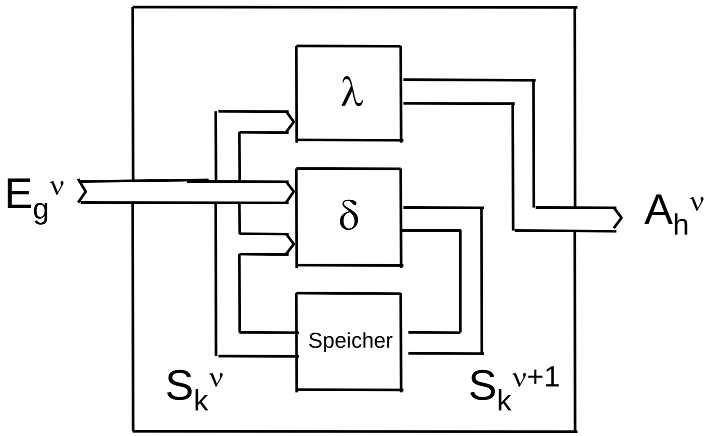
Automatentypen

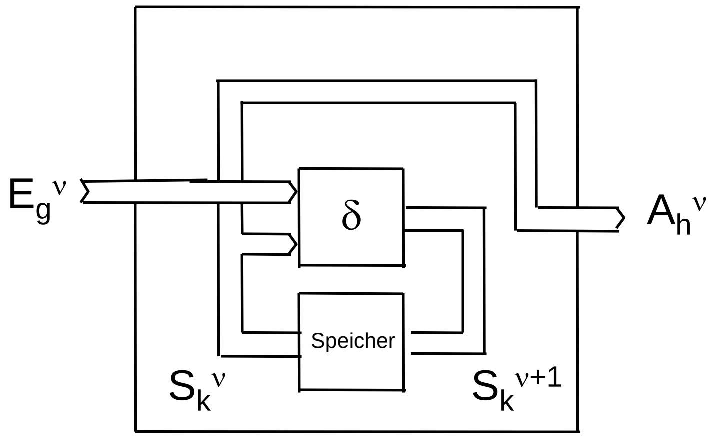
- **Wichtige Typklassen** von **Automaten** im Zusammenhang mit Digitalschaltungen: ***endliche***, ***diskrete*** und ***deterministische*** **Automaten**
- Bezüglich der **Ausgabefunktion** von E_g^ν und S_k^ν unterscheidet man **drei Fälle**:
  - **Mealy-Automat** mit $A_h^\nu = \lambda (E_g^\nu, S_k^\nu)$ als **allgemeinster** Fall:



Automatentypen

- **Moore-Automat** mit $A_h^\nu = \lambda (S_k^\nu)$ als ein **Spezialfall**, bei dem die **Ausgabe** allein vom **Zustand abhängt**
- **Medwedew-Automat** mit $A_h^\nu = S_k^\nu$ als dem **Spezialfall**, bei dem als **Ausgabe** der **Zustand selbst** dient

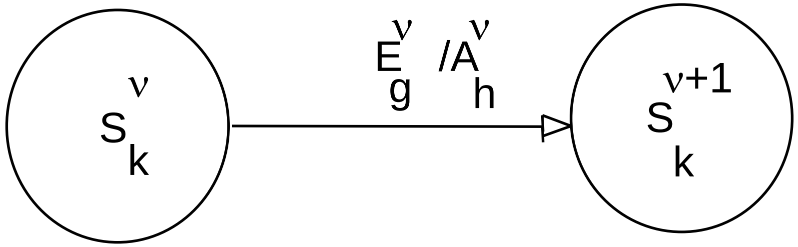




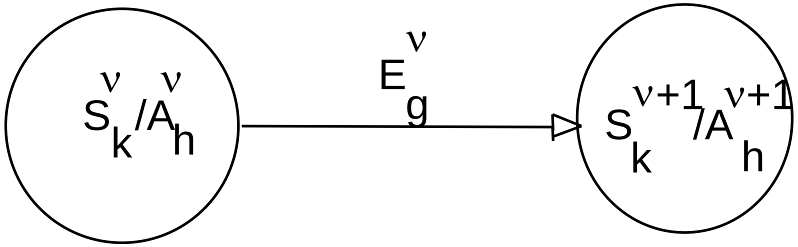
Automatengraphen

- **Knoten** repräsentieren **Zustände**, **Kanten** als **Zustandsübergänge**
- Beide **Elemente** des **Graphen** werden mit **Attributen** versehen, die sich auf **Eingangs-** und **Ausgangselemente** beziehen:

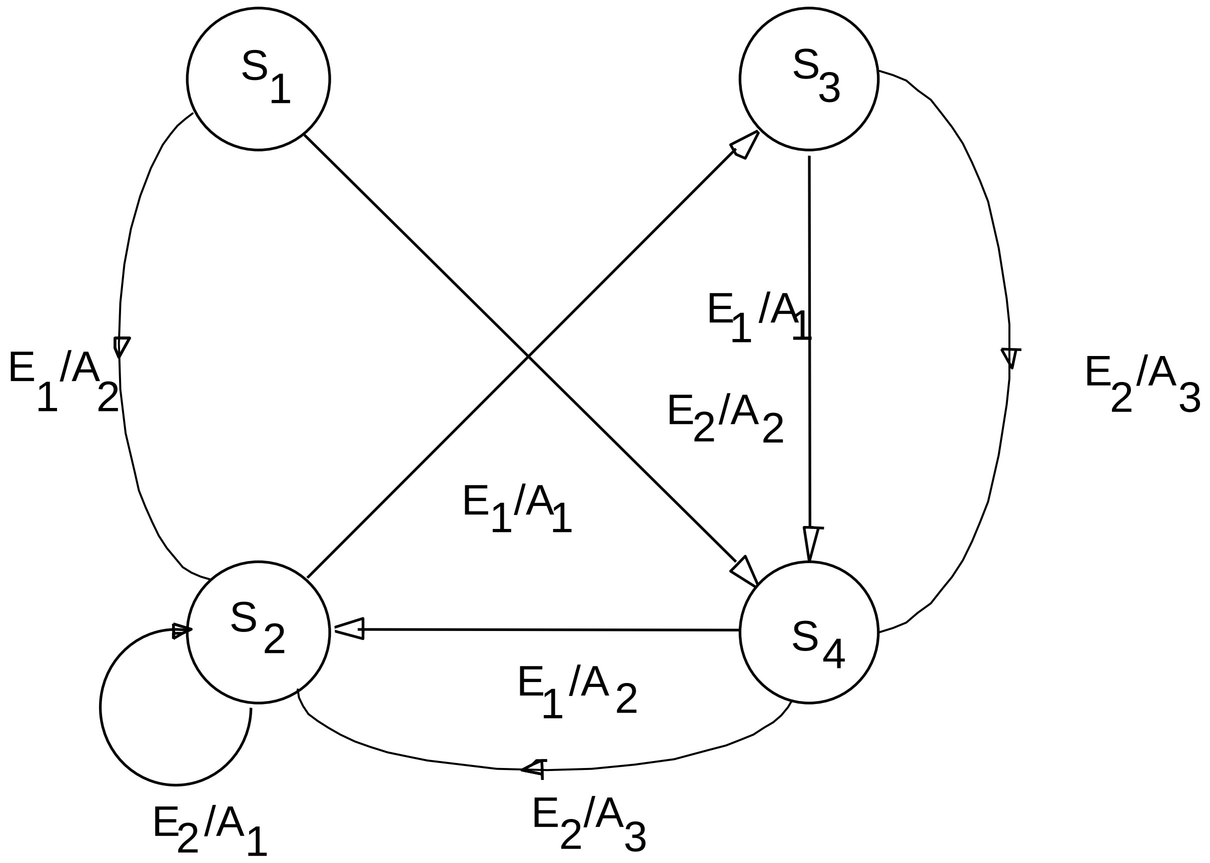
Mealy



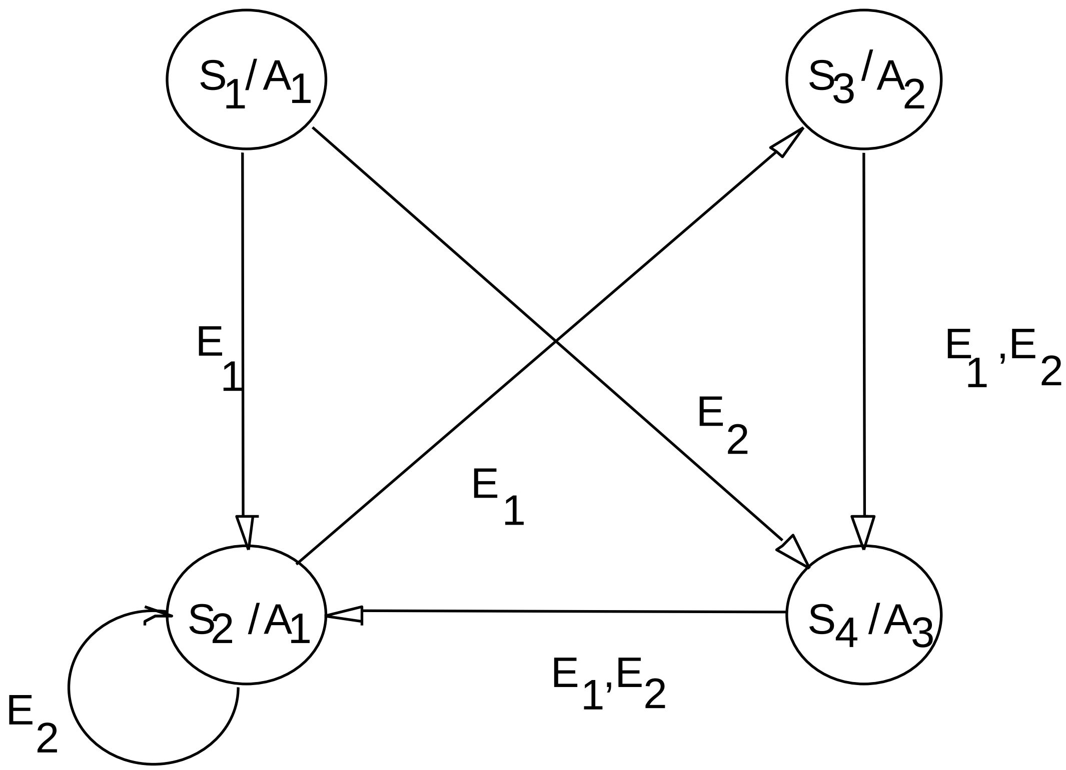
Moore



Beispiel Mealy-Automat



Beispiel Moore-Automat



Automatentafeln

- **Bilden** des **kartesischen Produktes** aus **Eingabe-** und **Zustandsmenge**
- An **Kreuzungsstellen** werden beim **Mealy-Automaten** der jeweilige **Folgezustand** und die **Ausgabe**, beim **Moore-Automaten** nur der **Folgezustand** eingetragen

Mealy

S^ν	E_1^ν	...	E_n^ν
		$S^{\nu+1}/A^\nu$	
...		...	
S^ν	...	$S_k^{\nu+1}/A_h^\nu$	...
...		...	

Moore

S^ν	E_1^ν	...	E_n^ν	A^ν
		$S^{\nu+1}$		
...		...		
S^ν	...	$S_k^{\nu+1}$	...	A_h^ν
...		...		

Automatentafeln

- Beispiele:

S^{ν}	$S^{\nu+1}/A^{\nu}$	
	E_1	E_2
S_1	S_2/A_2	S_4/A_2
S_2	S_3/A_1	S_2/A_1
S_3	S_4/A_1	S_4/A_3
S_4	S_2/A_2	S_2/A_3

S^{ν}	$S^{\nu+1}$		A^{ν}
	E_1	E_2	
S_1	S_2	S_4	A_1
S_2	S_3	S_2	A_1
S_3	S_4	S_4	A_2
S_4	S_2	S_2	A_3

Schaltwerke

- Die **technische Realisierung** eines endlichen, diskreten und deterministischen **Automaten** wird **Schaltwerk** genannt (im Gegensatz zum “gedächtnislosen“ Schaltnetz)
- Unterscheidung zwischen **Mealy**-, **Moore**- und **Medwedew-Schaltwerken** beim **Übergang** vom **Automaten** zum **Schaltwerk** ist eine **eineindeutige Abbildung** der **Automatenelemente** in **binäre Größen** (Bit-Leitungen) erforderlich:

$$\mathbf{E \leftrightarrow \{ X_j \} \quad \text{mit} \quad |E| \leq 2^n \quad \text{bei} \quad X = (x_n, \ldots , x_1)}$$

$$\mathbf{A \leftrightarrow \{ Y_i \} \quad \text{mit} \quad |A| \leq 2^m \quad \text{bei} \quad Y = (y_m, \ldots , y_1)}$$

Schaltwerke

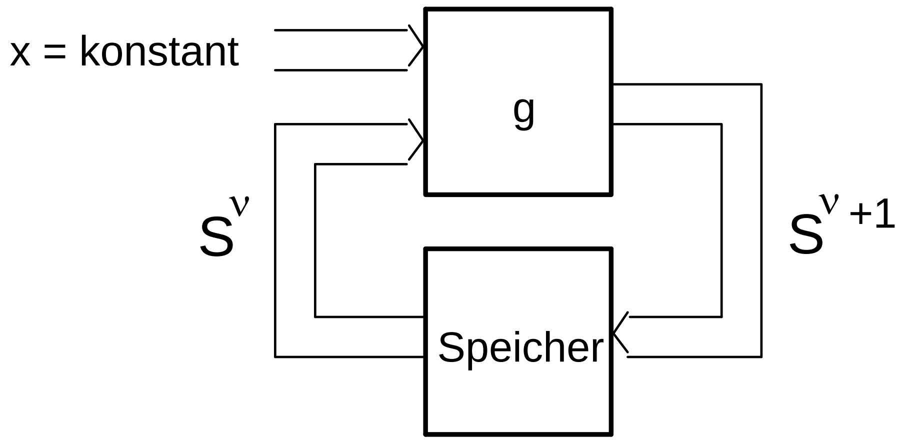
- **Neu:** **Eineindeutige Abbildung** der **Zustände S_k** der Menge S in die Menge der **Belegungen** eines weiteren **Binärvektors**, des **Zustandvektors Q**:

  $$S \leftrightarrow \{ Q_k \} \quad \text{mit} \quad |S| \leq 2^r \qquad \text{bei} \quad Q = (q_r, \ldots , q_1)$$

- ***Zustandskodierung***, λ und δ werden als **Schaltnetze** abgebildet
- **Spezifikation von Schaltwerken:**
  Automatengraphen und **Automatentafeln**
  -> entsprechende Automatenelemente werden durch ihre binäre Äquivalente ersetzt

Schaltwerke

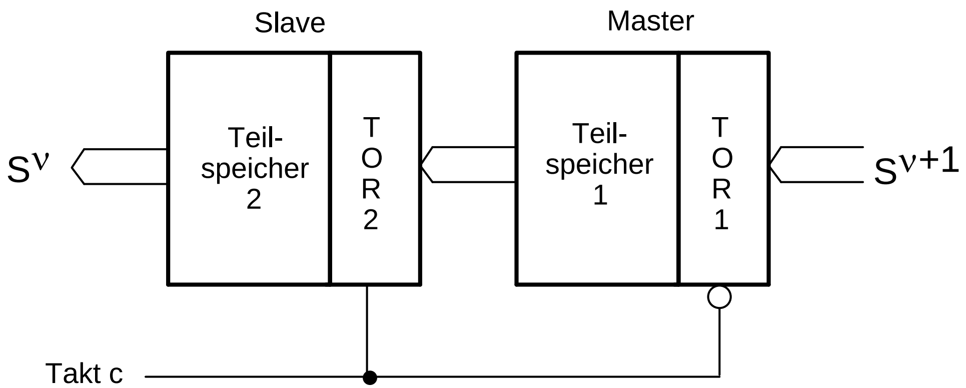
- Weiterhin: **Formal** eingeführte **Indizierung ν** der **zeitlichen Ordnung** muss nun in **technische Realisierung** überführt werden: **Rückführung** der **Zustände** über einen **Speicher**



- **Schaltwerke**, bei denen **Rückführungen direkt wirksam** werden, heißen **asynchron**.

Schaltwerke

- **Als zusätzliches informationsfreies Binärsignal wird der sogenannte *Takt c* benötigt**
- **Auftrennung** und **partielle Weiterleitung** des **Zustandes** bzw. **Folgezustandes** $\rightarrow$ **Zustandsspeicher** in **zwei Teilspeicher** zerlegen



- **Funktion:**

  c = 0: (Teilspeicher 1) = $S^{\nu+1}$

  c = 1: (Teilspeicher 2) = (Teilspeicher 1)

  immer: S^{ν} = (Teilspeicher 2)

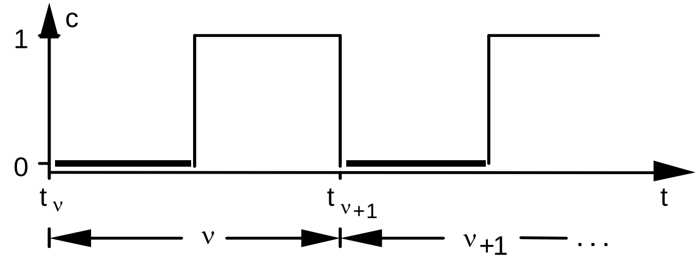
  Tor 1 und Tor 2 sind nie gleichzeitig aktiv

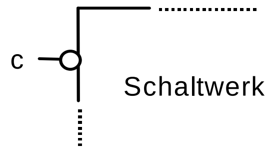
Schaltwerke

- Schaltwerke mit taktgesteuerten Speichern werden ***getaktet betrieben*** genannt
- Ist der **Takt** an allen **Schaltungsteilen gleichzeitig** aktiv, spricht man von einem ***synchronen*** **Schaltwerk**
- **Mealy-Automat:** **Änderung** der **Eingabe** bewirkt **asynchrone Ausgabenberechnung.**
- Die **Zählung** des **Indexes** ν muss aus dem **Taktsignal abgeleitet** werden

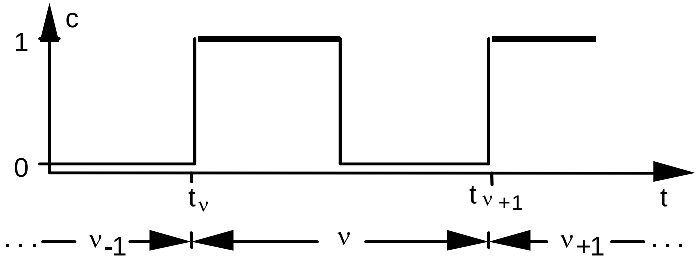
Taktsteuerungsarten

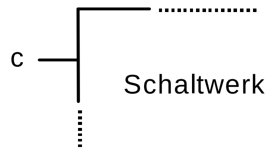
pegelgesteuert:



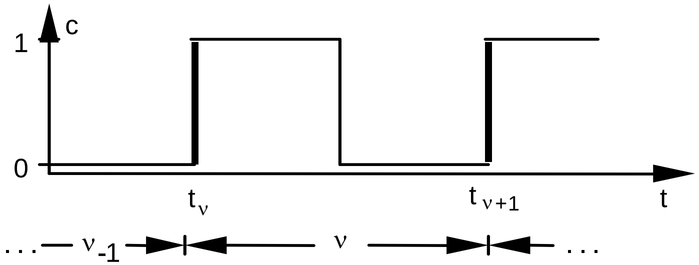


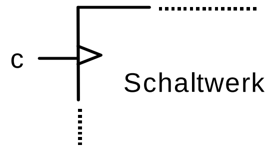
pegelgesteuert:



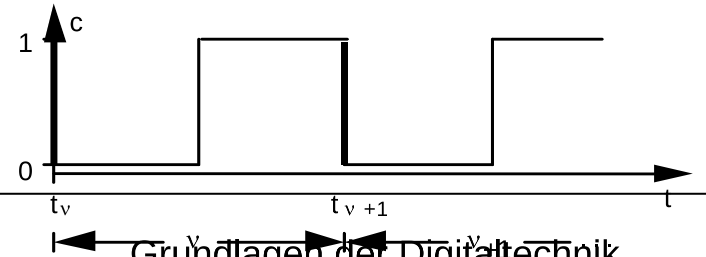


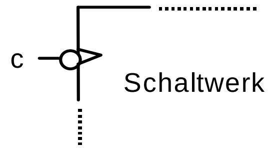
flankengesteuert:





flankengesteuert:





Binärspeicher (Flipflops) - Wiederholung

- **Binärspeicher** werden zur **Speicherung** von **Zuständen** der **Schaltwerke** benötigt
- Ist $|S|$ die **Zahl** der **Zustände**, berechnet sich die **Größe** des **Binärvektors Q** zu $\mathbf{r = \lceil ld |S| \rceil}$ mit $\mathbf{Q = (q_r, \ldots , q_k, \ldots , q_1)}$ mit $\mathbf{q_k}$ einer beliebigen **Zustandsvariable** des **Zustandsvektors**
  - einzelne **Zustandsvariablen** $\mathbf{q_k}$ werden **rekursiv** mit δ **berechnet**
  - eine **zwei Werte speichernde Einheit** wird als **Binärspeicher** bezeichnet: ***FlipFlop***
  - ein **Speicher** mit **r *FlipFlops*** kann $\mathbf{2^r}$ **unterschiedliche Zustände** speichern

Binärspeicher (Flipflops) - Wiederholung

- **Grundsätzliche Funktionen eines Binärspeichers:**
  - muss wahlweise eine **0 oder 1 eingespeichern** können (***schreiben***)
  - der **Wert im Speicher** muss extern **zur Verfügung** stehen (***lesen***)
  - **Vorhandensein eines Taktes**: **Rekursion (ν+1 => ν)** muss vom **Binärspeicher** unter **Einwirkung** des **Taktes** verwirklicht werden
- **Wir kennen: verschiedene Varianten** der **Binärspeicher mit verschiedenen Ansteuerfunktionen**

Flipflop-Varianten (Wiederholung)

Fall 1: RS-FlipFlop

Lesesignal $\triangleq$ Zustandswert:

$\mathbf{q_k} \in \mathbf{\{0,1\}}$

Schreiben einer 0, ***Rücksetzen:***

$\mathbf{R} \in \mathbf{\{0,1\}}$ 0 $\triangleq$ inaktiv
1 $\triangleq$ Schreiben 0

Schreiben einer 1, ***Setzen:***

$\mathbf{S} \in \mathbf{\{0,1\}}$ 0 $\triangleq$ inaktiv
1 $\triangleq$ Schreiben 1

Fall 2: D-FlipFlop

Lesesignal $\triangleq$ Zustandswert:

$\mathbf{q_k} \in \mathbf{\{0,1\}}$

Datensignal $\triangleq$ Zustandswert neu:

$\mathbf{D} \in \mathbf{\{0,1\}}$

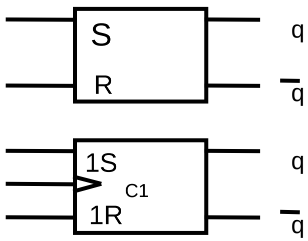
Schreiben des **Wertes** von **D**:

$\mathbf{W} \in \mathbf{\{0,1\}}$ 0 $\triangleq$ inaktiv
1 $\triangleq$ Schreiben

Das **gleichzeitige Schreiben** von **0** und **1** (R=1, S=1) beim **RS-FlipFlop** ist **unzulässig**

FlipFlops (Wiederholung)

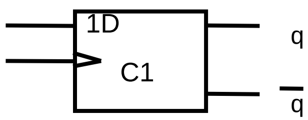
Symbole | **Charakteristische Gleichungen** | **Ansteuerfunktionen**



RS-FlipFlop

$$q^{\nu+1} = S \vee (q^{\nu} \,\&\, \overline{R})$$

q^{ν}	$q^{\nu+1}$	R	S
0	0	-	0
0	1	0	1
1	0	1	0
1	1	0	-



D-FlipFlop

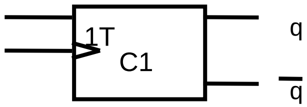
$$q^{\nu+1} = D$$

q^{ν}	$q^{\nu+1}$	D
0	0	0
0	1	1
1	0	0
1	1	1

FlipFlops (Wiederholung)

Symbole | **Charakteristische Gleichungen** | **Ansteuerfunktionen**

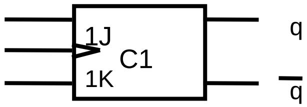
T-FlipFlop



$$q^{\nu+1} = (T \,\&\, \overline{q}^{\nu}) \vee (\overline{T} \,\&\, q^{\nu})$$

q^{ν}	$q^{\nu+1}$	T
0	0	0
0	1	1
1	0	1
1	1	0

JK-FlipFlop



$$q^{\nu+1} = (\overline{K} \,\&\, q^{\nu}) \vee (J \,\&\, \overline{q}^{\nu})$$

q^{ν}	$q^{\nu+1}$	J	K
0	0	0	-
0	1	1	-
1	0	-	1
1	1	-	0

Entwurf von Schaltwerken

- **Vorgehensweise:**
  - **1. Schritt:** **Definition und Kodierung** der **Ein- und Ausgabevariablen**
  - **2. Schritt:** Wahl des **Schaltwerktyps** und Erstellen der **Automatentafel** oder **Zustandsgraph** gemäß Aufgabenstellung
  - **3. Schritt:** **Zustandskodierung**
  - **4. Schritt:** Wahl des **FlipFlop-Typs** und Aufstellen der **Ansteuerfunktionen**
  - **5. Schritt:** Entwurf des **Schaltnetzes** für die **Überführungs-funktion** auf der Basis der **Ansteuerfunktionen**
  - **6. Schritt:** Entwurf des **Schaltnetzes** für die **Ausgabefunktion**
  - **7. Schritt:** Eventuell **Umformung** der **logischen Ausdrücke** in geeignete **Strukturausdrücke**
  - **8. Schritt:** Umsetzen in das **Schaltbild** des **Schaltwerkes**

Entwurf von Schaltwerken

- **1. Schritt:**
- **Eingabeereignisse**:
  - **Münze eingeworfen**: **m**
  - **Geldrückgabe**: **r**
  - **Getränkewahl 1** w_1
  - **Getränkewahl 2** w_2
- **Ausgabeereignisse**:
  - **keine Aktion** **k**
  - **Geldauswurf** **g**
  - **Getränkeausgabe 1** a_1
  - **Getränkeausgabe 2** a_2

Entwurf von Schaltwerken

- Definition und Kodierung der Eingaben:
  $|E| = 4 = 2^2$. Deshalb sind n=2 Variablen notwendig:
  $X = (x_2, x_1)$.

	x_2	x_1
m	0	0
r	0	1
w_1	1	0
w_2	1	1

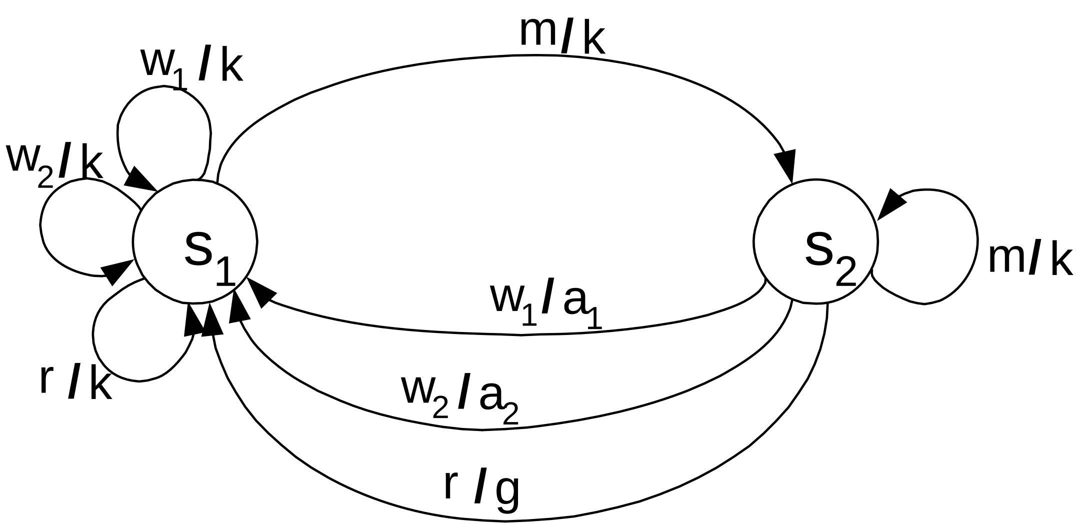
- Definition und Kodierung der Ausgaben:
  $|A| = 4 = 2^2$. Deshalb sind m=2 Variablen notwendig:
  $Y = (y_2, y_1)$.

	y_2	y_1
k	0	0
g	0	1
a_1	1	0
a_2	1	1

Entwurf von Schaltwerken

- **2. Schritt:**
- **Wahl des Schaltwerktyps:**
  - **Mealy**

    (Der Ausgang hängt vom aktuellen Zustand und von den Eingängen ab)



Entwurf von Schaltwerken

- **3. Schritt:**
- **Zustandskodierung:**
  - $|S| = 2 = 2^1$ Zustände. Deshalb ist $r = 1$ Zustandsvariable notwendig: $Q = (q)$.
  - Kodierung der beiden Zustände s_1 und s_2:

	q
s_1	0
s_2	1

Entwurf von Schaltwerken

- **4. Schritt:**
- **FlipFlop-Typ:**
  - D-FlipFlop
- **Automatentafel mit kodierten Ein-, Ausgaben und Zuständen:**

$X=(x_2, x_1)$ $\quad Y=(y_2, y_1)$

	δ/λ	m (0, 0)	r (0, 1)	w_1 (1, 0)	w_2 (1, 1)
$Q=(q)$	(s_1) 0	1/(0,0)	0/(0,0)	0/(0,0)	0/(0,0)
	(s_2) 1	1/(0,0)	0/(0,1)	0/(1,0)	0/(1,1)

Entwurf von Schaltwerken

- **Ansteuerfunktion des D-Flipflops:**

δ	m (0, 0)	r (0, 1)	w_1 (1, 0)	w_2 (1, 1)
$Q=(q)$ (s_1) 0	1	0	0	0
$Q=(q)$ (s_2) 1	1	0	0	0

$$D = \overline{x}_2 \cdot \overline{x}_1 \cdot \overline{q} + \overline{x}_2 \cdot \overline{x}_1 \cdot q$$

- **Ausgabefunktion**

λ	m (0, 0)	r (0, 1)	w_1 (1, 0)	w_2 (1, 1)
$Q=(q)$ (s_1) 0	(0,0)	(0,0)	(0,0)	(0,0)
$Q=(q)$ (s_2) 1	(0,0)	(0,1)	(1,0)	(1,1)

$$y_2 = x_2 \cdot \overline{x}_1 \cdot q + x_2 \cdot x_1 \cdot q$$

$$y_1 = \overline{x}_2 \cdot x_1 \cdot q + x_2 \cdot x_1 \cdot q$$

Entwurf von Schaltwerken

- **<u>5. Schritt:</u>**
- **KV-Diagramm für Überführungsfunktion δ:**

D:		x_2	x_2	
	1	0	0	1
x_1	0	0	0	0

$$\Rightarrow D = \overline{x}_1 \cdot \overline{x}_2$$

Entwurf von Schaltwerken

- <u>**6. Schritt:**</u>
- **Entwurf der Schaltnetze für die Ausgabefunktion(en) λ:**

y_2:

		x_2	x_2	
	0	0	1	0
x_1	0	0	1	0
			q	q

$$\Rightarrow y_2 = q \cdot x_2$$

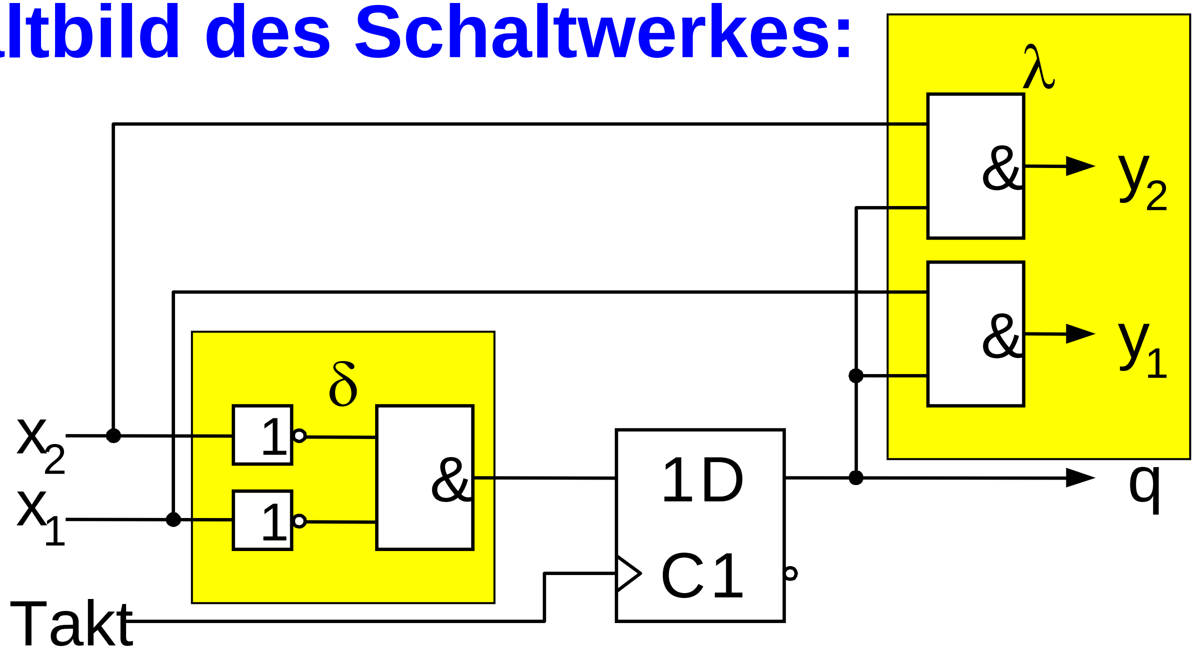
y_1:

		x_2	x_2	
	0	0	0	0
x_1	0	0	1	1
			q	q

$$\Rightarrow y_1 = q \cdot x_1$$

Entwurf von Schaltwerken

- **7. Schritt:** entfällt.
- **8. Schritt:**
- **Schaltbild des Schaltwerkes:**



Achtung: Der Takt muss so gestaltet sein, dass eine Vorderflanke nur im Falle einer Eingabeänderung erzeugt wird!

Spezielle Schaltwerke

- <u>Zähler</u>
  - **Zähler-Schaltwerke** dienen dazu, auf einer speziellen Signalleitung beobachtete **Ereignisse** zu **zählen**
  - Das Zählen entspricht einer **Abbildung** in den **Zustandsraum**, so dass ein **Zählcode** entsteht
  - Wie bei mechanischen Zählern können noch **weitere Eingriffmöglichkeiten** eingebaut werden:
    - Überlaufmeldung
    - Nullstellungsmeldung
    - Rückstellmöglichkeit auf 0

Dualzähler

- Beispiel:
  - Es soll ein Zähler als Schaltwerk realisiert werden, der...
    - **dual zyklisch** von 0 bis 7 zählt
    - eine **Rückstellmöglichkeit** auf 0 besitzt
    - bei dem Wert 7 das **Überlaufsignal** auf 1 setzt
    - auf den **Wechsel** der **Zählervariablen Z** von **0 auf 1** reagiert
  - Zur **Realisierung** sollen **JK-Flipflops** mit **0/1-Flankensteuerung** verwendet werden

Dualzähler

- Beispiel:
  - **Eingangsvariable**:

    N = 1: Rückstellen auf 0

    N = 0: sonst
  - **Ausgangsvariable**:

    Ü = 1: Zählerstand ist 7

    Ü = 0: sonst

  - Bei Zählern wird die **Zählervariable Z** häufig nicht als Eingangsvariable, sondern direkt **als Taktsignal** angeschlossen
  - Das **Zählereignis** „Wechsel von 0 auf 1“ kann so direkt mit einem **synchronen Schaltwerk** mit **positiver Flankensteuerung** realisiert werden

Dualzähler

- <u>Schaltwerkstabelle:</u>
  - Die **Kodierung** der **Zustände** ist die **duale Zahlendarstellung**
  - Bei **N = 1** ist der **Folgezustand** auf jeden Fall **0** (Rücksetzen)
  - Die **Übertragungsfunktion** kann direkt abgelesen werden:

$$Ü = q_3^{\nu} \,\&\, q_2^{\nu} \,\&\, q_1^{\nu}$$

	Q^{υ}		X^{υ}		$Q^{\upsilon+1}$		Y^{υ}
q_3	q_2	q_1	N	q_3	q_2	q_1	Ü
0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0
0	0	1	0	0	1	0	0
0	0	1	1	0	0	0	0
0	1	0	0	0	1	1	0
0	1	0	1	0	0	0	0
0	1	1	0	1	0	0	0
0	1	1	1	0	0	0	0
1	0	0	0	1	0	1	0
1	0	0	1	0	0	0	0
1	0	1	0	1	1	0	0
1	0	1	1	0	0	0	0
1	1	0	0	1	1	1	0
1	1	0	1	0	0	0	0
1	1	1	0	0	0	0	1
1	1	1	1	0	0	0	1

Dualzähler

- <u>Ansteuerfunktionen:</u>
  - durch **Ablesen** oder mit Hilfe von ***Symmetrie-diagrammen*** erhält man:

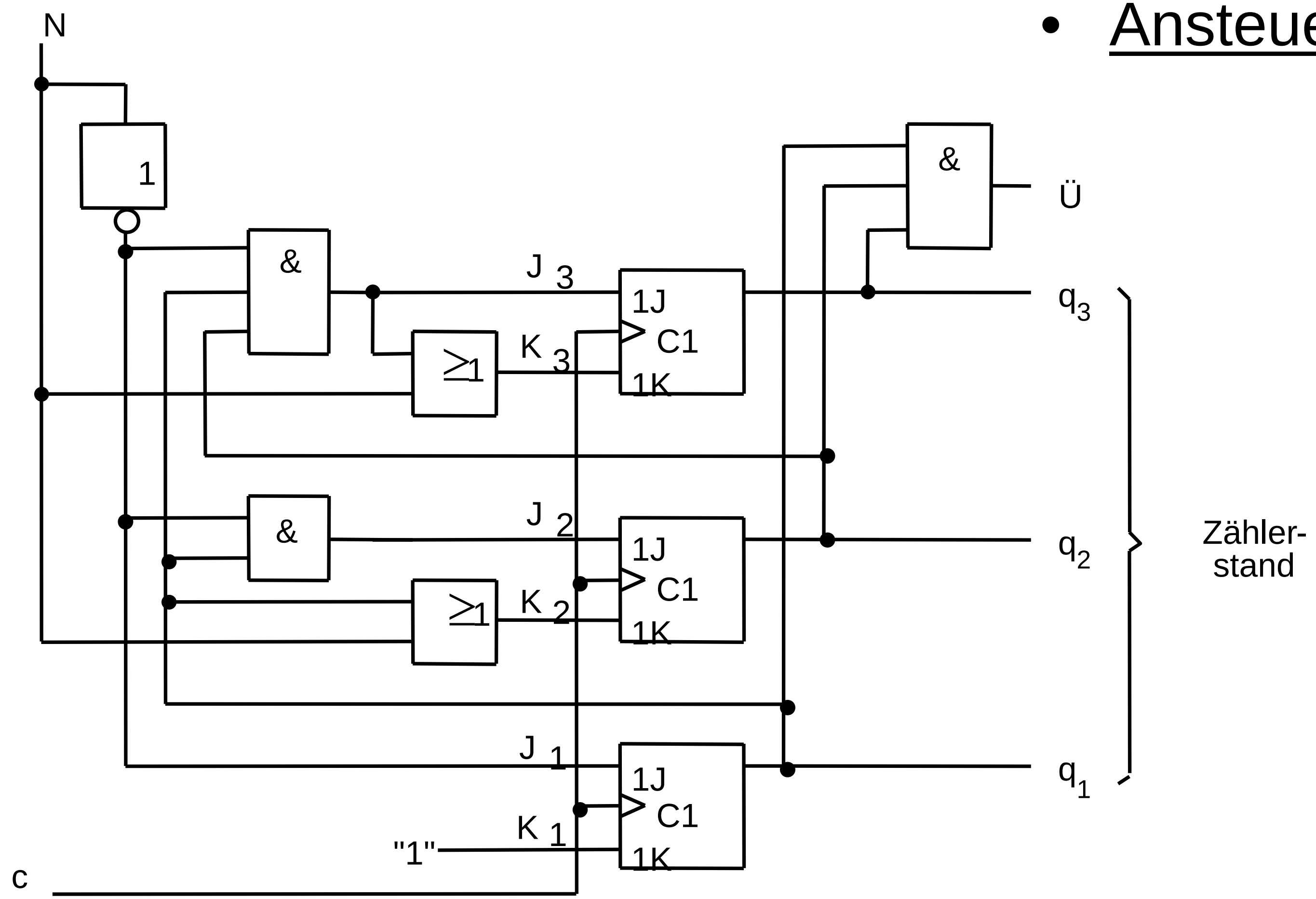
$$K_1 = 1$$
$$K_2 = N \vee q_1^\nu$$
$$K_3 = N \vee q_2^\nu q_1^\nu$$

$$J_1 = \overline{N}$$
$$J_2 = \overline{N} q_1^\nu$$
$$J_3 = \overline{N} q_2^\nu q_1^\nu$$

Q^υ			X^υ	$Q^{\upsilon+1}$			Y^υ						
q_3	q_2	q_1	N	q_3	q_2	q_1	Ü	J_3	K_3	J_2	K_2	J_1	K_1
0	0	0	0	0	0	1	0	0	-	0	-	1	-
0	0	0	1	0	0	0	0	0	-	0	-	0	-
0	0	1	0	0	1	0	0	0	-	1	-	-	1
0	0	1	1	0	0	0	0	0	-	0	-	-	1
0	1	0	0	0	1	1	0	0	-	-	0	1	-
0	1	0	1	0	0	0	0	0	-	-	1	0	-
0	1	1	0	1	0	0	0	1	-	-	1	-	1
0	1	1	1	0	0	0	0	0	-	-	1	-	1
1	0	0	0	1	0	1	0	-	0	0	-	1	-
1	0	0	1	0	0	0	0	-	1	0	-	0	-
1	0	1	0	1	1	0	0	-	0	1	-	-	1
1	0	1	1	0	0	0	0	-	1	0	-	-	1
1	1	0	0	1	1	1	0	-	0	-	0	1	-
1	1	0	1	0	0	0	0	-	1	-	1	0	-
1	1	1	0	0	0	0	1	-	1	-	1	-	1
1	1	1	1	0	0	0	1	-	1	-	1	-	1

Dualzähler

- Blockschema des Zählers:



- Ansteuerfunktionen:

$$J_1 = \overline{N}$$

$$J_2 = \overline{N} q_1^{\nu}$$

$$J_3 = \overline{N} q_2^{\nu} q_1^{\nu}$$

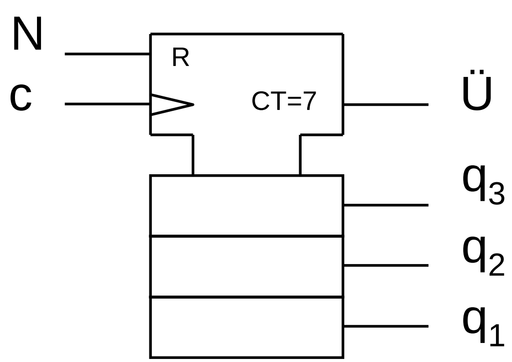
$$K_1 = 1$$

$$K_2 = N \vee q_1^{\nu}$$

$$K_3 = N \vee q_2^{\nu} q_1^{\nu}$$

Dualzähler

- Symboldarstellung:
  - Um bei Zählern von **Realisierungsdetails** zu **abstrahieren,** führt man **Blocksymbole** ein
  - Das **Symbol** besteht aus einem **Daten-** und einem **Steuerteil**
  - Für das besprochene Beispiel sieht der **Zähler** so aus:



Dualzähler

- Eigenschaften:
  - Hier noch die **wichtigsten Merkmale** von **Zählern**:
    - **Art des Zählcodes**
      -> Dualzahlen, 1-aus-n-Code, BCD-Code,...
    - **Art des Zählerreignisses**
      -> 0->1, 1->0, Auftreten von Werten, ...
    - **Steuersignal** für **Zählen/Anhalten**
    - **Steuersignal** für **Zählrichtung** (aufwärts/abwärts)
    - **Synchrone** oder **asynchrone Rückstellmöglichkeit**
    - **Steuersignal** für Zählen/Laden eines **Anfangswertes**