Karlsruher Institut für Technologie

Institut für Technik der Informationsverarbeitung

Prof. Jürgen Becker

becker@kit.edu

Karlsruher Institut für Technologie (**KIT**)

Institut für Technik der Informationsverarbeitung (ITIV)

Digitaltechnik

Bausteine der Digitaltechnik
- Automaten und Binärspeicher -

Automaten - Motivation

- **Schaltnetze:** ermöglichen die Bildung von Ausgangsgrößen unmittelbar aus Eingangsgrößen

- **maximal** sinnvolle **Größe** der **Schaltnetze** ist jedoch aus wirtschaftlichen und technischen Gründen **beschränkt**

  Notwendig: ein **neues Verarbeitungskonzept**, welches die **Mächtigkeit** des **Eingangsalphabets reduziert** und die **Komplexität** der **Ausgabegröße** in **andere Dimension** verlagert

- **Beispiel für Mächtigkeit / Komplexität:** Ausgabe von Automatenware

  - Bei **Einsatz** von **Schaltnetzen** zur Verarbeitung der Eingaben wäre hier eine **Eingabe aller Parameter** zur **gleichen Zeit notwendig**!!!

  - **reales Verhalten:** Eingabe der Reihe nach

- Im folgenden wird gezeigt, warum zur **Speicherung aller „vorherigen“ Eingaben** die Verwendung eines sogenannten “**Zustandes**“ sinnvoll ist

Automaten - Allgemein

Verallgemeinerung des Problems:

- endliches Eingabealphabet: $E = \{ E_1, E_2, \dots , E_g, \dots , E_u \}$
- endliches Ausgabealphabet: $A = \{ A_1, A_2, \dots , A_h, \dots , A_v \}$

➢ das **neue Konzept** beschrieben durch 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:

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

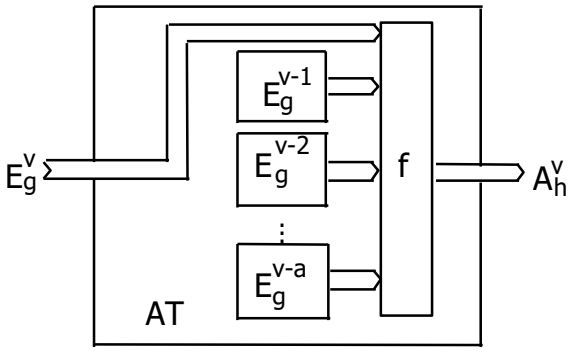
- zur **Unterscheidung zeitlicher Reihung:** Einführung eines **Ordnungsindizes ν**

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

- Für die neue **Einheit AT** muss gelten: $\mathbf{A_h^{\nu} = f (E_g^{\nu}, E_g^{\nu-1}, E_g^{\nu-2}, \dots , E_g^{\nu-\alpha})}$
  - insgesamt werden α+1 Elemente des Eingabealphabets beobachtet
  - α ist die **zeitliche Tiefe** der **Verarbeitung**

Automaten - Allgemein

Nachteil des Konzeptes: **Speicherung** von α **Eingabeelementen** erforderlich, um A_h^{ν} **zu berechnen**:



- **Erhalt neuen Elements** → älteste Element $E_g^{\nu-\alpha}$ entfällt aus Folge mit Länge (α+1)
  → bei **Normierung** der **Indizes** dem **Alter nach** erhält man:

v: $E_g^{v} E_g^{v-1} E_g^{v-2} \ldots\ldots E_g^{v-a+1} E_g^{v-a}$

v+1: $E_g^{v+1} E_g^{v} E_g^{v-1} E_g^{v-2} \ldots\ldots E_g^{v-a+1} E_g^{v-a}$

v+1: $E_g^{v+1} E_g^{v} E_g^{v-1} E_g^{v-2} \ldots\ldots E_g^{v-a+1} \cancel{E_g^{v-a}}$ **reduziert**

v+1 $\Rightarrow$ v: $E_g^{v} E_g^{v-1} E_g^{v-2} \ldots\ldots E_g^{v-a+1} E_g^{v-a}$ **normiert**

Automaten - Allgemein

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

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

- das **Element** S_k wird **Zustand** von **AT** genannt
- unter **Beachtung** der **zeitlichen Reihung** erhält man nun:

$$A_h^{\nu} = \lambda (E_g^{\nu}, S_k^{\nu}), \quad \text{mit } \lambda \text{ als } \textit{Ausgabefunktion} \text{ von AT}$$

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

$$S_k^{\nu+1} = \delta (E_g^{\nu}, S_k^{\nu}), \quad \text{mit } \delta \text{ als Überführungsfunktion der Zustände}$$

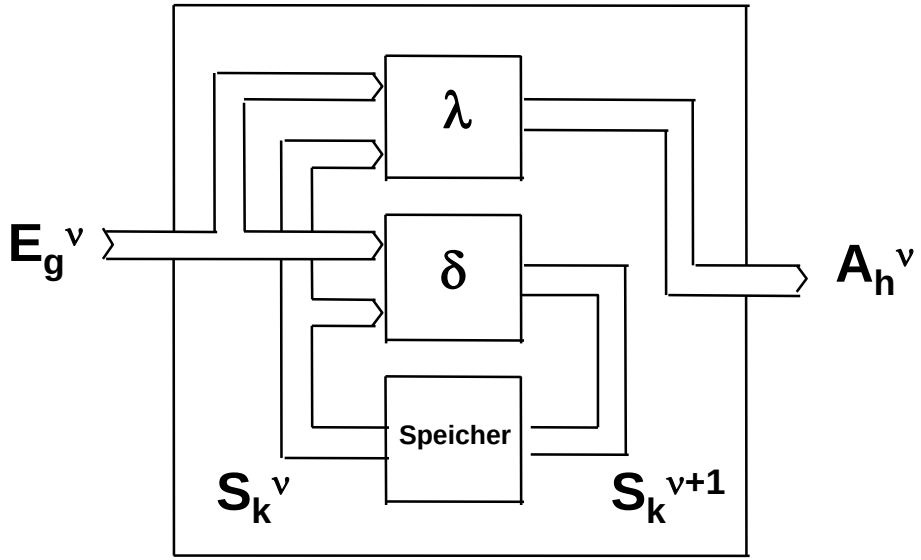
- Kennzeichnung der **Einheit AT als Quintupel**

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

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

Automaten- Allgemein

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



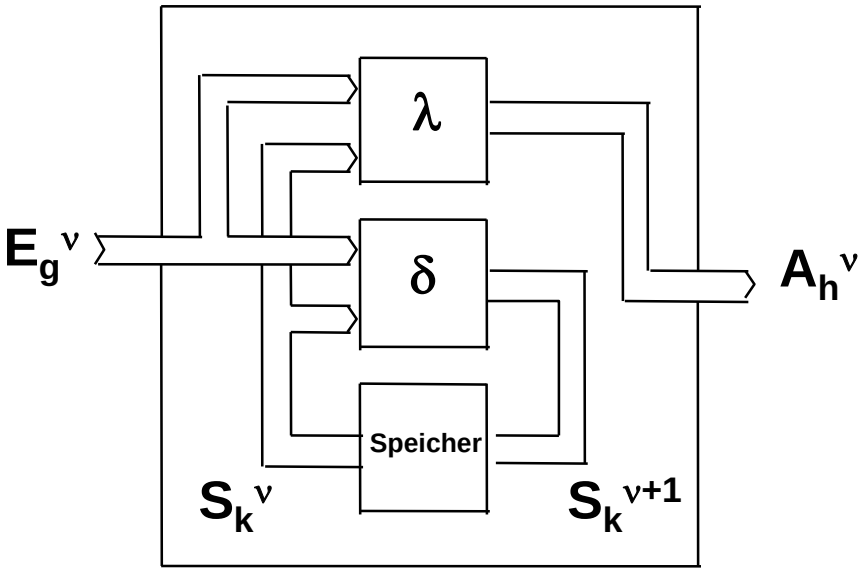
- **rekursive Anordnung** für die **Bestimmung** des **neuen Zustandes**
- der **Speicher** nimmt den momentanen **Zustand S_k^{ν}** auf
- der **neue Zustand $S_k^{\nu+1}$** muss zum richtigen **Zeitpunkt übernommen** werden

Automaten - Typen

Wichtige Typklassen von **Automaten** im Zusammenhang mit Digitalschaltungen:

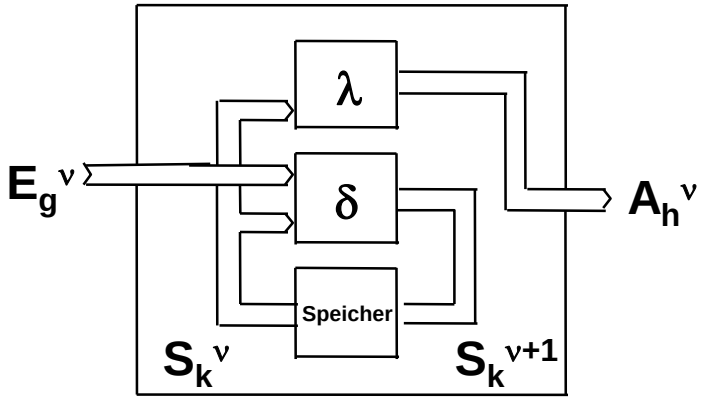
- ***endliche***, ***diskrete*** und ***deterministische*** **Automaten**
- Bezüglich der Ausgabefunktion von E_g^{ν} und S_k^{ν} unterscheidet man **drei Fälle**:

- **Mealy-Automat** mit $A_h^{\nu} = \lambda (E_g^{\nu}, S_k^{\nu})$ als allgemeinster Fall:

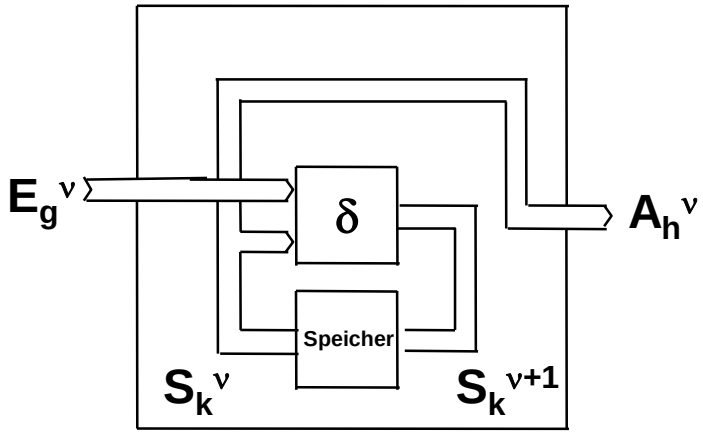


Automaten - Typen

- **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:



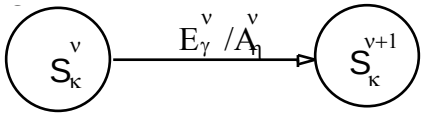
Automaten - Typen

Beschreibung des Verhaltens: **Automatengraphen** und **Automatentafeln**

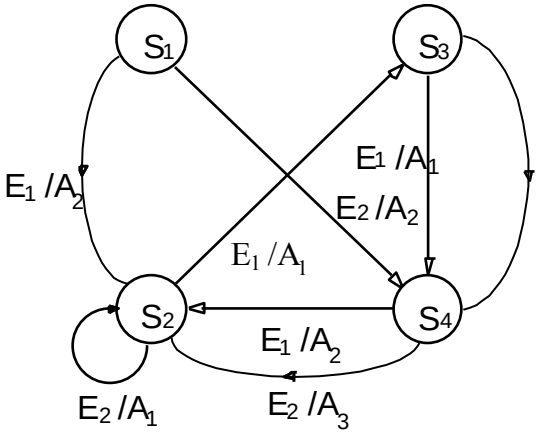
- **Automatengraphen:** **Knoten** repräsentieren **Zustände**, **Kanten** als **Zustandsübergänge**
- Beide **Elemente** des **Graphen** werden mit **Attributen** versehen, die sich auf **Eingangs-** und **Ausgangselemente** beziehen

Mealy

allgemein

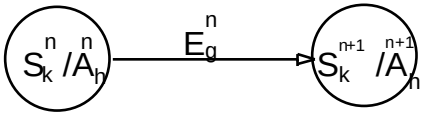


Beispiel:

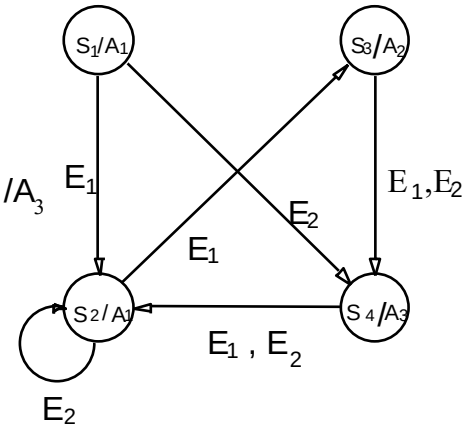


Moore

allgemein:



Beispiel:



Automaten - Typen

Automatentafel:

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

Moore

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

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:

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

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

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

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

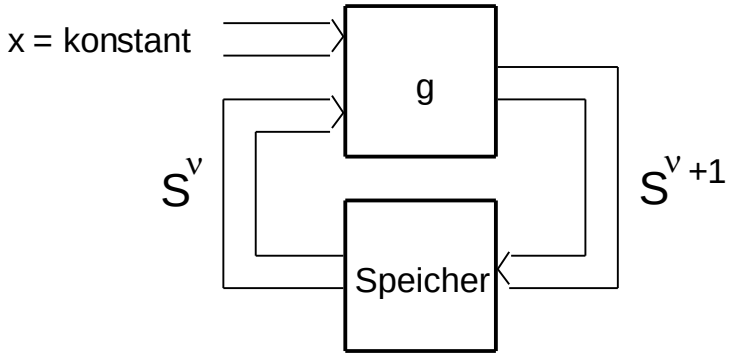
→ ***Zustandskodierung***, λ und δ werden als **Schaltnetze** abgebildet

- **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**:



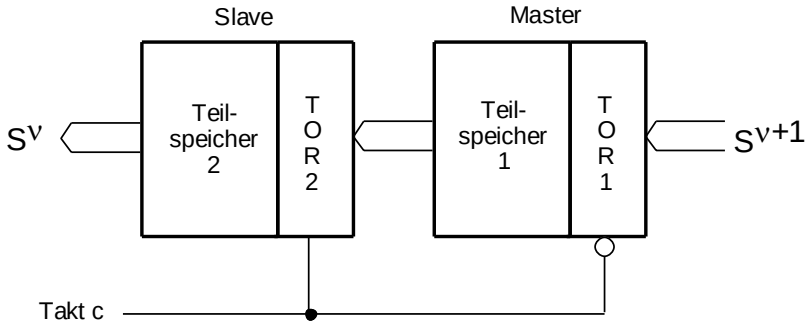
- **Schaltwerke**, bei denen **Rückführungen** direkt wirksam werden, heißen **asynchron** → sie sind schwieriger zu entwerfen

Also: ein **Speicher** wird benötigt, der die Schwierigkeit der unmittelbar wirksamen Rückkopplung vermeidet, indem nur zu **bestimmten Zeitpunkten** die am Speicher anliegenden **Werte übernommen** werden

Schaltwerke

Als zusätzliches informationsfreies Binärsignal, wird der sogenannte *Takt c* benötigt

- **Auftrennung** und **partielle Weiterleitung** des **Zustandes** bzw. **Folgezustandes**
  → **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 mit taktgesteuerten Speichern werden ***getaktet betrieben*** genannt
  → **Takte** an allen Schaltungsteilen **gleichzeitig** aktiv → ***synchrones* Schaltwerk**

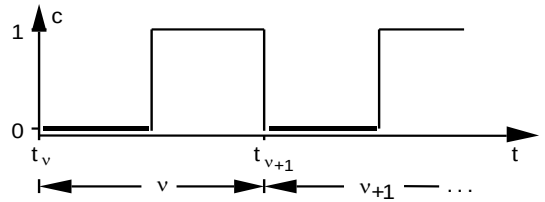
Schaltwerke

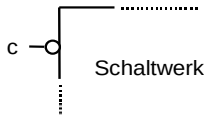
Mealy-Automat: Änderung der **Eingabe** bewirkt **asynchrone Ausgabenberechnung**

→ die **Zählung** des **Indexes ν** muss aus dem **Taktsignal abgeleitet** werden

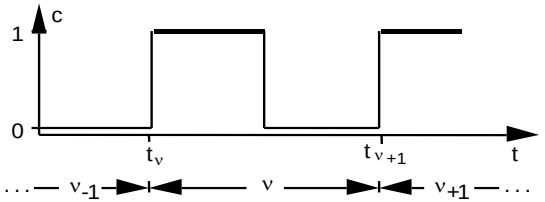
Steuerungsarten des Taktes:

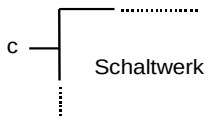
pegelgesteuert:



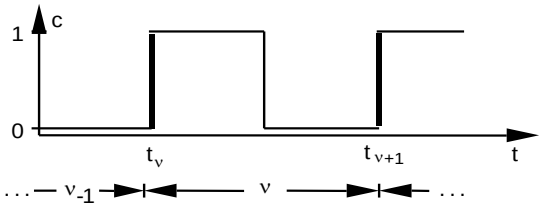


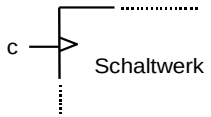
pegelgesteuert:



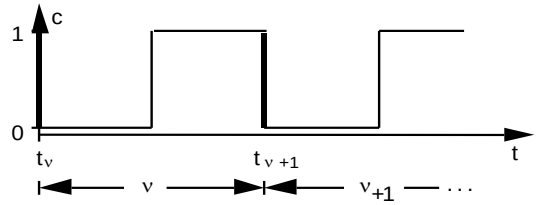


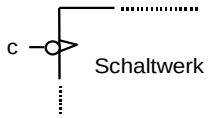
flankengesteuert:





flankengesteuert:





Binärspeicher (FlipFlops)

Binärspeicher werden zur Speicherung von **Zuständen** der **Schaltwerke** benötigt

→ Ist $|S|$ die Zahl der **Zustände**, berechnet sich die Größe des **Binärvektors Q** zu:

$$r = \lceil \mathrm{ld} |S| \rceil \qquad \text{mit} \quad Q = (q_r, \ldots , q_k, \ldots , q_1)$$

mit q_k einer beliebigen **Zustandsvariable** des **Zustandsvektors**

- einzelne **Zustandsvariablen q_k** werden **rekursiv** mit δ berechnet
- eine **zwei Werte speichernde Einheit** wird als **Binärspeicher** bezeichnet: ***FlipFlop***
- ein **Speicher** mit **r *FlipFlops*** kann 2^r unterschiedliche **Zustände** speichern

➢ **grundsätzliche Funktionen eines Binärspeichers:**

- muss wahlweise eine **0** oder **1 einspeichern** 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

Also: **verschiedene Varianten** der **Binärspeicher** diskutieren

Binärspeicher (FlipFlops)

Binärspeicher-Varianten:

Fall 1: RS-FlipFlop

Lesesignal $\triangleq$ Zustandswert:

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

Schreiben einer 0, *Rücksetzen:*

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

Schreiben einer 1, *Setzen:*

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

Fall 2: D-FlipFlop

Lesesignal $\triangleq$ Zustandswert:

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

Datensignal $\triangleq$ Zustandswert neu:

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

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

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

→ das **gleichzeitige Schreiben** von **0** und **1** (R=1, S=1) bei **RS-FlipFlop** ist **unzulässig**

Funktionstabellen:

q_k^{alt}	R	S	q_k^{neu}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	-
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	-

q_k^{alt}	D	W	q_k^{neu}
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

Binärspeicher (FlipFlops)

Funktionstabellen:

q_k^{alt}	R	S	q_k^{neu}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	-
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	-

q_k^{alt}	D	W	q_k^{neu}
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

Übertragung der **zeitlichen Indizes** ν und $\nu+1$ auf die **Schaltnetzgrößen**:

$$q_k^{alt} = q_k^{\nu}, \qquad q_k^{neu} = q_k^{\nu+1}, R^{\nu}, S^{\nu}, D^{\nu}, W^{\nu}$$

- **Überführungsfunktion** zur **Bildung** von $q_k^{\nu+1}$ für **beide Fälle**:

R^{ν} \ S^{ν}	0	1	1	1
q_k^{ν}				
0	0	-	-	0

D^{ν} \ W^{ν}	0	0	0	1
q_k^{ν}				
0	0	1	1	1

- die **charakteristischen Gleichungen** lauten:

$$q_k^{\nu+1} = S^{\nu} \vee (q_k^{\nu} \& \overline{R^{\nu}}) \;,\; \text{mit } S^{\nu} \& R^{\nu} = 0 \qquad q_k^{\nu+1} = (D^{\nu} \& W^{\nu}) \vee (q_k^{\nu} \& \overline{W^{\nu}})$$

Binärspeicher (FlipFlops)

Zusammenhang für einen bestimmten Werteübergang: $q_k^{\nu} \rightarrow q_k^{\nu+1}$

→ gibt die sogenannte ***Ansteuerfunktion*** an:

Beispiele:

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

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

daraus ergibt sich:

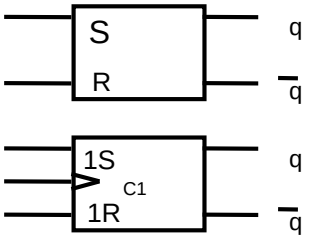
$R^{\nu} = \overline{q_k^{\nu+1}}$ $\quad S^{\nu} = q_k^{\nu+1}$ $\quad D^{\nu} = q_k^{\nu+1}$ $\quad W^{\nu} = q_k^{\nu+1} \not\equiv q_k^{\nu}$

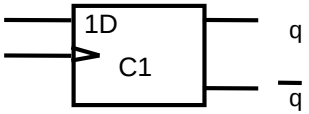
oder: $D^{\nu} = \overline{q_k^{\nu}}$

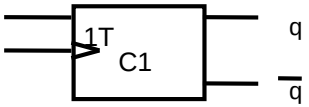
- neben der formalen Behandlung des **Binärspeicherverhaltens** ist die **technische Realisierung** solcher Elemente von Bedeutung

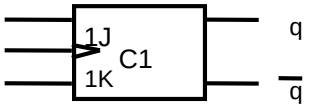
Binärspeicher (FlipFlops)

Symbole









Charakteristische Gleichungen

RS-FlipFlop

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

D-FlipFlop

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

T-FlipFlop

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

JK-FlipFlop

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

Ansteuerfunktionen

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

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

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

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

Ablaufdiagramm und Ablauftabelle

Beschreibung des Automatenverhaltens:

→ **nicht** alle Komponenten des **Eingabevektors** beim Übergang zum **Folgezustand** sind relevant und müssen berücksichtigt werden

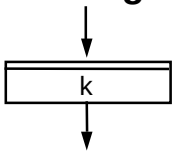
- **Schaltwerkstafel** bzw. **Schaltwerksgraph** sind als **Beschreibungsmittel** oftmals **ungünstig**

Daher: an dieser Stelle werden das **Ablaufdiagramm** bzw. die **Ablauftabelle** eingeführt, die sich durch eine **platzsparendere** und **effizientere Beschreibungsform** auszeichnen

Ablaufdiagramm und Ablauftabelle

Ablaufdiagramm:

1) Zustandsübergang

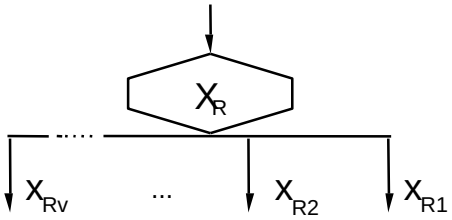


Übergang in den Zustand k,

Ausgelöst

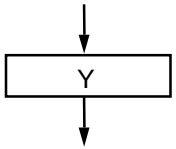
durch das Taktsignal

2) Abfrage (Verzweigung)



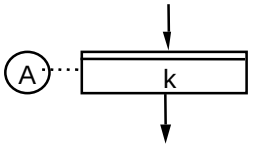
Abfrage der im aktuellen Zustand k relevanten Eingangsvariablen auf ihre Werte $X_R = (x_{Rv}, \ldots, x_{R2}, x_{R1})$

3) Ausgabe



Wert des Ausgangsvektors $Y = (y_m, \ldots, y_1)$ im aktuellen Zustand

4) Markierung des Anfangszustands



Ablaufdiagramm und Ablauftabelle

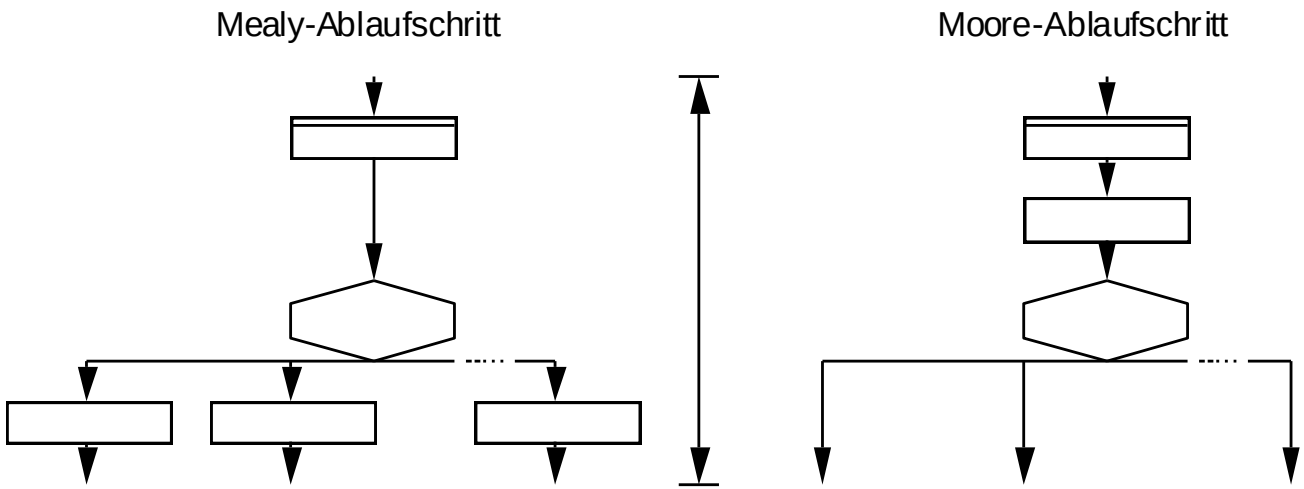
Mealy- und **Moore-Schaltwerke** werden entsprechend der **Abhängigkeiten**

$$Y^\nu = \lambda(Q^\nu, X^\nu) \quad \text{bzw.} \quad Y^\nu = \lambda(Q^\nu)$$

durch die **Reihenfolge** der **Symbole für Verzweigungen** und **Ausgabe berücksichtigt**

→ **Definition des *Ablaufschrittes***:

- **Tripel** aus **Zustandsübergang**, **Ausgabe** und **Verzweigung**, das jeweils zu einem **Wert des Index ν** gehört



Ablaufdiagramm und Ablauftabelle

Komplexere Schaltwerksbeschreibungen: **größere Zustandsanzahl**

→ es eignet sich eine **tabellenorientierte Darstellung** besser:

Ablauftabelle:

Q^{ν}	$X_R(k)$	$Q^{\nu+1}$	$Y(k)$
k			

- die **Ablauftabelle** setzt sich aus **Teiltabellen für jeden Ablaufschritt** zusammen
  → **Teiltabellen** bestehen aus je **vier Spalten**:
  - für den **momentanen Zustand Q^{ν}**
  - für die **Eingabeblöcke** der jeweils **relevanten Eingabevariablen $X_R(k)$**
  - für den **Folgezustand $Q^{\nu+1}$**
  - für die **Ausgabeblöcke Y(k)**

- Zustände sind in Ablauftabellen i.a. nicht kodiert

Ablaufdiagramm und Ablauftabelle

Beispiel:

Q^{ν}	$X_R(3)$				$Q^{\nu+1}$	$Y(3)$
	x_6	x_4	x_3	x_1		
l_1	0	-	0	1	l_1	Y_1
	0	1	0	0	l_2	
	0	1	1	1	l_2	
	1	1	0	1	l_2	
	1	0	-	-	l_3	
	-	0	1	1	l_3	
	1	1	1	0	l_4	

l_1, l_2, l_3, l_4 seien Folgezustände zum Zustand l_1,
Y_1 die Ausgabe gemäß Schaltwerkstyp (hier Moore).

Entwurf von Schaltwerken

Schaltwerkentwurf gliedert sich in folgende Schritte:

1. Schritt: **Definition** der **Ein- und Ausgabevariablen**

2. Schritt: Wahl des **Schaltwerktyps** und Erstellen des ***Ablaufdiagramms*** bzw. der ***Ablauftabelle*** gemäß Aufgabenstellung

3. Schritt: **Zustandskodierung**

4. Schritt: Wahl des **FlipFlop-Typs** und Aufstellen der **Ansteuerfunktionen**

5. Schritt: Entwurf des **Schaltnetzes** für die **Überführungsfunktion** 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

Beispiel: Verbale Aufgabenstellung

Eine **Schaltung** für einen sehr **einfachen Anrufbeantworter** soll zunächst ein Klingelzeichen des Telefons abwarten. Wenn bis zum Beginn des zweiten Klingelzeichens der Hörer nicht abgenommen wurde, soll der **Anrufbeantworter eingeschaltet** werden. Dieser **spielt** dann seine **Mitteilung ab**, egal wie lange der Anrufer tatsächlich zuhört. Erst wenn das Band vollständig abgespielt ist, wird der Anrufbeantworter wieder ausgeschaltet. Der Ausschaltvorgang beinhaltet ein automatisches Rückspulen des Bandes.

1. Schritt: **Eingabevariablen**: **Klingelzeichen**: **K = 1**

Klingelpause: **K = 0**

Timeout, d.h. Klingelpause dauert zu lange;
Anrufer hat **nach** dem **ersten Klingelzeichen**
aufgelegt: **T = 1**

sonst: T = 0

Bandende: **B = 1**

sonst: B = 0

Ausgabevariablen: **Anrufbeantworter ein**: **A = 1**

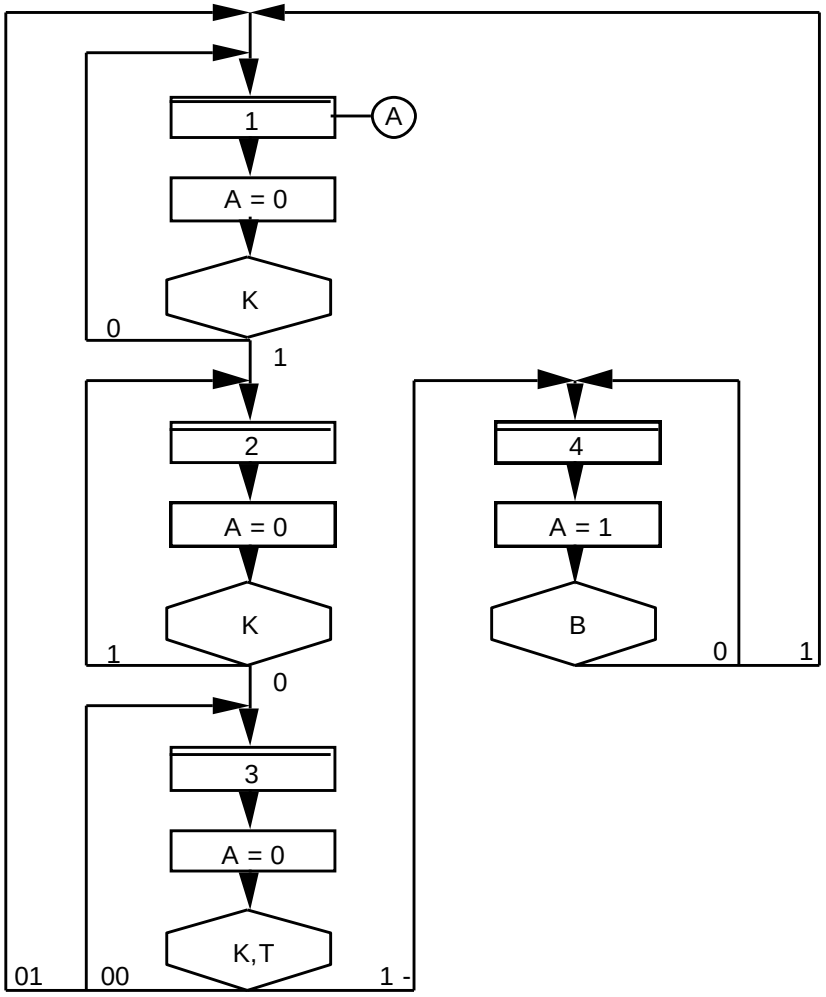
Anrufbeantworter aus: **A = 0**

Entwurf von Schaltwerken

2. Schritt:

Schaltwerkstyp: Moore

Ablaufdiagramm:



Entwurf von Schaltwerken

3. + 4. Schritt:

Für **4 Zustände** werden **2 Zustandsvariablen (q_2, q_1)** benötigt.
Erstellen der **kodierten Ablauftabelle**.
Die Kodierung wird willkürlich, z.B. als sogenannte Zählerkodierung ausgeführt.

Kodierte Ablauftabelle:

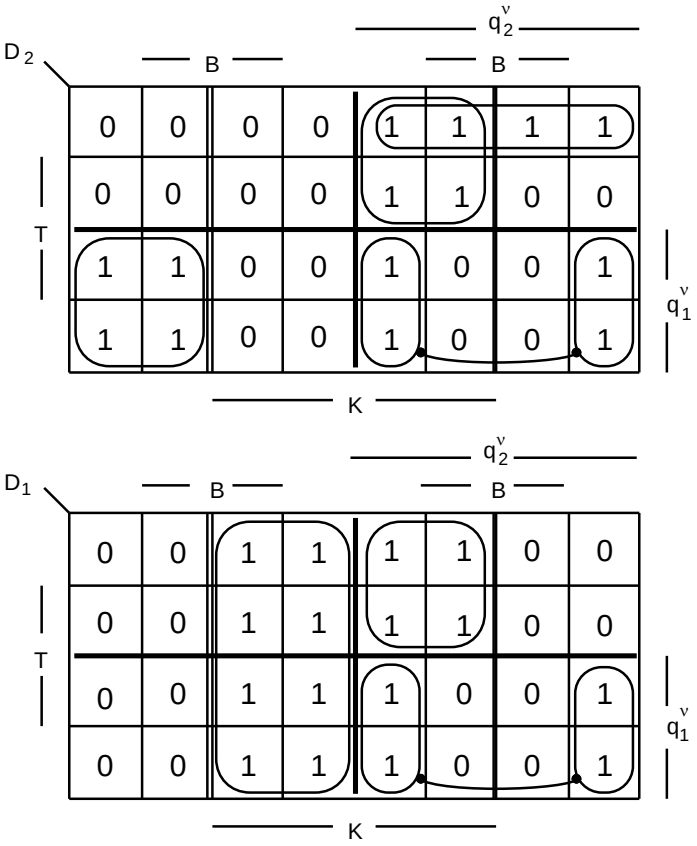
Q^ν q_2^ν	q_1^ν		$Q^{\nu+1}$ $q_2^{\nu+1}$	$q_1^{\nu+1}$	Y A
		$X_R(1)$:K			
0	0	0	0	0	0
		1	0	1	
		$X_R(2)$:K			
0	1	0	1	0	0
		1	0	1	
		$X_R(3)$:K,T			
1	0	0 0	1	0	0
		1 -	1	1	
		0 1	0	0	
		$X_R(4)$:B			
1	1	0	1	1	1
		1	0	0	

	q_2	q_1
Zustand 1	0	0
Zustand 2	0	1
Zustand 3	1	0
Zustand 4	1	1

Es sollen **D-Flipflops** verwendet werden, d.h. $q_2 \triangleq D_2$ und $q_1 \triangleq D_1$

Entwurf von Schaltwerken

5. Schritt:



Symmetriediagramme für Ansteuerfunktionen

$$D_2 = \overline{q_2^v} q_1^v \overline{K} \vee q_2^v \overline{q_1^v} \overline{T} \vee q_2^v \overline{q_1^v} K \vee q_2^v q_1^v \overline{B}$$

$$D_1 = \overline{q_2^v} K \vee q_2^v \overline{q_1^v} K \vee q_2^v q_1^v \overline{B}$$

Entwurf von Schaltwerken

6. Schritt: Aus der **Ablauftabelle** folgt sofort: $A = q_2^\nu q_1^\nu$

7. Schritt: Wenn zum Beispiel ausschließlich **Gatter** mit **2 Eingängen** zur Verfügung stehen

→ man muss man eine **entsprechende Umformung** vornehmen:

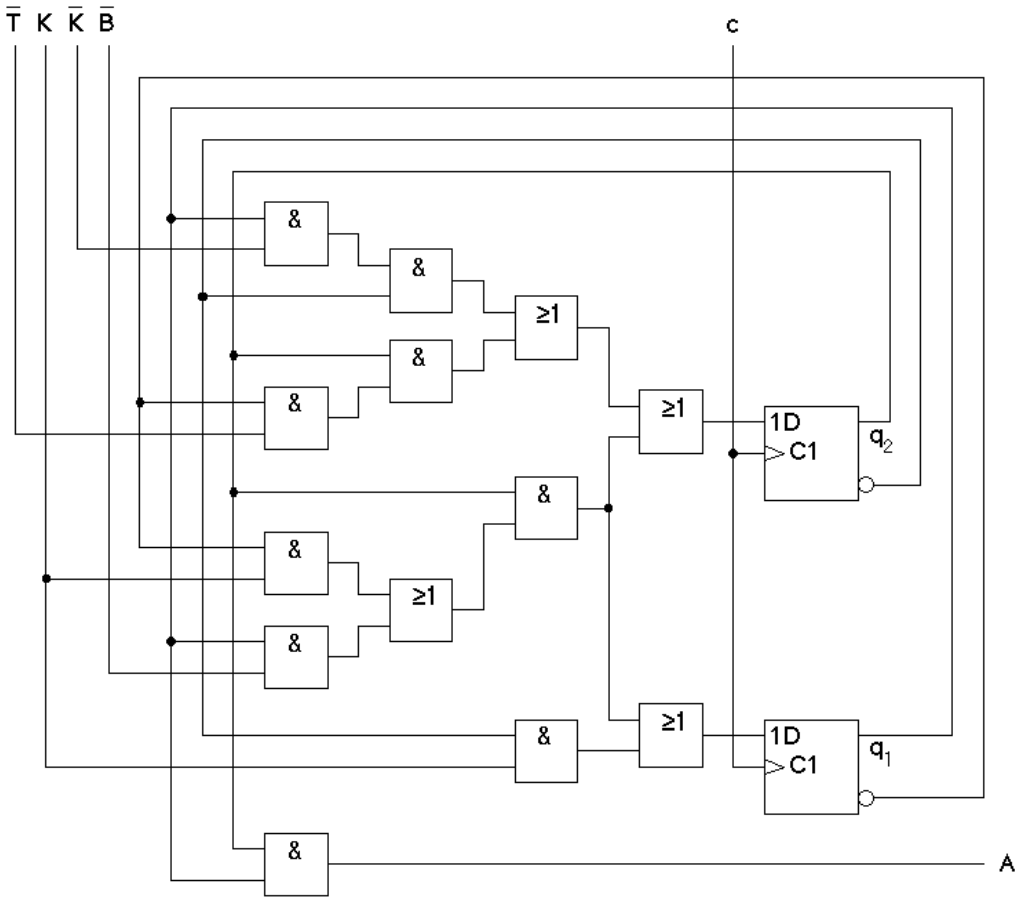
$$D_2 = \left[\overline{q_2^\nu} (q_1^\nu \overline{K}) \vee q_2^\nu (\overline{q_1^\nu} \overline{T}) \right] \vee q_2^\nu (\overline{q_1^\nu} K \vee q_1^\nu \overline{B})$$

$$D_1 = \overline{q_2^\nu} K \vee q_2^\nu (\overline{q_1^\nu} K \vee q_1^\nu \overline{B})$$

$$A = q_2^\nu q_1^\nu$$

Entwurf von Schaltwerken

8. Schritt: Blockschema (Schaltbild) des Schaltwerkes



Entwurf von Schaltwerken

Analog zu den Schaltnetzen:

- es besteht die Möglichkeit **zweistufige Logik** in **programmierbare integrierte Schaltungen** einzubetten
  - dies kann ebenso **bei Schaltwerken** geschehen
  - anhand des **Bausteins AmPAL16R6** soll dies **für das Beispiel** gezeigt werden
  - **Wegen negierter Ausgänge** des **Bausteins** muss unsere **Lösung umgeformt** werden:

$$A = q_2^v q_1^v = \overline{\overline{q_2^v q_1^v}} = \overline{\overline{q_2^v} \vee \overline{q_1^v}}$$

  - **Ansteuerfunktionen** für D_1 und D_2 können Schritt 5 entnommen werden

Entwurf von Schaltwerken

Einbettung in einen AmPAL16R6: