

Fabian Kempf

kempf@kit.edu

Karlsruher Institut für Technologie (KIT)

Institut für Technik der Informationsverarbeitung (ITIV)

Digitaltechnik

7. Übung

Fabian Kempf

kempf@kit.edu

Karlsruher Institut für Technologie (KIT)

Institut für Technik der Informationsverarbeitung (ITIV)

1. Aufgabe

- Bezeichnet eine formale Methode, die es erlaubt, eine Funktion $f(x_n, \dots, x_3, x_2, x_1)$ in eine DNF (oder KNF) umzuwandeln
- Die Entwicklung der Funktion f nach der Variablen x_i
→ Aufteilung in zwei Fälle: $x_i = 1$ und $x_i = 0$

$$\begin{aligned} f(x_n, \dots, x_i, \dots, x_1) &= [x_i \& f(x_n, \dots, 1, \dots, x_1)] \vee [\bar{x}_i \& f(x_n, \dots, 0, \dots, x_1)] \\ &= [x_i \& f \Big|_{x_i=1}] \vee [\bar{x}_i \& f \Big|_{x_i=0}] \end{aligned}$$

- Die nach x_i entwickelten Funktionen $f \Big|_{x_i=1}$ und $f \Big|_{x_i=0}$ werden als Restfunktionen oder Kofaktoren bezeichnet

- Bei der Auswertung des Entwicklungssatzes für n Variablen: n Klammerebenen im Funktionsterm:

$$F(d, c, b, a) = b[c(1) \vee \bar{c}(\bar{a})] \vee \bar{b}[c(d(1) \vee \bar{d}(\bar{a})) \vee \bar{c}(d(a) \vee \bar{d}(0))] \dots$$



- Methode zur erleichterten Übersicht: Rechnen mit Restfunktionen (für Multiplexer-Realisierung hilfreich):

- Beispiel: $y = f(x_1, x_2, x_3, \dots)$

- Entwicklung nach x_1 :

$$f(1, x_2, x_3, \dots) = ?,$$

$$f(0, x_2, x_3, \dots) = ?$$

- Entwicklung nach x_2 :

$$f(1, 1, x_3, \dots) = ?,$$

$$f(0, 1, x_3, \dots) = ?,$$

$$f(1, 0, x_3, \dots) = ?,$$

$$f(0, 0, x_3, \dots) = ?$$

- Entwicklung nach x_3 : (...), usw.

- Weitere Auffächerung der Restfunktionen wird abgebrochen, sobald konstant wertig

1. Aufgabe – 1.1

- Die gegebene Schaltfunktion $y = f(d, c, b, a)$ soll mit 2:1 Multiplexern realisiert werden. Dazu muss die Funktion nach jeder Variablen mit Hilfe des Entwicklungssatzes entwickelt werden.

$$y = \overline{\overline{a}}\overline{\overline{c}} + b + \overline{\overline{d}}\overline{\overline{c}} + adc$$

- 1.1 Entwickeln Sie die Schaltfunktion nach der Variable b . Geben Sie alle Zwischenschritte an.

$$y = f(d, c, b, a) = \overline{\overline{a}}\overline{\overline{c}} + b + \overline{\overline{d}}\overline{\overline{c}} + adc$$

$$\begin{aligned} f_b(d, c, 1, a) &= \overline{\overline{a}}\overline{\overline{c}} + 1 + \overline{\overline{d}}\overline{\overline{c}} + adc \\ &= 1 \end{aligned}$$

$$\begin{aligned} f_{\overline{b}}(d, c, 0, a) &= \overline{\overline{a}}\overline{\overline{c}} + 0 + \overline{\overline{d}}\overline{\overline{c}} + adc \\ &= \overline{\overline{a}}\overline{\overline{c}} + \overline{\overline{d}}\overline{\overline{c}} + adc \end{aligned}$$

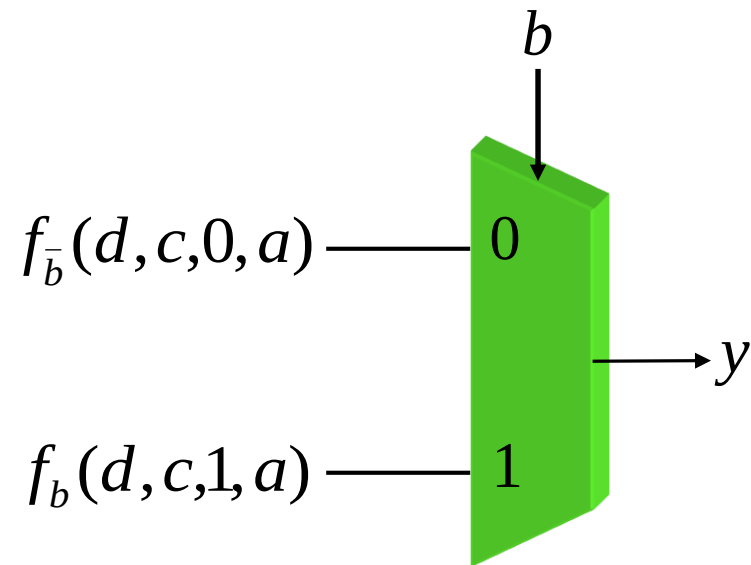
1. Aufgabe – 1.1

- 1.1 Entwickeln Sie die Schaltfunktion nach der Variable b . Geben Sie alle Zwischenschritte an.

Teilrealisierung mit 2:1 Multiplexern:

$$f_{\bar{b}}(d, c, 0, a) = \bar{a}\bar{c} + \bar{d}\bar{c} + adc$$

$$f_b(d, c, 1, a) = 1$$



1. Aufgabe – 1.2

- 1.2 Entwickeln Sie die Restfunktionen zuerst nach der Variablen c und dann, falls erforderlich, nach den verbleibenden Variablen, so dass als Restfunktionen nur noch Konstanten übrig bleiben.

$f_b(d, c, 1, a) = 1 \rightarrow$ konstante Funktion, Betrachtung nicht erforderlich!!!

$f_{\bar{b}}(d, c, 0, a) = \bar{a}\bar{c} + \bar{d}\bar{c} + adc \rightarrow$ Entwicklung nach c !!!

1. Aufgabe – 1.2

$$f_{\bar{b}}(d, c, 0, a) = \bar{\bar{a}}\bar{c} + \bar{\bar{d}}\bar{c} + adc$$

Entwicklung nach c:

$$\begin{aligned} f_{\bar{b}\bar{c}}(d, 0, 0, a) &= \bar{a} \cdot \bar{0} + \bar{d} \cdot \bar{0} + ad \cdot 0 \\ &= \bar{a} \cdot 1 + \bar{d} \cdot 1 \\ &= \bar{a} + \bar{d} \\ f_{\bar{b}c}(d, 1, 0, a) &= \bar{a} \cdot \bar{1} + \bar{d} \cdot \bar{1} + ad \cdot 1 \\ &= \bar{a} \cdot 0 + \bar{d} \cdot 0 + ad \\ &= ad \end{aligned}$$

Weitere Entwicklung beider Funktionen nach d erforderlich !!!

1. Aufgabe – 1.2

Zunächst die Restfunktion $f_{\bar{b}\bar{c}}(d,0,0,a) = \bar{a} + \bar{d}$

Entwicklung nach d:

$$f_{\bar{b}\bar{c}\bar{d}}(0,0,0,a) = \bar{a} + \bar{0} = \bar{a} + 1 = 1$$
$$f_{\bar{b}\bar{c}\bar{d}}(1,0,0,a) = \bar{a} + \bar{1} = \bar{a} + 0 = \bar{a}$$

Nun die Restfunktion $f_{\bar{b}\bar{c}}(d,1,0,a) = ad$

Entwicklung nach d:

$$f_{\bar{b}\bar{c}\bar{d}}(0,1,0,a) = a \cdot 0 = 0$$
$$f_{\bar{b}\bar{c}\bar{d}}(1,1,0,a) = a \cdot 1 = a$$

Weitere Entwicklung nach a erforderlich !!!

1. Aufgabe – 1.2

Entwicklung nach a:

$$f_{bcd}^-(1,1,0,a) = a$$

$$f_{bcd\bar{a}}^-(1,1,0,0) = 0$$

$$f_{bcd\bar{a}}^-(1,1,0,1) = 1$$

$$f_{bcd}^{\bar{\bar{}}} (1,0,0,a) = \bar{a}$$

$$f_{bcd\bar{a}}^{\bar{\bar{}}} (1,0,0,0) = 1$$

$$f_{bcd\bar{a}}^{\bar{\bar{}}} (1,1,0,1) = 0$$

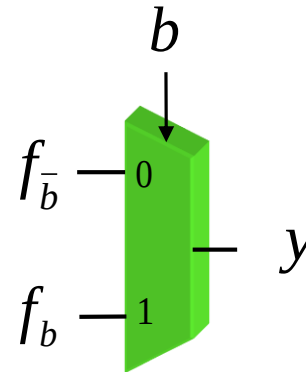
Die Funktion y lautet nun:

$$\begin{aligned} y = f(d,c,b,a) &= \bar{b} \cdot (\bar{c} \cdot (\bar{d} \cdot f_{bcd}^{\bar{\bar{}}} + d \cdot (\bar{a} \cdot f_{bcd\bar{a}}^{\bar{\bar{}}} + a \cdot f_{bcd\bar{a}}^-)) + \\ &\quad c \cdot (\bar{d} \cdot f_{bcd}^- + d \cdot (\bar{a} \cdot f_{bcd\bar{a}}^- + a \cdot f_{bcd\bar{a}}^-))) + b \cdot f_b \\ &= \bar{b} \cdot (\bar{c} \cdot (\bar{d} \cdot 1 + d \cdot (\bar{a} \cdot 1 + a \cdot 0)) + c \cdot (\bar{d} \cdot 0 + d \cdot (\bar{a} \cdot 0 + a \cdot 1))) + b \cdot 1 \end{aligned}$$

1. Aufgabe – 1.2

- Multiplexerrealisierung durch Heranziehen aller entwickelten Teilfunktionen

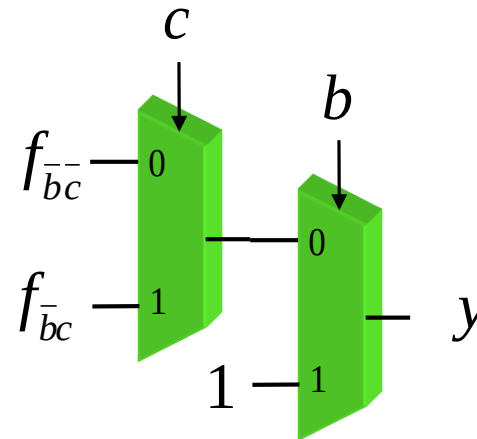
$$y = \bar{b} \cdot (\bar{c} \cdot (\bar{d} \cdot 1 + d \cdot (\bar{a} \cdot 1 + a \cdot 0)) + c \cdot (\bar{d} \cdot 0 + d \cdot (\bar{a} \cdot 0 + a \cdot 1))) + b \cdot 1$$



1. Aufgabe – 1.2

- Multiplexerrealisierung durch Heranziehen aller entwickelten Teilfunktionen

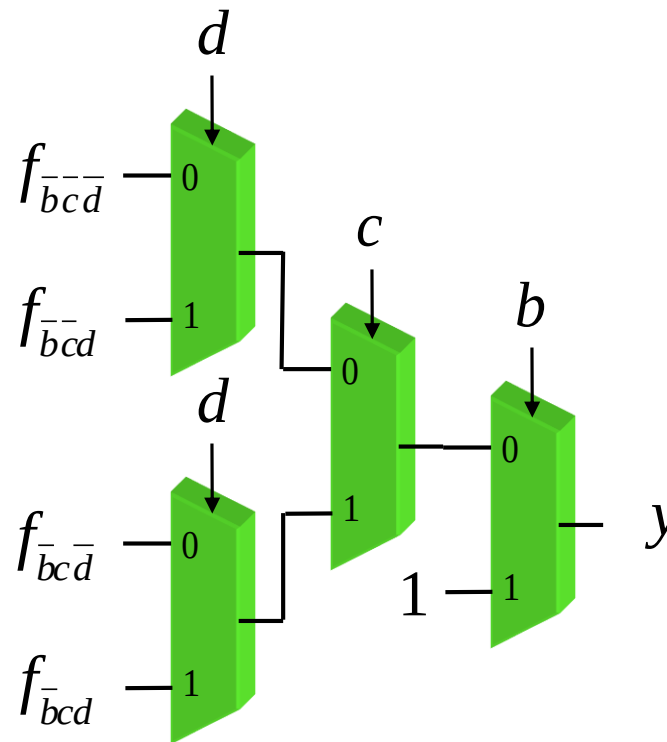
$$y = \bar{b} \cdot (\bar{c} \cdot (\bar{d} \cdot 1 + d \cdot (\bar{a} \cdot 1 + a \cdot 0)) + c \cdot (\bar{d} \cdot 0 + d \cdot (\bar{a} \cdot 0 + a \cdot 1))) + b \cdot 1$$



1. Aufgabe – 1.2

- Multiplexerrealisierung durch Heranziehen aller entwickelten Teilfunktionen

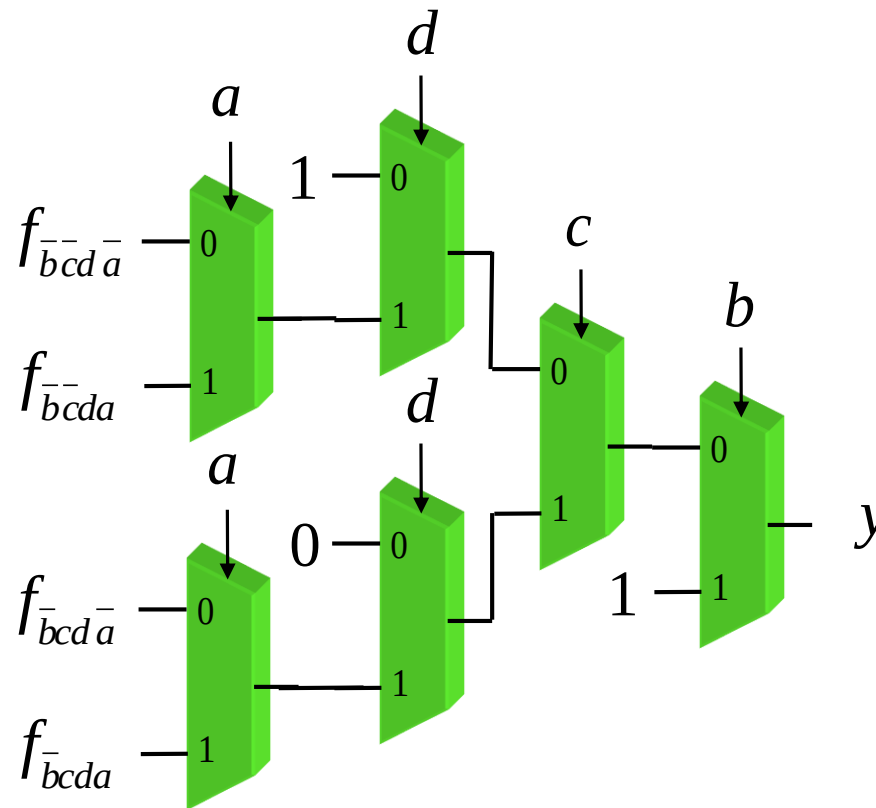
$$y = \bar{b} \cdot (\bar{c} \cdot (\bar{d} \cdot 1 + d \cdot (\bar{a} \cdot 1 + a \cdot 0)) + c \cdot (\bar{d} \cdot 0 + d \cdot (\bar{a} \cdot 0 + a \cdot 1))) + b \cdot 1$$



1. Aufgabe – 1.2

■ Multiplexerrealisierung durch Heranziehen aller entwickelten Teilfunktionen

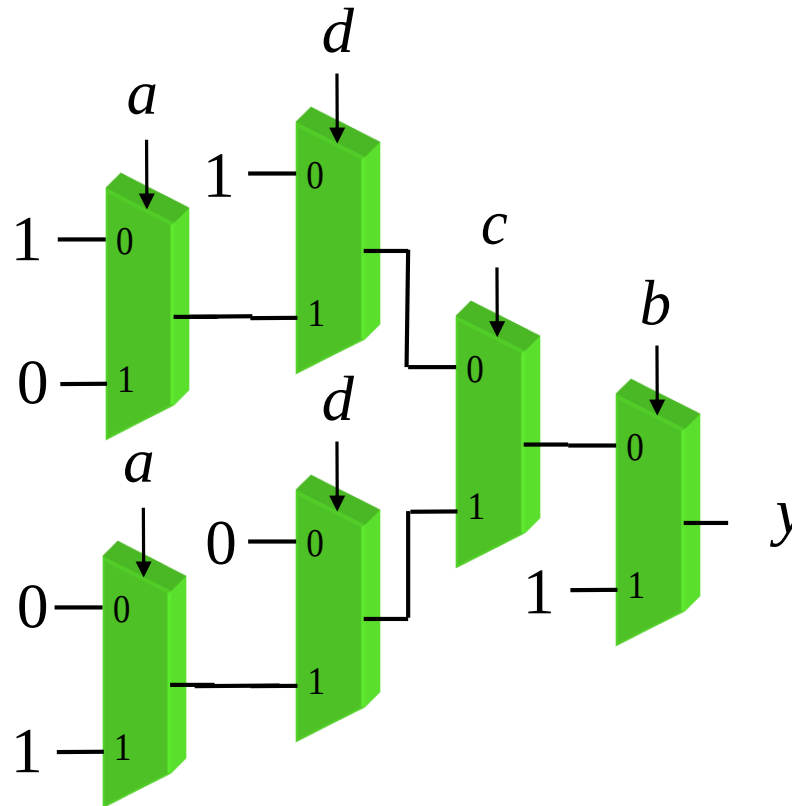
$$y = \bar{b} \cdot (\bar{c} \cdot (\bar{d} \cdot 1 + d \cdot (\bar{a} \cdot 1 + a \cdot 0)) + c \cdot (\bar{d} \cdot 0 + d \cdot (\bar{a} \cdot 0 + a \cdot 1))) + b \cdot 1$$



1. Aufgabe – 1.2

■ Multiplexerrealisierung durch Heranziehen aller entwickelten Teilfunktionen

$$y = \bar{b} \cdot (\bar{c} \cdot (\bar{d} \cdot 1 + d \cdot (\bar{a} \cdot 1 + a \cdot 0)) + c \cdot (\bar{d} \cdot 0 + d \cdot (\bar{a} \cdot 0 + a \cdot 1))) + b \cdot 1$$



1. Aufgabe – 1.2

■ Alternative Realisierung mittels 8:1 Multiplexer

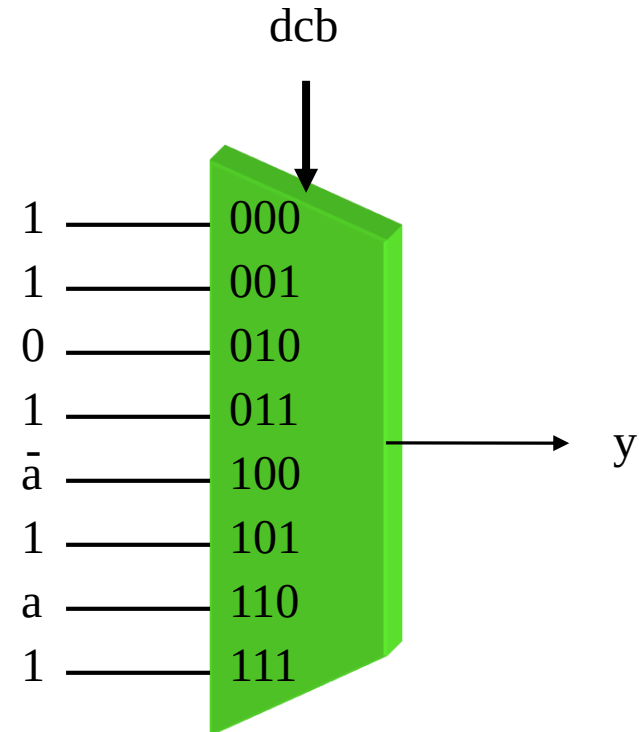
$$f_b(d, c, 1, a) = 1$$

$$f_{\bar{b}\bar{c}\bar{d}}(0, 1, 0, a) = 0$$

$$f_{\bar{b}cd}(1, 1, 0, a) = a$$

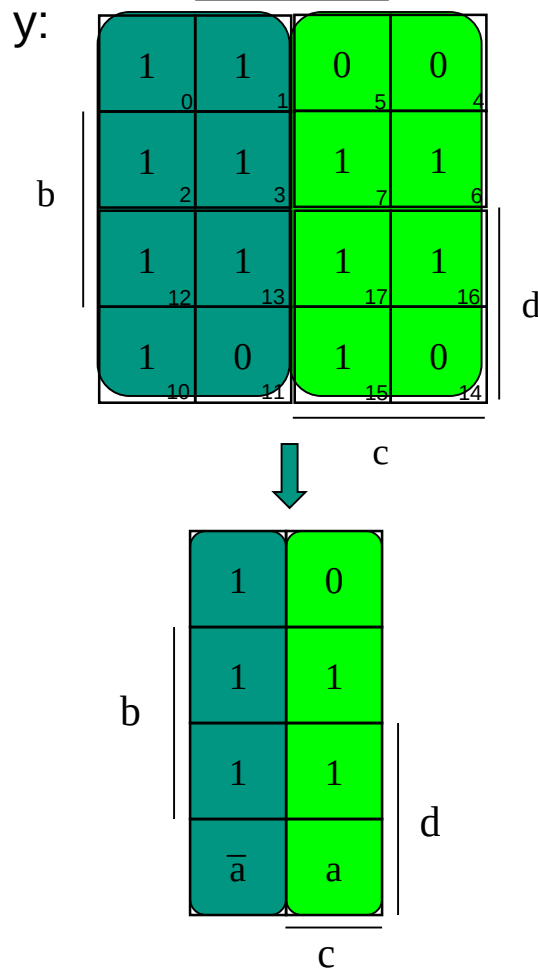
$$f_{\bar{b}\bar{c}\bar{d}}(0, 0, 0, a) = 1$$

$$f_{\bar{b}cd}(1, 0, 0, a) = \bar{a}$$

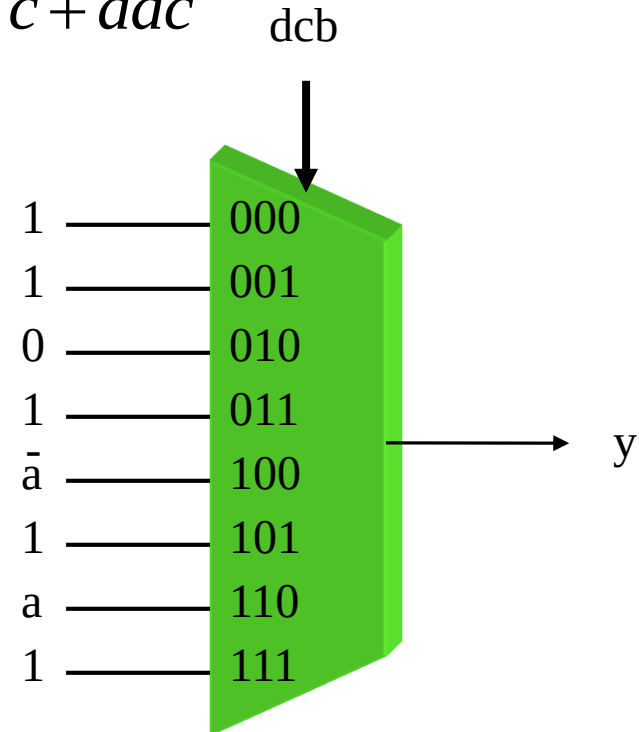


1. Aufgabe – 1.2

Alternative Realisierung mittels 8:1 Multiplexer



$$y = \bar{a}\bar{c} + b + \bar{d}\bar{c} + adc$$



Fabian Kempf

kempf@kit.edu

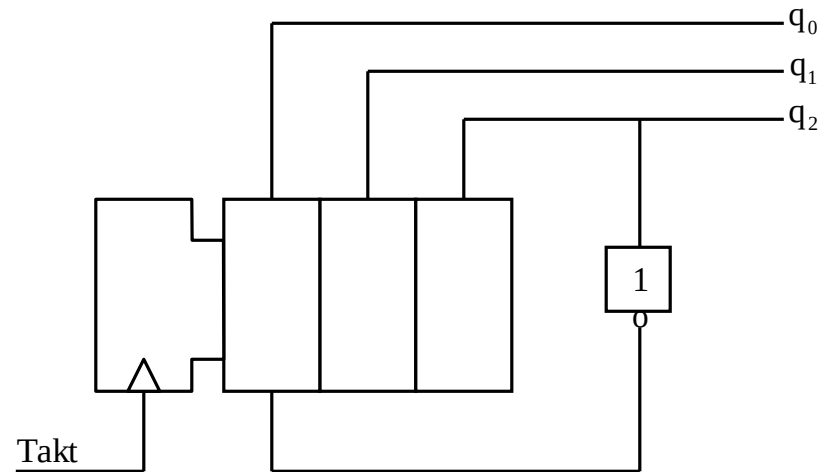
Karlsruher Institut für Technologie (KIT)

Institut für Technik der Informationsverarbeitung (ITIV)

3. Aufgabe

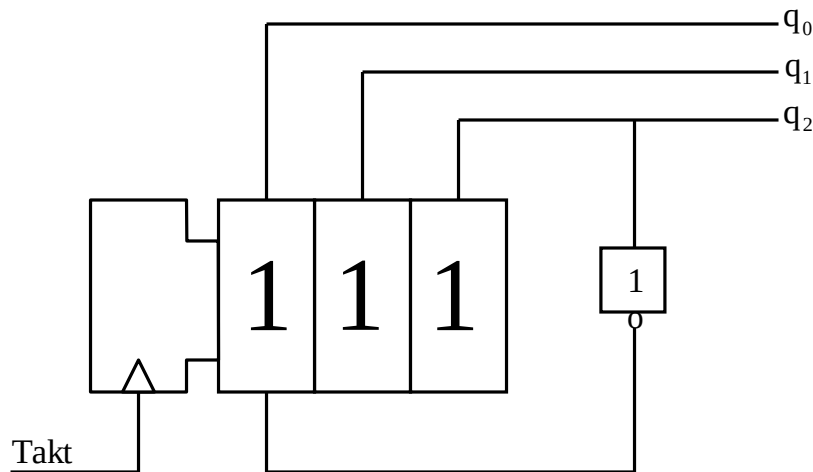
3. Aufgabe

- Im unteren Bild ist ein rückgekoppeltes Schieberegister dargestellt.



3. Aufgabe – 3.1

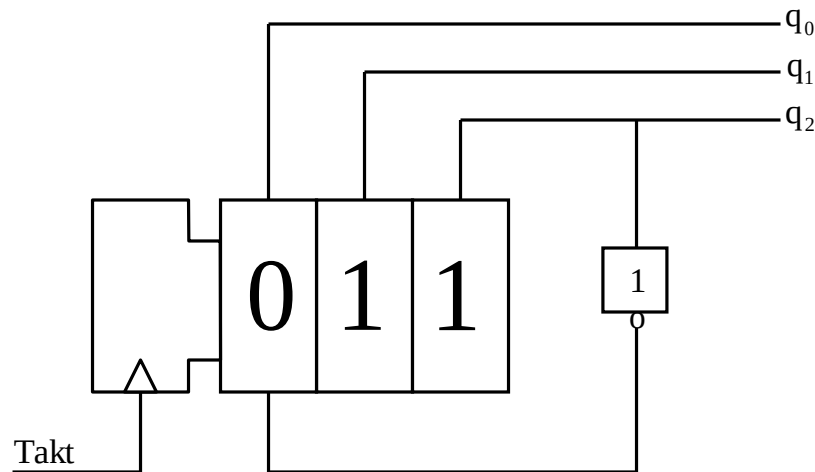
- 3.1 Bestimmen Sie den Zählzyklus, wenn das Schieberegister am Anfang mit 111 (q_2, q_1, q_0) belegt ist.



Takt	Belegung (q_2, q_1, q_0)
0	111
1	
2	
3	
4	
5	
6	

3. Aufgabe – 3.1

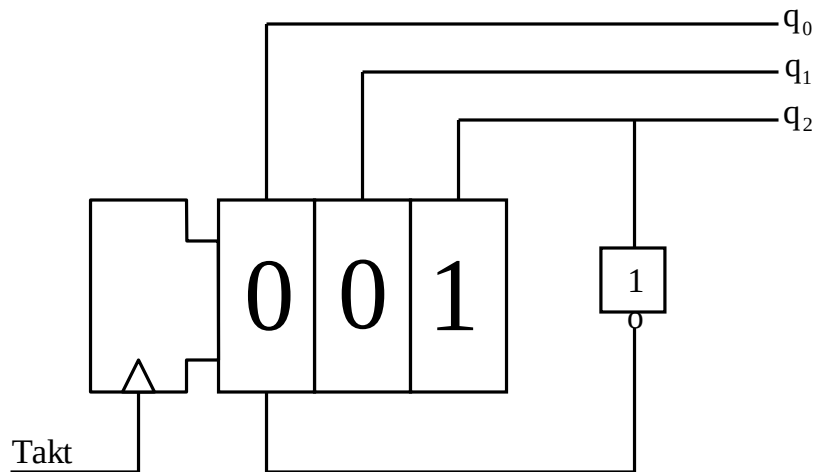
- 3.1 Bestimmen Sie den Zählzyklus, wenn das Schieberegister am Anfang mit 111 (q_2, q_1, q_0) belegt ist.



Takt	Belegung (q_2, q_1, q_0)
0	111
1	110
2	
3	
4	
5	
6	

3. Aufgabe – 3.1

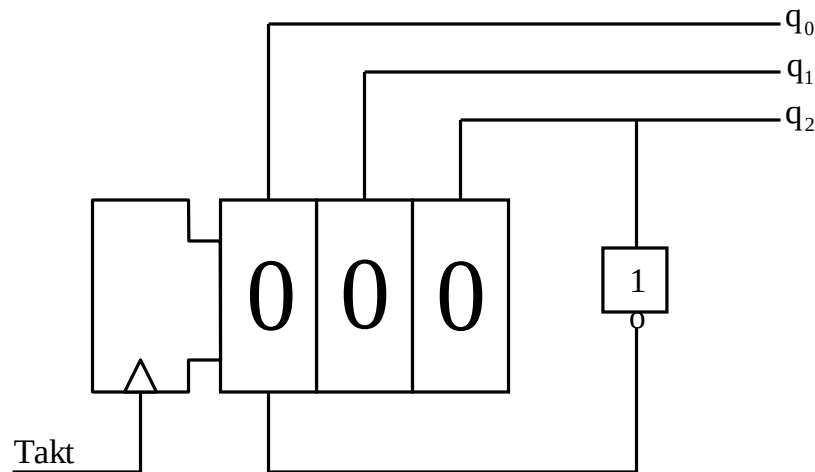
- 3.1 Bestimmen Sie den Zählzyklus, wenn das Schieberegister am Anfang mit 111 (q_2, q_1, q_0) belegt ist.



Takt	Belegung (q_2, q_1, q_0)
0	111
1	110
2	100
3	
4	
5	
6	

3. Aufgabe – 3.1

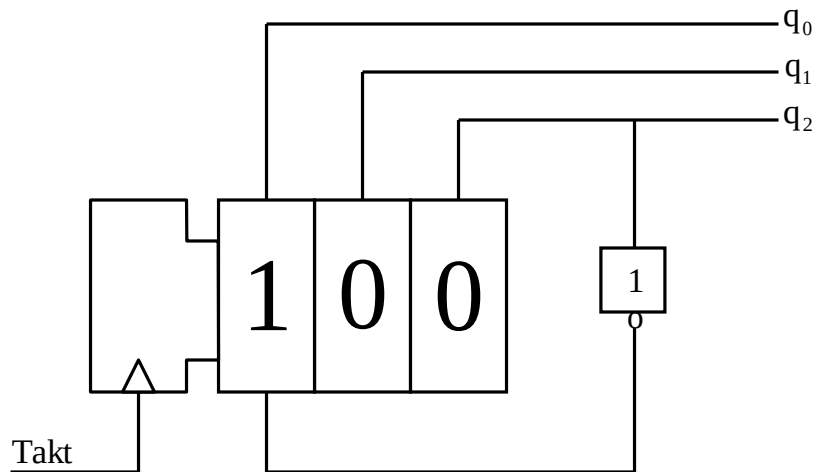
- 3.1 Bestimmen Sie den Zählzyklus, wenn das Schieberegister am Anfang mit 111 (q_2, q_1, q_0) belegt ist.



Takt	Belegung (q_2, q_1, q_0)
0	111
1	110
2	100
3	000
4	
5	
6	

3. Aufgabe – 3.1

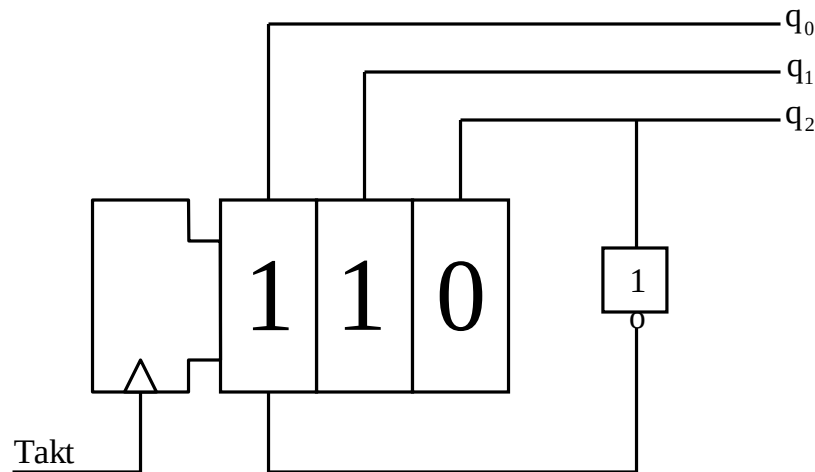
- 3.1 Bestimmen Sie den Zählzyklus, wenn das Schieberegister am Anfang mit 111 (q_2, q_1, q_0) belegt ist.



Takt	Belegung (q_2, q_1, q_0)
0	111
1	110
2	100
3	000
4	001
5	
6	

3. Aufgabe – 3.1

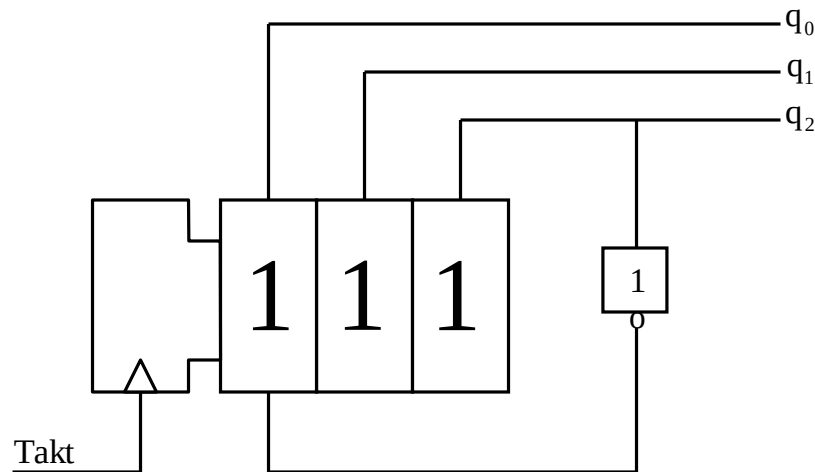
- 3.1 Bestimmen Sie den Zählzyklus, wenn das Schieberegister am Anfang mit 111 (q_2, q_1, q_0) belegt ist.



Takt	Belegung (q_2, q_1, q_0)
0	111
1	110
2	100
3	000
4	001
5	011
6	

3. Aufgabe – 3.1

- 3.1 Bestimmen Sie den Zählzyklus, wenn das Schieberegister am Anfang mit 111 (q_2, q_1, q_0) belegt ist.



Takt	Belegung (q_2, q_1, q_0)
0	111
1	110
2	100
3	000
4	001
5	011
6	111

A curved arrow points from the state 111 at Takt 6 back to the state 111 at Takt 0, indicating a cycle. A dashed horizontal line is drawn between Takt 5 and Takt 6.

→ ist ein zyklischer Vorgang !!!

3. Aufgabe – 3.2

3.2 Welche Belegungen werden bei diesem Zähler nicht erreicht?

Takt	Belegung (q_2, q_1, q_0)
0	111
1	110
2	100
3	000
4	001
5	011

- Von 8 möglichen Belegungen, werden lediglich 6 Belegungen erreicht.
- es fehlen also 2 Belegungen →

(q_2, q_1, q_0)
010
101

Fabian Kempf

kempf@kit.edu

Karlsruher Institut für Technologie (KIT)

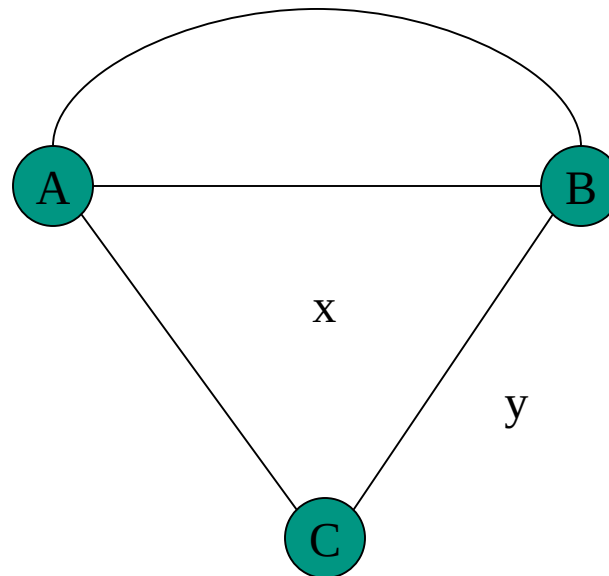
Institut für Technik der Informationsverarbeitung (ITIV)

5. Aufgabe

5. Aufgabe – 5.1

- 5.1 Zeichnen Sie ein Beispiel für einen ungerichteten Graphen F mit 3 Knoten, der folgende Eigenschaften hat: Der Graph F ist isomorph zu seinem dualen Graphen F_D

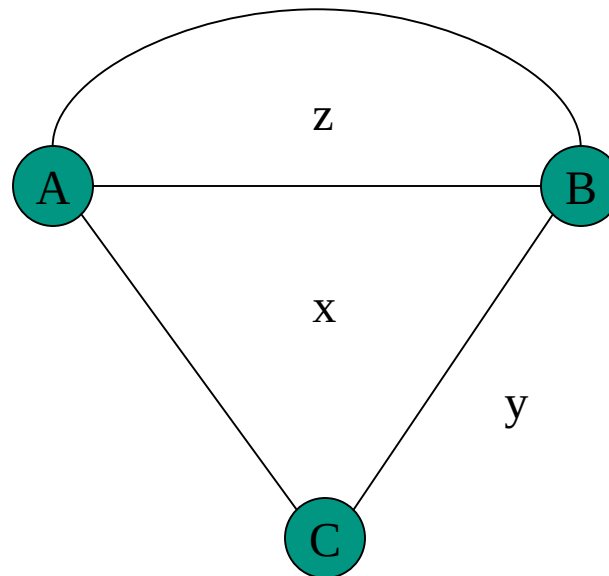
F



5. Aufgabe – 5.1

- 5.1 Zeichnen Sie ein Beispiel für einen ungerichteten Graphen F mit 3 Knoten, der folgende Eigenschaften hat: Der Graph F ist isomorph zu seinem dualen Graphen F_D

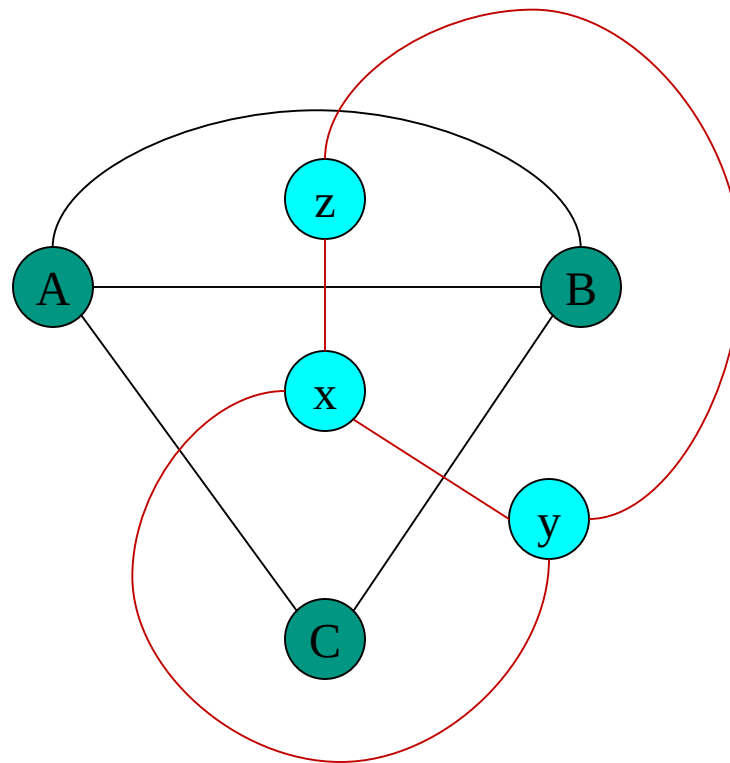
F



5. Aufgabe – 5.1

- 5.1 Zeichnen Sie ein Beispiel für einen ungerichteten Graphen F mit 3 Knoten, der folgende Eigenschaften hat: Der Graph F ist isomorph zu seinem dualen Graphen F_D

F



F_D

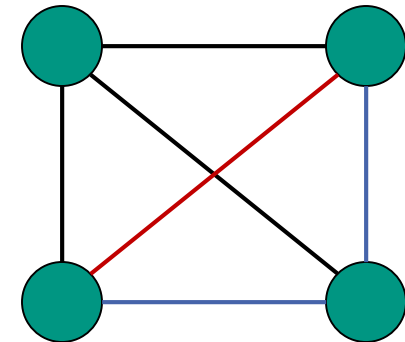


5. Aufgabe – 5.2

- 5.2 Vorgegeben seien die folgenden Eigenschaften eines Graphen: er ist ungerichtet, vollständig, einfach, hat 7 Knoten und keine Schleifen. Berechnen Sie die Anzahl der Kanten

bei 4 Knoten:

$$3 + 2 + 1 = 6$$



bei 7 Knoten:

$$6 + 5 + 4 + 3 + 2 + 1 = 21 !!!$$

Fabian Kempf

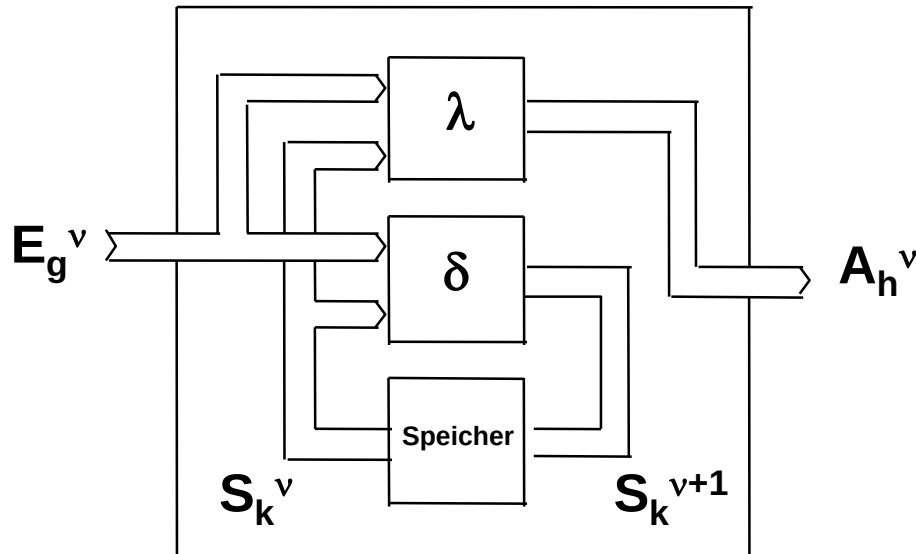
kempf@kit.edu

Karlsruher Institut für Technologie (KIT)

Institut für Technik der Informationsverarbeitung (ITIV)

2. Aufgabe

- Darstellung des Automaten durch folgende Struktur:



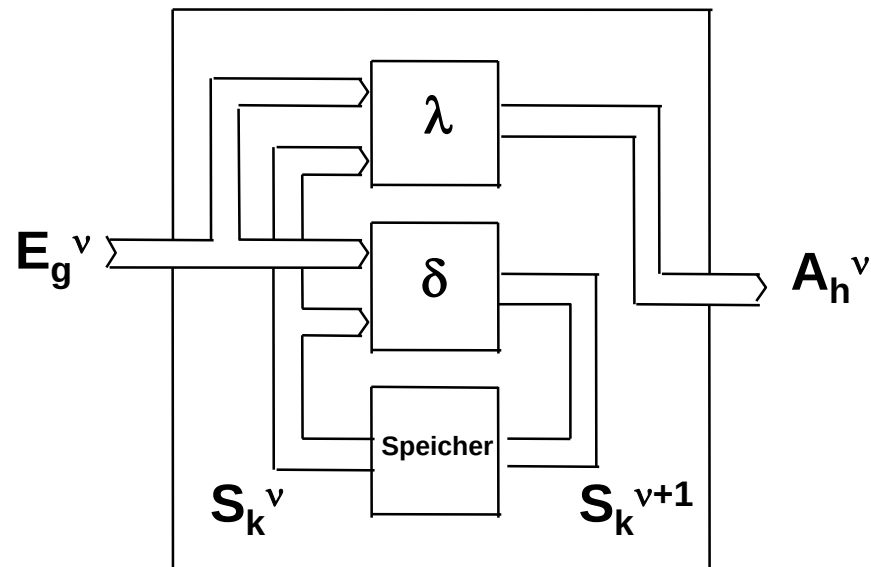
- rekursive Anordnung für die Bestimmung des neuen Zustandes
- der Speicher nimmt den momentanen Zustand S_k^v auf
- der neue Zustand S_k^{v+1} muss zum richtigen Zeitpunkt übernommen werden

Wichtige Typklassen von Automaten im Zusammenhang mit Digitalschaltungen:

- *endliche, diskrete* und *deterministische* Automaten
- Bezüglich der Ausgabefunktion unterscheidet man *drei Fälle*
- **Mealy-Automat:** mit $A_h^v = \lambda(E_g^v, S_k^v)$ als allgemeinster Fall:

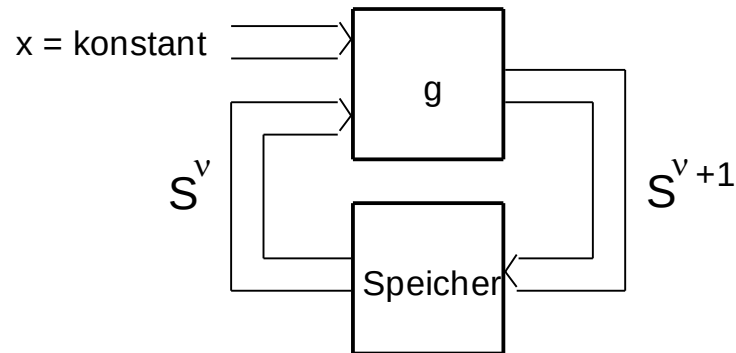
- A_h^v : Ausgansvektor
- S_k^v : Zustandsvektor
- E_g^v : Eingangsvektor
- λ : Ausgabefunktion
- δ : Transitionsfunktion

$$s_i^{v+1} = \delta_i(S_k^v, E_g^v), \text{ mit } s_i^{v+1} \in S_k^{v+1}$$



Weiterhin: formal eingeführte Indizierung v der zeitlichen Ordnung
muss nun in technische Realisierung überführt werden

→ Rückführung der Zustände über einen Speicher:



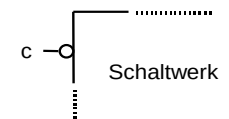
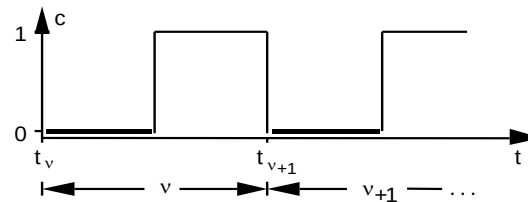
- Schaltwerke, bei denen Rückführungen direkt wirksam werden, heißen **asynchron** → sie sind schwieriger zu entwerfen

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

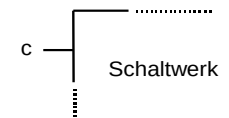
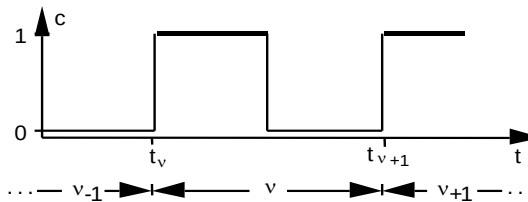
Mealy-Automat: Änderung der **Eingabe** bewirkt **asynchrone Ausgabenberechnung**
 → die **Zählung** des **Indexes v** muss aus dem **Taktsignal** abgeleitet werden

Steuerungsarten des Taktes:

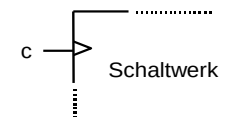
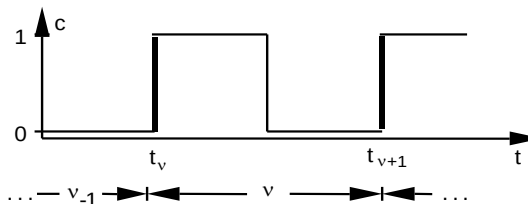
pegelgesteuert:



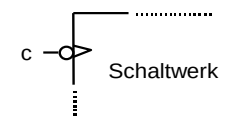
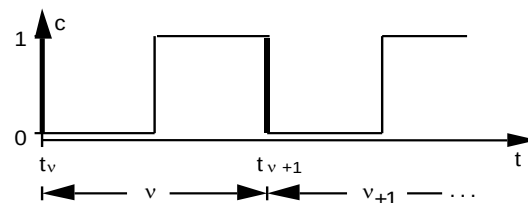
pegelgesteuert:



flankengesteuert:

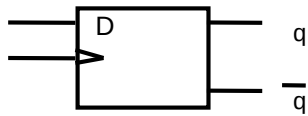
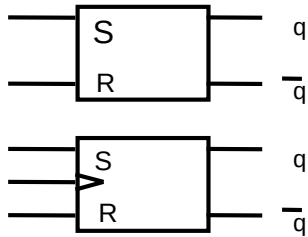


flankengesteuert:



Binärspeicher (FlipFlops)

Symbole



Charakteristische Gleichungen

RS-FlipFlop

$$q^{v+1} = S \vee (q^v \& \bar{R})$$

D-FlipFlop

$$q^{v+1} = D$$

T-FlipFlop

$$q^{v+1} = (T \& \bar{q}^v) \vee (T \& q^v)$$

JK-FlipFlop

$$q^{v+1} = (\bar{K} \& q^v) \vee (J \& \bar{q}^v)$$

Ansteuerfunktionen

q^v	q^{v+1}	R	S
0	0	-	0
0	1	0	1
1	0	1	0
1	1	0	-

q^v	q^{v+1}	D
0	0	0
0	1	1
1	0	0
1	1	1

q^v	q^{v+1}	T
0	0	0
0	1	1
1	0	1
1	1	0

q^v	q^{v+1}	K	J
0	0	-	0
0	1	-	1
1	0	1	-
1	1	0	-

2. Aufgabe

- Gegeben ist die codierte Ablauftafel eines Automaten. Bestimmen Sie die Ansteuerfunktionen für eine Realisierung des Automaten mit JK-Flipflops.

Q^n		X		Q^{n+1}		Flipflop2		Flipflop1	
q_2^n	q_1^n	b	a	q_2^{n+1}	q_1^{n+1}	J_2	K_2	J_1	K_1
0	0	0	0	0	1	0	-	1	-
0	0	0	1	0	1	0	-	1	-
0	0	1	-	1	1	1	-	1	-
0	1	-	0	1	0	1	-	-	1
0	1	-	1	0	1	0	-	-	0
1	0	-	0	0	0	-	1	0	-
1	0	0	1	1	1	-	0	1	-
1	0	1	1	1	0	-	0	0	-
1	1	-	-	0	0	-	1	-	1

Ansteuerfunktion
eines JK-Flipflops:

q^n	q^{n+1}	J	K
0	0	0	-
0	1	1	-
1	0	-	1
1	1	-	0

2. Aufgabe – 2.1

- 2.1 Können Sie anhand der Tabelle den Automatentyp angeben (mit Begründung)?

Nein, da die Unterscheidung der Automaten anhand der Ausgabefunktionen erfolgt !!!

Q^n		X		Q^{n+1}		Flipflop2		Flipflop1	
q_2^n	q_1^n	b	a	q_2^{n+1}	q_1^{n+1}	J_2	K_2	J_1	K_1
0	0	0	0	0	1	0	-	1	-
0	0	0	1	0	1	0	-	1	-
0	0	1	-	1	1	1	-	1	-
0	1	-	0	1	0	1	-	-	1
0	1	-	1	0	1	0	-	-	0
1	0	-	0	0	0	-	1	0	-
1	0	0	1	1	1	-	0	1	-
1	0	1	1	1	0	-	0	0	-
1	1	-	-	0	0	-	1	-	1

2. Aufgabe – 2.2

- 2.2 Tragen Sie die Ansteuerfunktionen J1, K1, J2, K2 in Symmetriediagramme ein und bestimmen Sie die disjunktiven Minimalformen.

J₁

a

	1	1	-	-	
	0	1	5	4	
b	1	1	-	-	
	2	3	7	6	
	0	0	-	-	
	12	13	17	16	
	0	1	-	-	
	10	11	15	14	
					q ₁
					q ₂

$$\underline{\underline{J_1 = \overline{q_2} + a\overline{b}}}$$

Q ⁿ		X		Q ⁿ⁺¹		FF2		FF1	
q ₂ ⁿ	q ₁ ⁿ	b	a	q ₂ ⁿ⁺¹	q ₁ ⁿ⁺¹	J ₂	K ₂	J ₁	K ₁
0	0	0	0	0	1	0	-	1	-
0	0	0	1	0	1	0	-	1	-
0	0	1	-	1	1	1	-	1	-
0	1	-	0	1	0	1	-	-	1
0	1	-	1	0	1	0	-	-	0
1	0	-	0	0	0	-	1	0	-
1	0	0	1	1	1	-	0	1	-
1	0	1	1	1	0	-	0	0	-
1	1	-	-	0	0	-	1	-	1

2. Aufgabe – 2.2

- 2.2 Tragen Sie die Ansteuerfunktionen J1, K1, J2, K2 in Symmetriediagramme ein und bestimmen Sie die disjunktiven Minimalformen.

K₁

a

	-	-	0	1
	0	1	5	4
	-	-	0	1
	2	3	7	6
	-	-	1	1
	12	13	17	16
	-	-	1	1
	10	11	15	14

q₁

b

q₂

Q ⁿ		X		Q ⁿ⁺¹		FF2		FF1	
q ₂ ⁿ	q ₁ ⁿ	b	a	q ₂ ⁿ⁺¹	q ₁ ⁿ⁺¹	J ₂	K ₂	J ₁	K ₁
0	0	0	0	0	1	0	-	1	-
0	0	0	1	0	1	0	-	1	-
0	0	1	-	1	1	1	-	1	-
0	1	-	0	1	0	1	-	-	1
0	1	-	1	0	1	0	-	-	0
1	0	-	0	0	0	-	1	0	-
1	0	0	1	1	1	-	0	1	-
1	0	1	1	1	0	-	0	0	-
1	1	-	-	0	0	-	1	-	1

$$\underline{\underline{K_1 = q_2 + \bar{a}}}$$

2. Aufgabe – 2.2

J_2

	a			
	0	0	0	1
	₀	₁	₅	₄
	1	1	0	1
	₂	₃	₇	₆
b	-	-	-	-
	₁₂	₁₃	₁₇	₁₆
	-	-	-	-
	₁₀	₁₁	₁₅	₁₄
	q_1			

q_2

K_2

	a			
	-	-	-	-
	₀	₁	₅	₄
	-	-	-	-
	₂	₃	₇	₆
b	1	0	1	1
	₁₂	₁₃	₁₇	₁₆
	1	0	1	1
	₁₀	₁₁	₁₅	₁₄
	q_1			

q_2

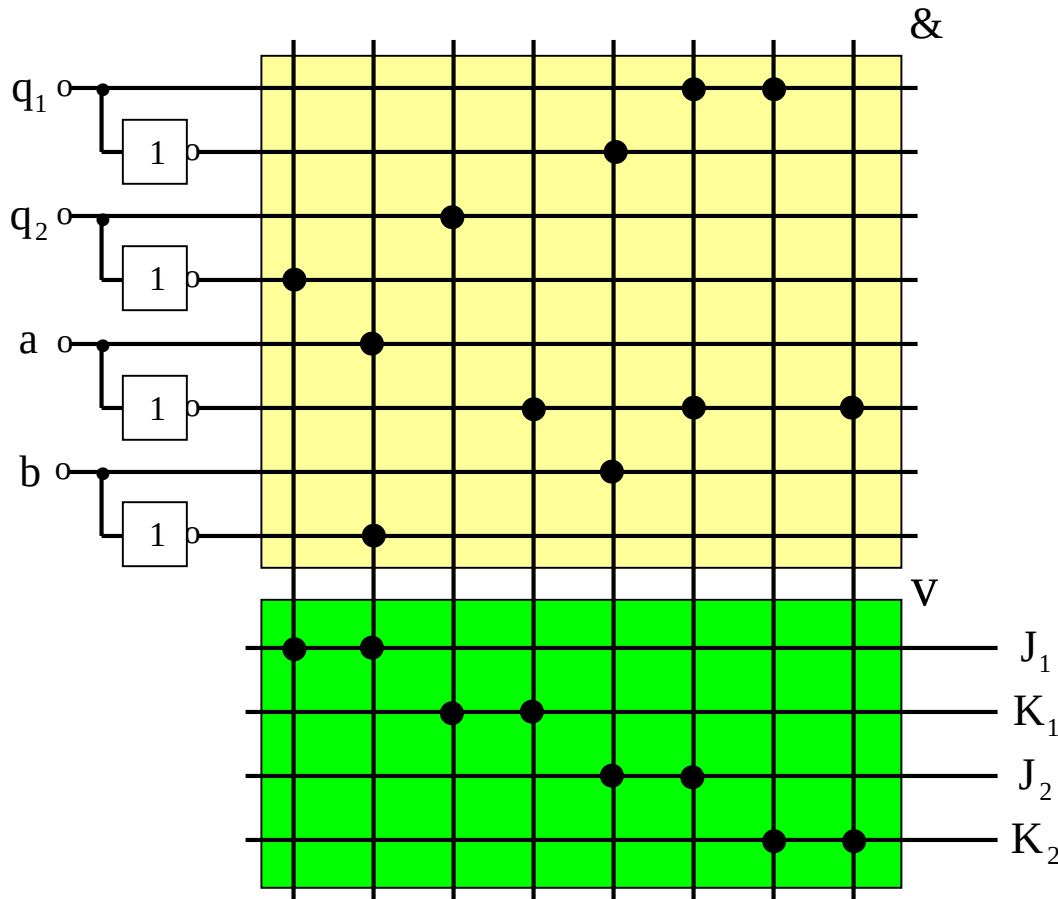
Q^n		X		Q^{n+1}		FF2		FF1	
q_2^n	q_1^n	b	a	q_2^{n+1}	q_1^{n+1}	J_2	K_2	J_1	K_1
0	0	0	0	0	1	0	-	1	-
0	0	0	1	0	1	0	-	1	-
0	0	1	-	1	1	1	-	1	-
0	1	-	0	1	0	1	-	-	1
0	1	-	1	0	1	0	-	-	0
1	0	-	0	0	0	-	1	0	-
1	0	0	1	1	1	-	0	1	-
1	0	1	1	1	0	-	0	0	-
1	1	-	-	0	0	-	1	-	1

$$\underline{\underline{J_2 = \overline{q_1}b + q_1\overline{a}}}$$

$$\underline{\underline{K_2 = q_1 + \overline{a}}}$$

2. Aufgabe – 2.3

- **2.3** Realisieren Sie die Ansteuerfunktionen unter Verwendung eines PLA-Bausteins und zeichnen Sie den vollständigen Automaten.



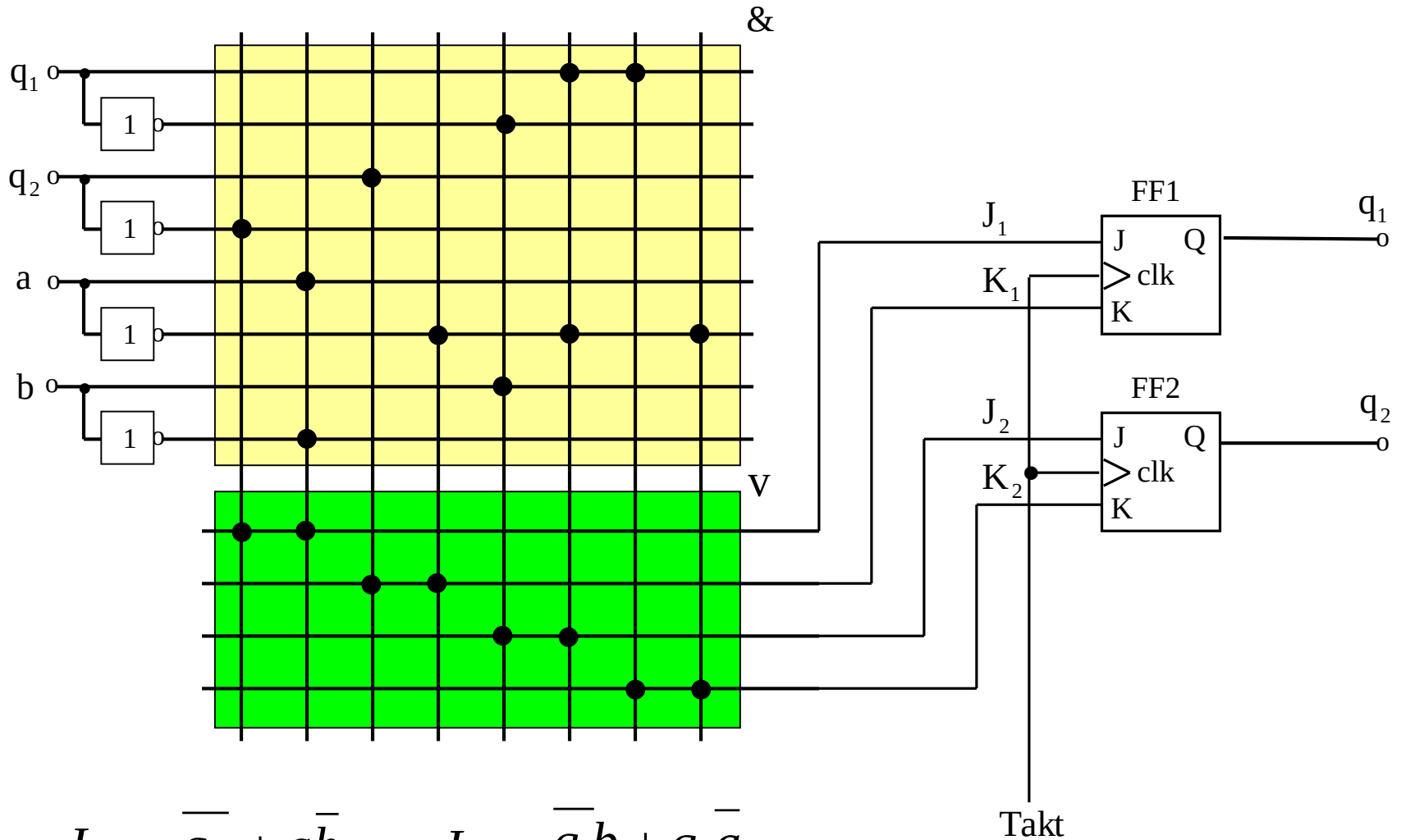
$$J_1 = \overline{q_2} + a\overline{b}$$

$$K_1 = \overline{q_2} + a$$

$$J_2 = \overline{q_1}b + q_1\overline{a}$$

$$K_2 = \overline{q_1} + a$$

2. Aufgabe – 2.3



$$J_1 = \overline{q_2} + a\overline{b} \quad J_2 = \overline{q_1}b + q_1\overline{a}$$

$$K_1 = q_2 + \overline{a} \quad K_2 = q_1 + \overline{a}$$

Fabian Kempf

kempf@kit.edu

Karlsruher Institut für Technologie (KIT)

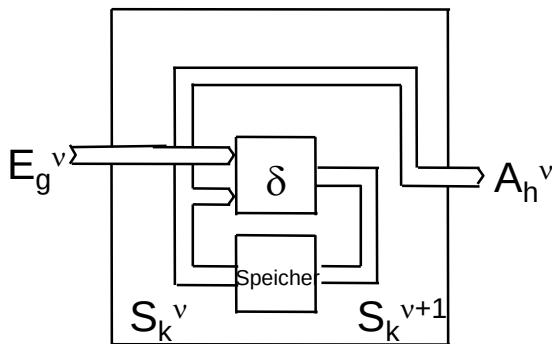
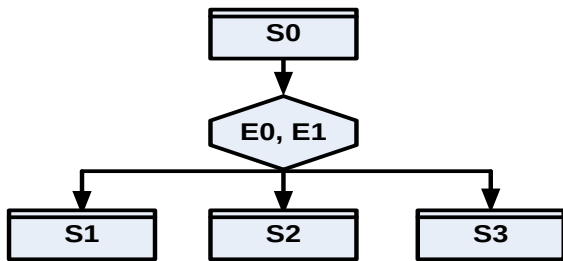
Institut für Technik der Informationsverarbeitung (ITIV)

6. Aufgabe

Medwedew-Automat

$$A_h^v = S_k^v$$

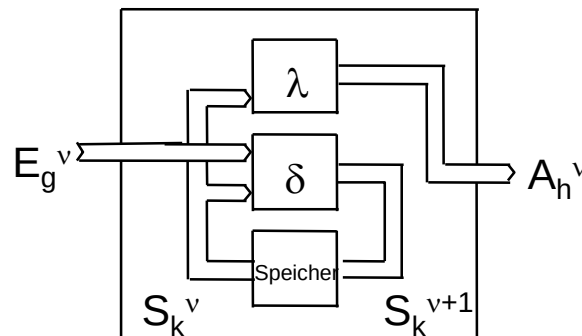
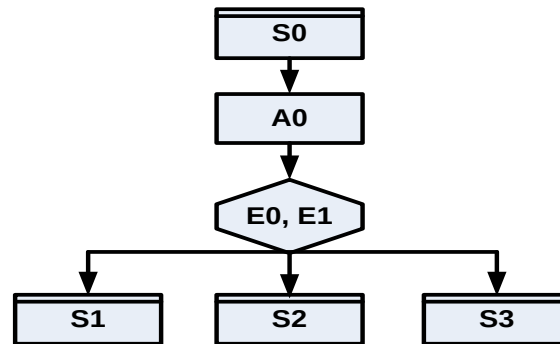
Die Ausgabe des Automaten ist **identisch mit dem aktuellen Zustand**



Moore-Automat

$$A_h^v = \lambda (S_k^v)$$

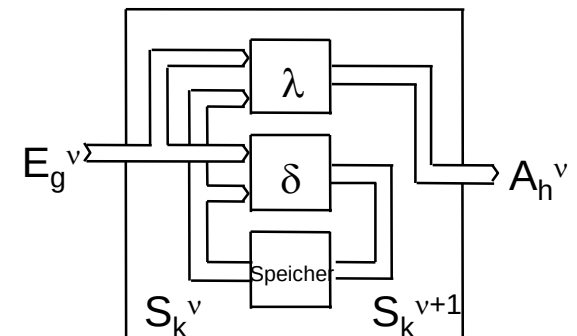
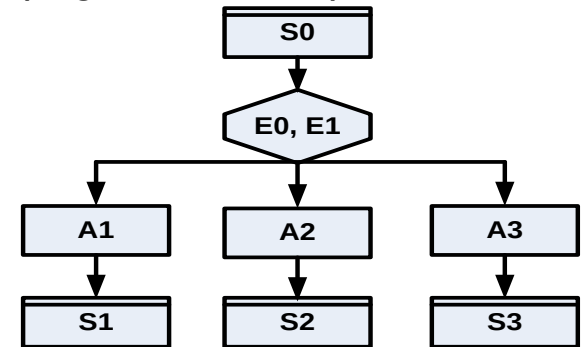
Die Ausgabe des Automaten ergibt sich ausschließlich aus dem **aktuellen Zustand**



Mealy-Automat

$$A_h^v = \lambda (E_g^v, S_k^v)$$

Die Ausgabe des Automaten ergibt sich aus **aktuellem Zustand** und einer **Eingabe** (allgemeiner Fall)



6. Aufgabe

- Für einen Vergnügungspark sollen Sie einen sprechenden Papageien entwerfen. Der Papagei versucht, dem Kunden Wundereier zum Preis von je 1€ aufzuschwatzen. Dies erfolgt mittels eines Endlos-Tonbandes, das jeweils nur im Grundzustand laufen soll. Der Kunde signalisiert seine Kaufbereitschaft durch Betätigen einer „Ja“-Taste. Dadurch öffnet sich erst der Geldeinwurf. Es können Münzen zu 0,50 € und 1 € eingeworfen werden. Andere Münzen oder Überzahlungen werden einbehalten. Ist mindestens 1 € bezahlt, öffnet sich der Warenschacht, das Ei wird ausgeworfen und der Geldeinwurf geschlossen. Anschließend erfolgt automatisch die Rückkehr in den Grundzustand. Ein Betätigen der „Ja“-Taste führt stets, gegebenenfalls unter Verlust bereits gezahlter Beträge, in den Zustand „Geldeinwurf“.
- Entwerfen Sie den Automatengraphen für eine Realisierung als Moore-Automat.

6. Aufgabe

■ Eingangsvariablen:

- JA 1: “JA”-Taste gedrückt
0: “JA”-Taste nicht gedrückt
- M1 1: es wurde ein 50Ct-Stück eingeworfen
0: es wurde kein 50Ct-Stück eingeworfen
- M2 1: es wurde ein 1 €-Stück eingeworfen
0: es wurde kein 1 €-Stück eingeworfen

■ Ausgangsvariablen:

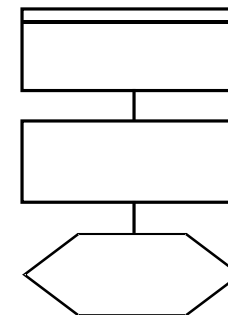
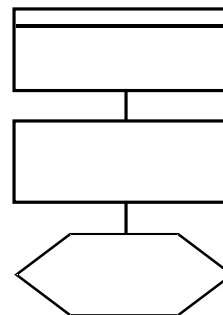
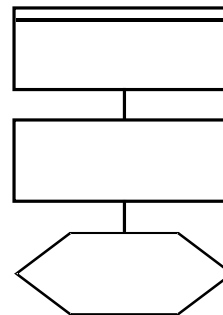
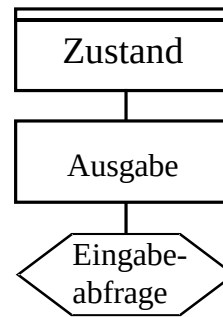
- E 1: es wird ein Ei ausgegeben
0: es wird kein Ei ausgegeben
- B 1: das Tonband läuft
0: das Tonband läuft nicht
- G 1: der Geldeinwurf ist geöffnet
0: der Geldeinwurf ist geschlossen

■ Zustandsnamen:

- “Grundzustand”, “Geldeinwurf”, “0.50 Ct”, “1 €”
- Verwenden Sie für weitere Zustände erklärende Namen

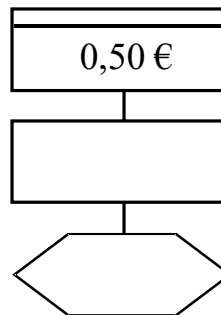
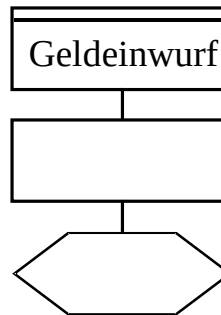
6. Aufgabe

- Moore-Automat
- 4 Zustände



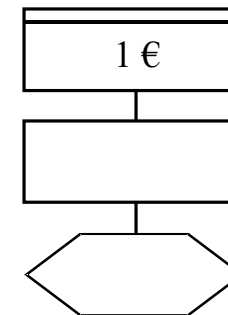
6. Aufgabe

- Moore-Automat
- 4 Zustände



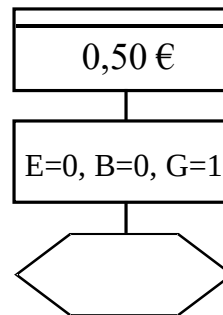
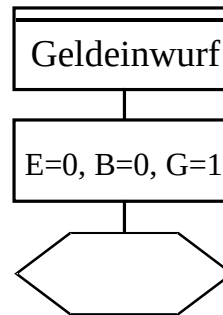
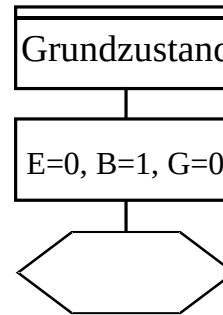
- Zustandsnamen:

- "Grundzustand"
- "Geldeinwurf"
- "0.50 Ct"
- "1 €"

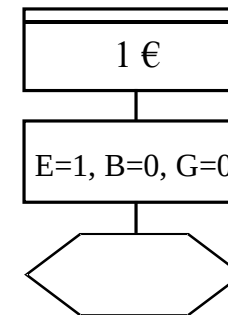


6. Aufgabe

- Moore-Automat
- 4 Zustände
- Ausgaben festlegen

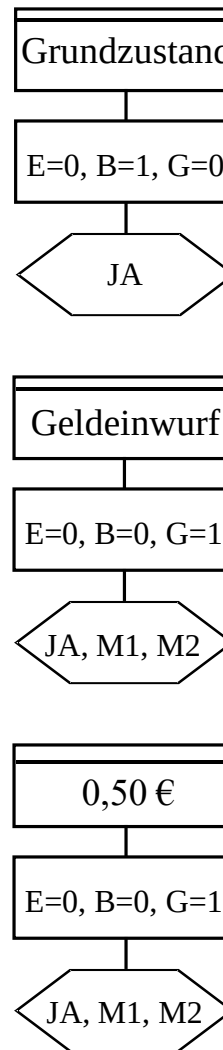


- Ausgangsvariablen:
 - E 1: ein Ei ausgegeben
E 0: kein Ei ausgegeben
 - B 1: Tonband läuft
B 0: Tonband läuft nicht
 - G 1: Geldeinwurf ist geöffnet
G 0: Geldeinwurf ist geschlossen



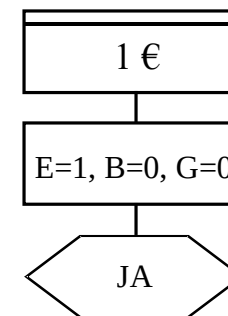
6. Aufgabe

- Moore-Automat
- 4 Zustände
- Ausgaben festlegen
- Festlegen der relevanten Abfragen



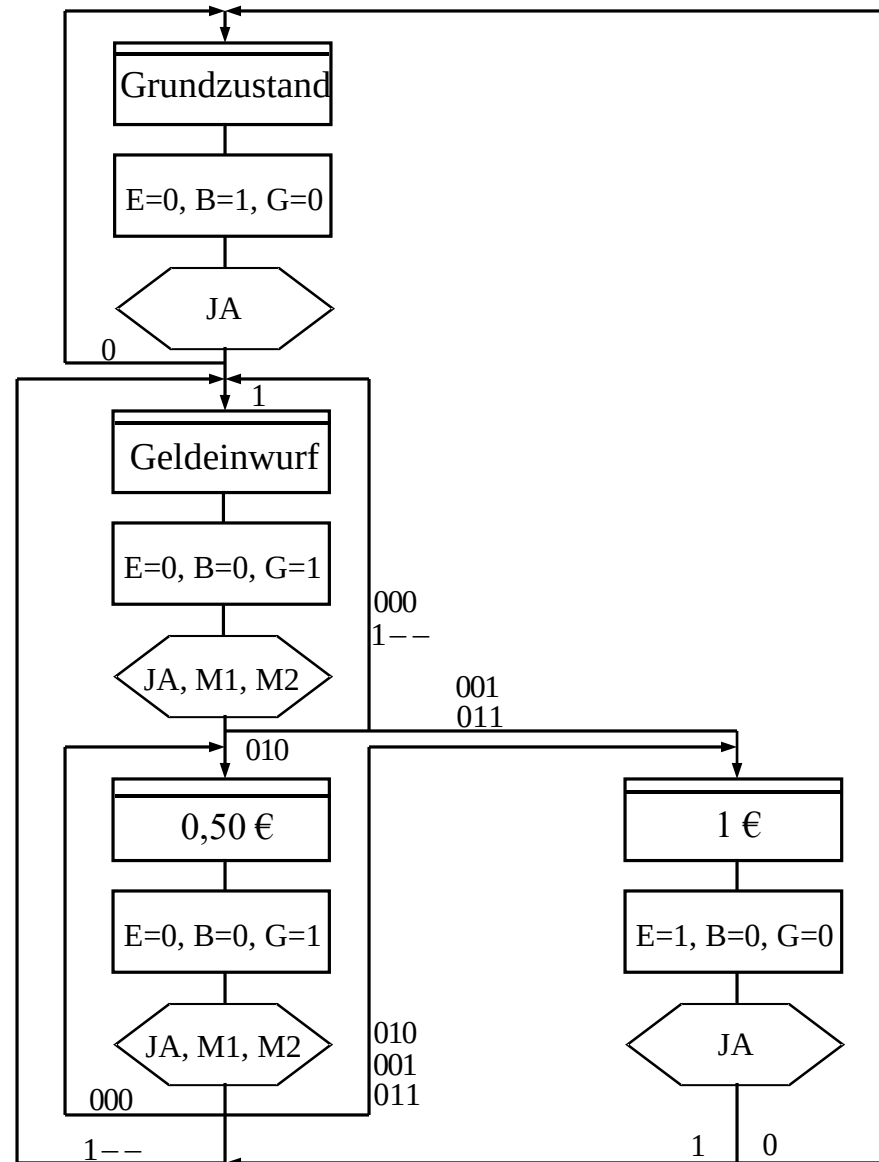
■ Eingangsvariablen:

- Ja 1/0:
"JA"-Taste gedrückt
- M1 1/0:
50Ct eingeworfen
- M2 1/0:
1€ eingeworfen



6. Aufgabe

- Moore-Automat
- 4 Zustände
- Ausgaben festlegen
- Festlegen der relevanten Abfragen
- Übergänge + Bedingungen einzeichnen



Fabian Kempf

kempf@kit.edu

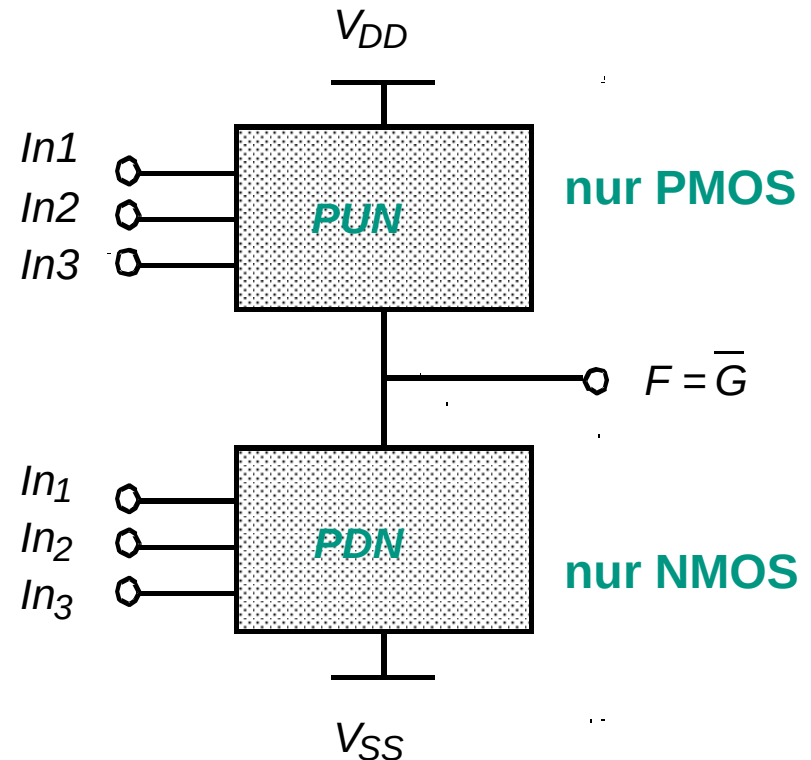
Karlsruher Institut für Technologie (KIT)

Institut für Technik der Informationsverarbeitung (ITIV)

7. Aufgabe

PUN und PDN -> duale (komplementäre) Netze:

- Die CMOS-Technologie basiert auf komplementären Netzen
→ **PUN** und **PDN**
(Pull-Up/Pull-Down Netz)
- Ist der **Ausgang** der realisierten Funktion gleich 1, schaltet das **PUN-Netz** den **Ausgang F** auf V_{DD}
- Ist der **Ausgang** hingegen gleich 0, schaltet das **PDN-Netz** den **Ausgang F** auf V_{SS}
- Zur Realisierung der **PUN-Netze** werden **PMOS-Transistoren** eingesetzt, **PDN-Netze** werden mit **NMOS-Transistoren** implementiert
- Da außer zu Schaltzeitpunkten keine direkte Verbindung zwischen V_{DD} und V_{SS} besteht, wird **während des Haltens sehr wenig Energie verbraucht**

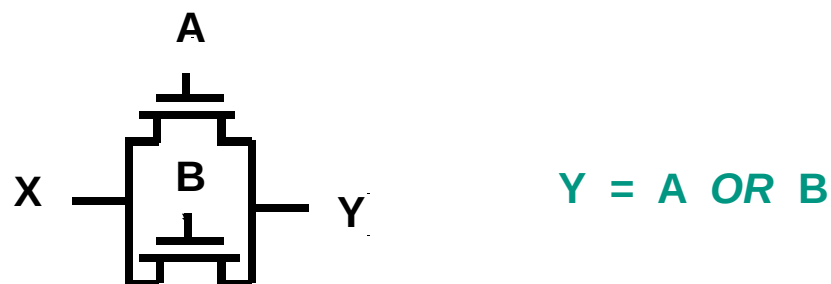


NMOS-Transistoren in Serien/Parallelschaltung

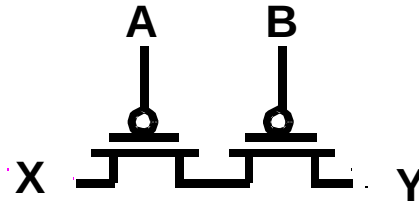
Transistoren → Schalterrealisierung

→ Steuerung durch Gate-Signal

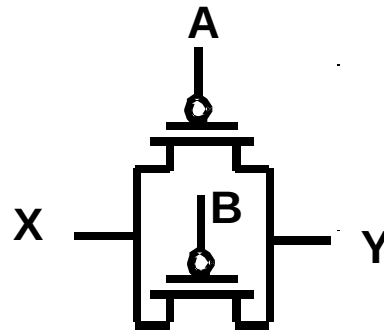
NMOS-Schalter schließt, wenn das Gate auf “high” (H, “1”) ist



PMOS-Schalter schließt, wenn das Gate auf “low” (L, “0”) ist



$$Y = \overline{A} \text{ AND } \overline{B} = \overline{A + B}$$

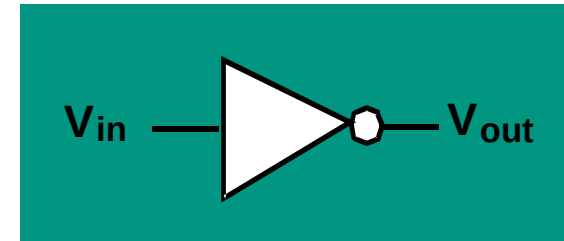
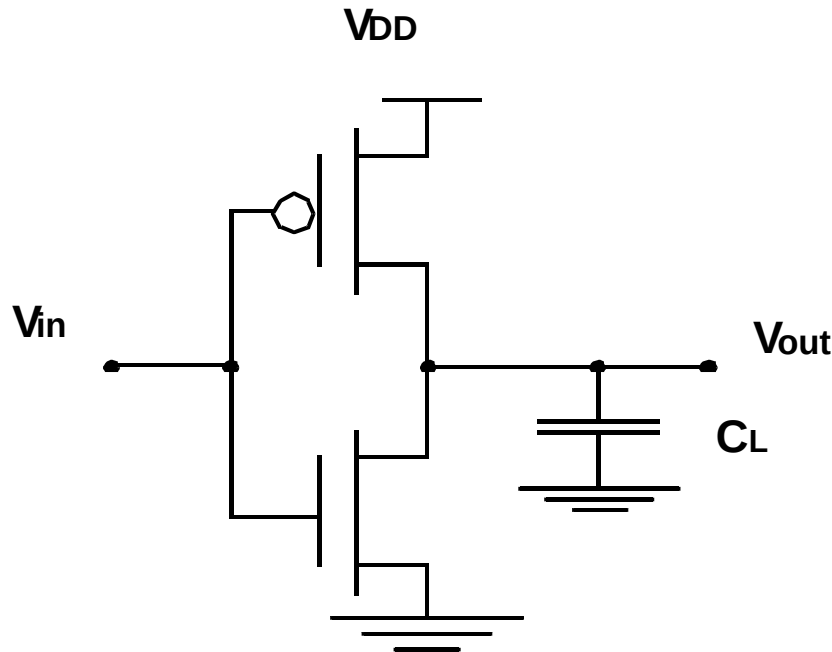


$$Y = \overline{A} \text{ OR } \overline{B} = \overline{AB}$$

NMOS-, PMOS-Schaltungen: Anbindung an Axiome der Schaltalgebra

→ Realisierung der Konjunktion, Disjunktion und Negation

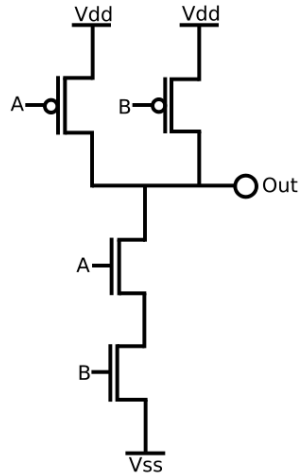
CMOS: *Inverter*-Schaltung



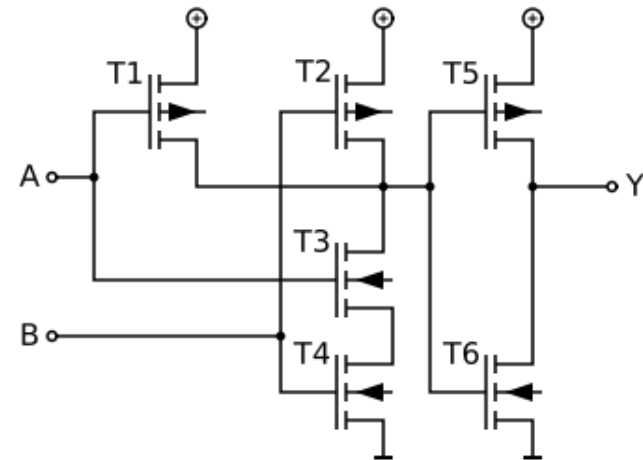
- Der **Inverter** besteht aus **zwei Transistoren**, einem NMOS und einem PMOS
- Ist V_{in} auf **high**, so **öffnet** der **NMOS-Transistor** und V_{out} wird auf **GND** gezogen, während dessen ist der **PMOS-Transistor** **gesperrt**
- Ist V_{in} hingegen auf **low**, so **öffnet** der **PMOS-Transistor** und V_{out} wird auf **V_{DD}** gezogen, der **NMOS-Transistor** wird hingegen **gesperrt**
- Strom zwischen V_{DD} und GND kann nur fließen, wenn die Transistoren umschalten und kurzzeitig beide offen sind

Einfach CMOS Grundsaltungen

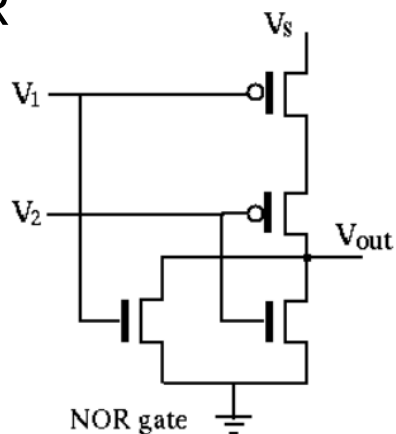
NAND



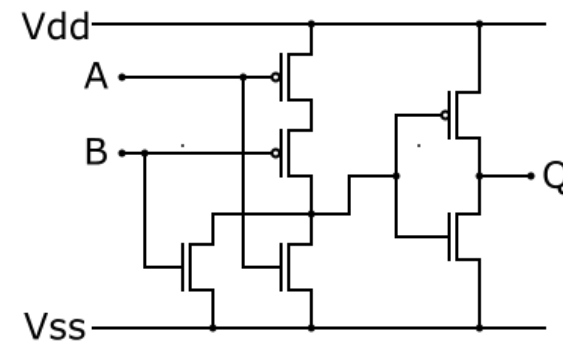
AND



NOR

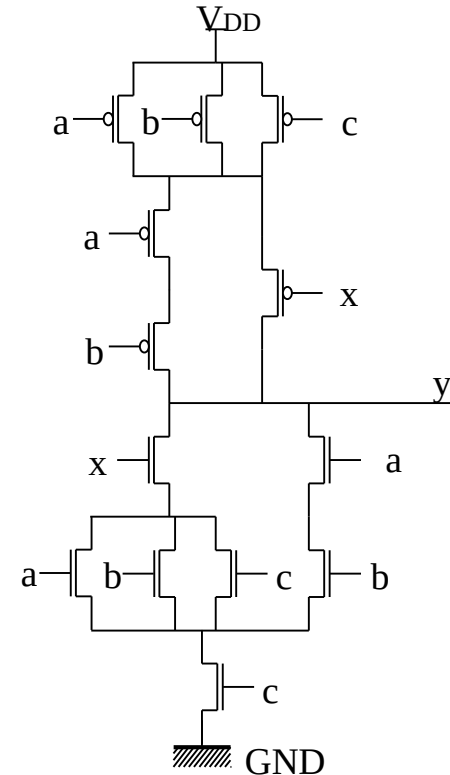
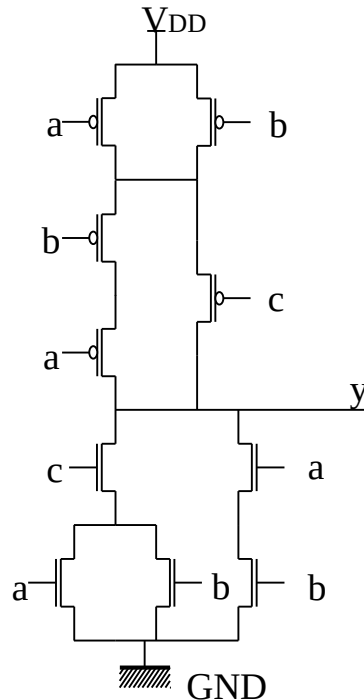


OR



7. Aufgabe

■ Gegeben seien die folgenden CMOS-Gatternetze:



■ Bestimmen Sie die logischen Funktionen der p- und n-Netze. Überprüfen Sie, ob die logischen Funktionen wohldefiniert sind. Geben Sie alle Eingangskombinationen für eventuelles Fehlverhalten an.

7. Aufgabe

- P-Netz (v_1) stellt Verbindung zu V_{DD} her
- N-Netz (v_0) stellt Verbindung zu GND her
- Forderungen an fehlerfreies und wohldefiniertes Verhalten:

1. $v_0 \cdot v_1 = 0$ (kein Kurzschluss)
2. $v_0 + v_1 = 1$ (Vollständigkeit)

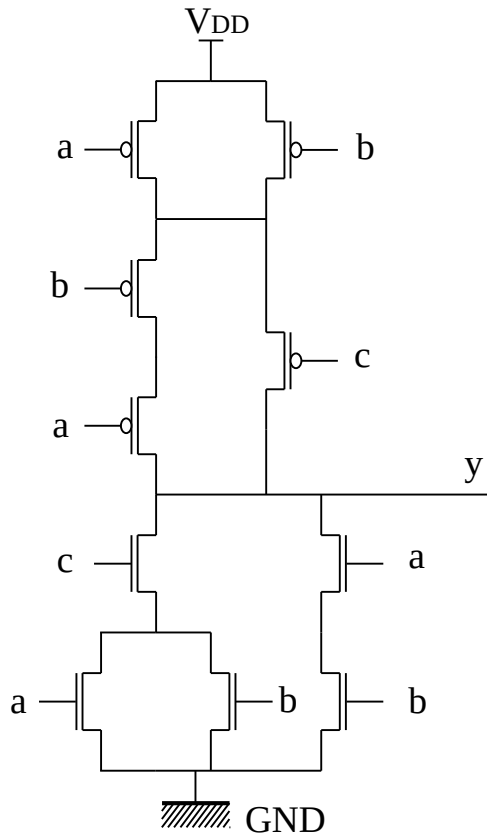
oder auch wenn Dualität gilt:

$$v_0 = \overline{v_1} \text{ (wohl definiert)}$$

- Bedeutung im Symmetriediagramm:
 - Keine Überlappung von Einsen und Nullen (1)
 - Alle Felder im S-Diagramm sind belegt (2)
(kein undefiniertes Verhalten)

7. Aufgabe

■ Teil 1



■ P-Netz:

$$v_1 = (\bar{a} + \bar{b}) \cdot (\bar{a}\bar{b} + \bar{c})$$

■ N-Netz:

$$v_0 = (ab) + (c \cdot (a + b))$$

■ Forderung:

$$v_0 = \bar{v}_1$$

$$\bar{v}_1 = \overline{(\bar{a} + \bar{b}) \cdot (\bar{c} + \bar{a}\bar{b})}$$

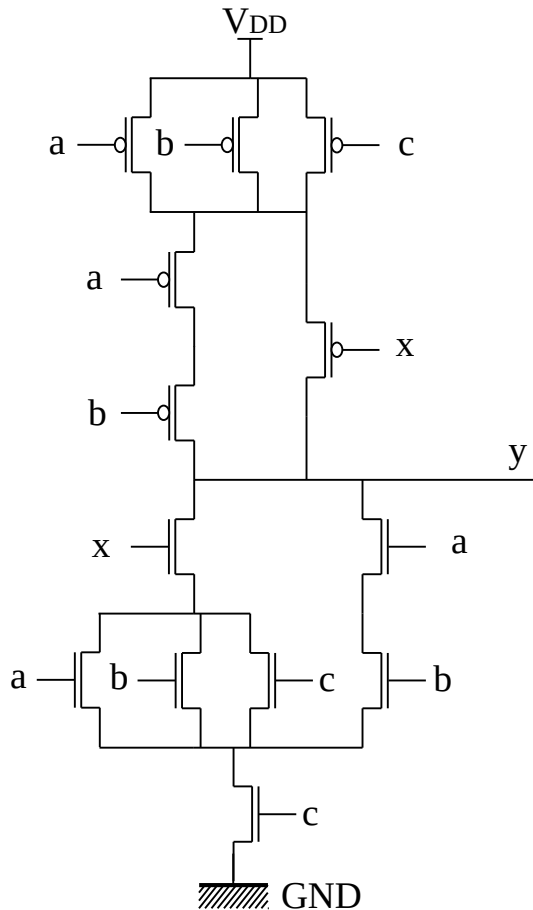
$$= (ab) + \overline{\bar{c} + \bar{a}\bar{b}}$$

$$= (ab) + (c \cdot (a + b)) = v_0$$

erfüllt, daher ist die Funktion wohldefiniert!!!

7. Aufgabe

Teil 2



■ P-Netz:

$$v_1 = (\bar{a} + \bar{b} + \bar{c}) \cdot (\bar{x} + \bar{a}\bar{b})$$

■ N-Netz:

$$v_0 = c \cdot (x \cdot (a + b + c) + ab)$$

■ Forderung:

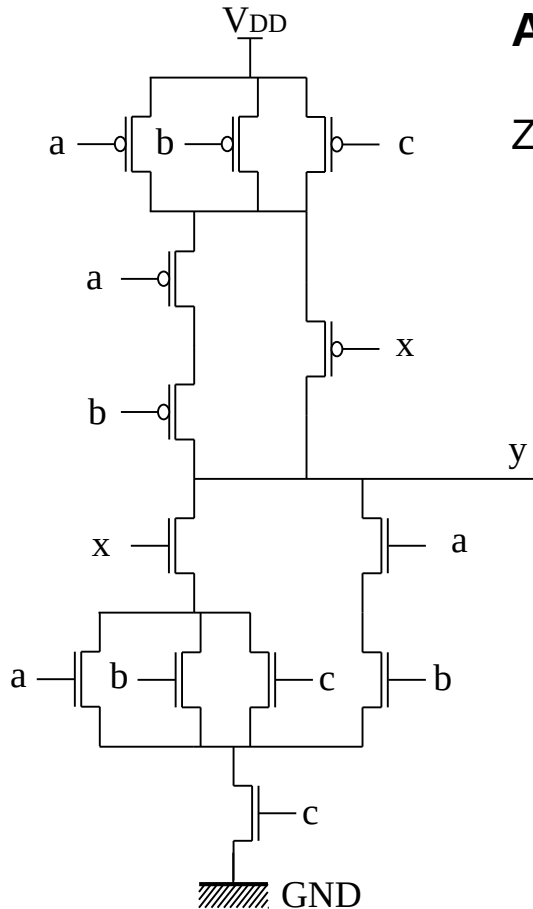
$$v_0 \cdot v_1 = 0$$

$$\begin{aligned} & ((\bar{a} + \bar{b} + \bar{c}) \cdot (\bar{x} + \bar{a}\bar{b})) \cdot c \cdot (x \cdot (a + b + c) + ab) \\ &= (\bar{a}\bar{x} + \bar{b}\bar{x} + \bar{c}\bar{x} + \bar{a}\bar{b} + \bar{a}\bar{b}\bar{c}) \cdot c \cdot (ax + bx + cx + ab) \\ &= (\bar{a}\bar{x}c + \bar{b}\bar{x}c + \bar{a}\bar{b}c) \cdot (ax + bx + cx + ab) \\ &= \underline{\underline{\bar{a}\bar{b}c x}} \end{aligned}$$

bereits die erste Bedingung ist verletzt, ein Konflikt liegt vor !!!

7. Aufgabe

Teil 2



Alle Eingangskombinationen mit Fehlverhalten finden:

Zwei Lösungsmöglichkeiten:

1. Algebraisch:

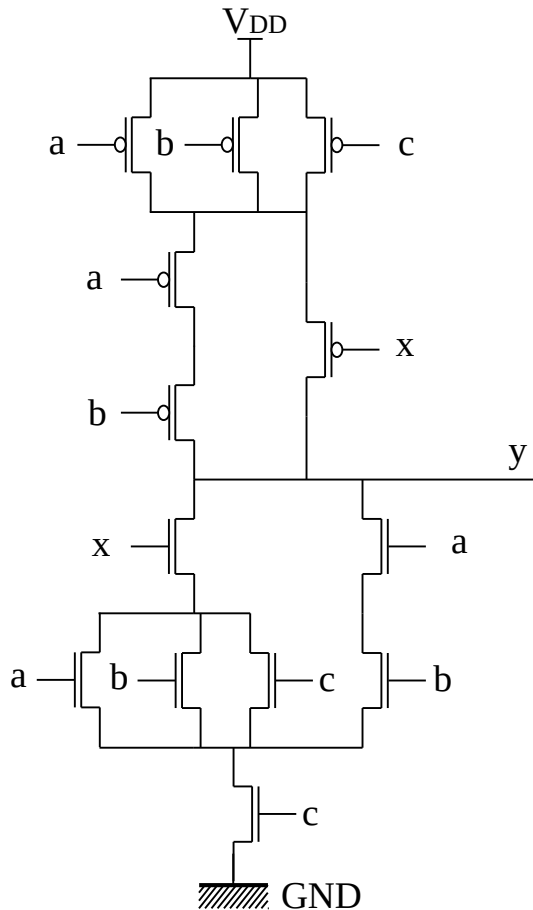
- Schaltung ist nicht wohldefiniert wenn: $v_0 \neq \overline{v_1}$
- Kurzschlüsse bei: $v_0 \cdot v_1 \neq 0$
- Undefiniertheit bei: $v_0 + v_1 \neq 1 \rightarrow \overline{v_0 + v_1} \neq 0$

2. Wahrheitstabelle:

- Wahrheitstabelle für die Funktionen v_0 und v_1 aufstellen
- Kurzschluss bei: $v_0(a, b...) = v_1(a, b...) = 1$
- Undefiniertheit bei: $v_0(a, b...) = v_1(a, b...) = 0$

7. Aufgabe

Teil 2



■ Lösung mit Wahrheitstabelle:

■ P-Netz: $v_1 = (\bar{a} + \bar{b} + \bar{c}) \cdot (\bar{x} + \bar{a}\bar{b})$

■ N-Netz: $v_0 = c \cdot (x \cdot (a + b + c) + ab)$

x	c	b	a	v_1	v_0	y
0	0	0	0	1	0	1
0	0	0	1	1	0	1
0	0	1	0	1	0	1
0	0	1	1	1	0	1
0	1	0	0	1	0	1
0	1	0	1	1	0	1
0	1	1	0	1	0	1
0	1	1	1	0	1	0
1	0	0	0	1	0	1
1	0	0	1	0	0	Undefiniert
1	0	1	0	0	0	Undefiniert
1	0	1	1	0	0	Undefiniert
1	1	0	0	1	1	Kurzschluss
1	1	0	1	0	1	0
1	1	1	0	0	1	0
1	1	1	1	0	1	0

■ Kurzschluss für: $(x, c, b, a) = (1, 1, 0, 0)$

■ Undefiniert für: $(x, c, b, a) = (1, 0, 0, 1) \vee (1, 0, 1, 0) \vee (1, 0, 1, 1)$

Fabian Kempf

kempf@kit.edu

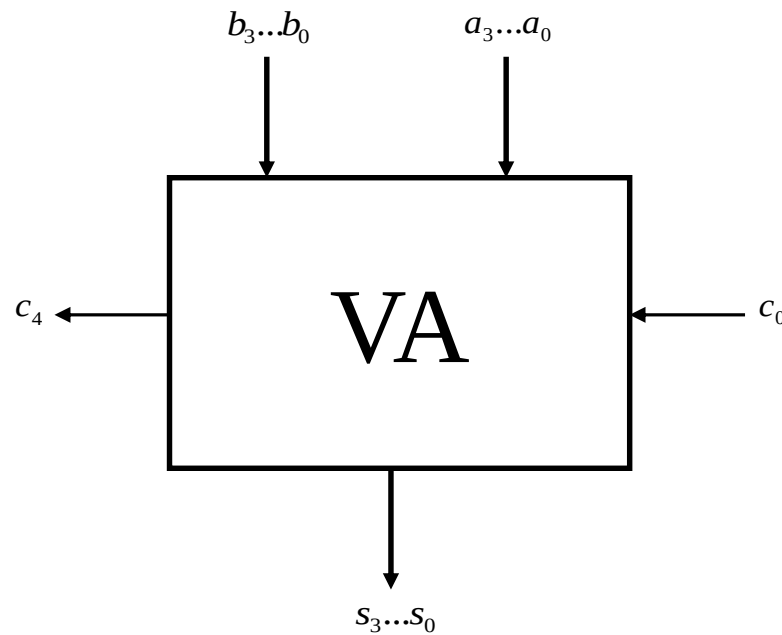
Karlsruher Institut für Technologie (KIT)

Institut für Technik der Informationsverarbeitung (ITIV)

4. Aufgabe

4. Aufgabe

- Ein 4-Bit-Volladdierer habe folgendes Blockschaltbild:

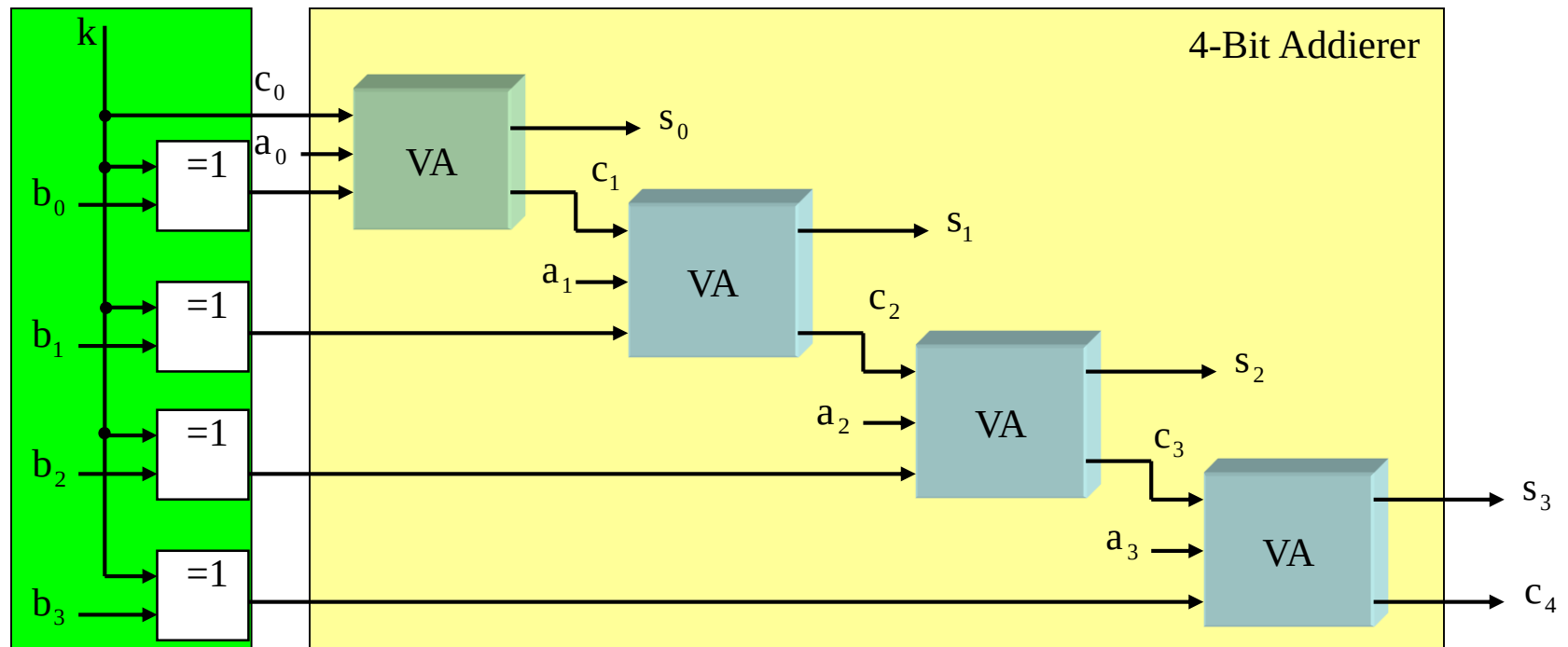


- 4.1** Konstruieren Sie aus dem angegebenen Blockschaltbild des 4-bit-VA und weiteren Gattern eine Schaltung, die für einen zusätzlichen Eingang $sub=1$ $a-b$ berechnet. Für $sub=0$ soll weiterhin $a+b$ berechnet werden.

4. Aufgabe

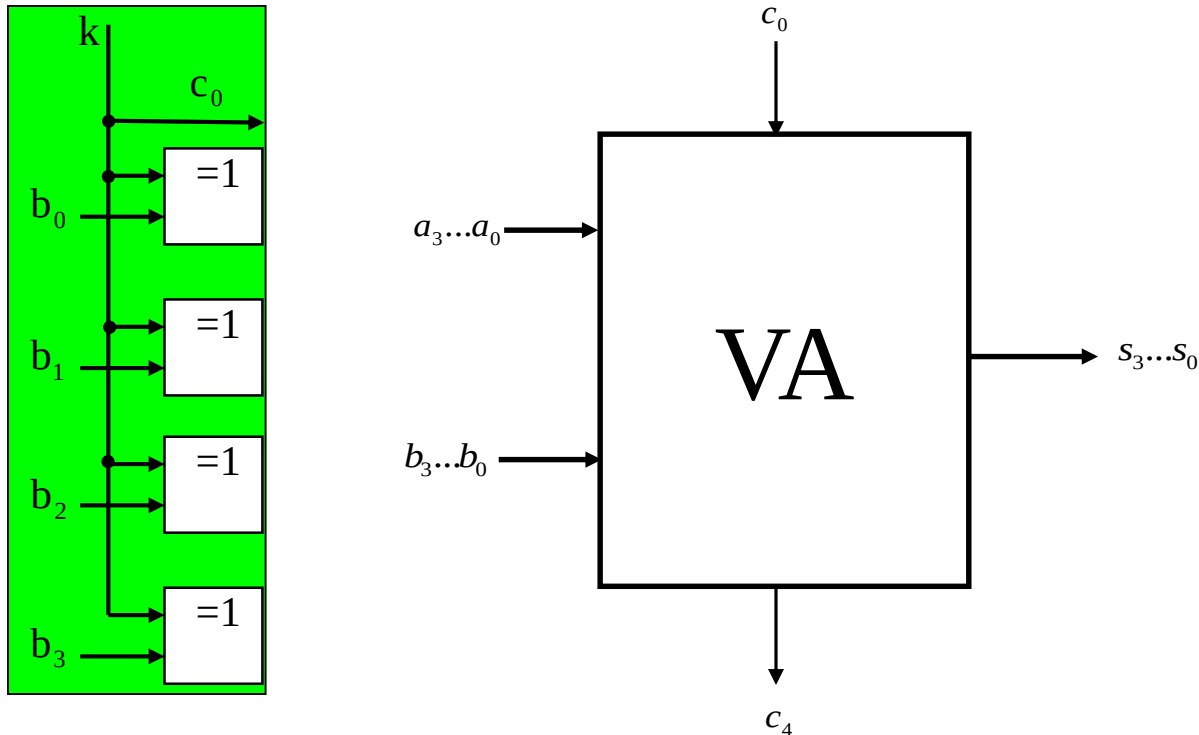
■ Erinnerung:

- Beispiel: 4-Bit Addierer/Subtrahierer
- Addierer mit $k=0$, Subtrahierer mit $k=1$



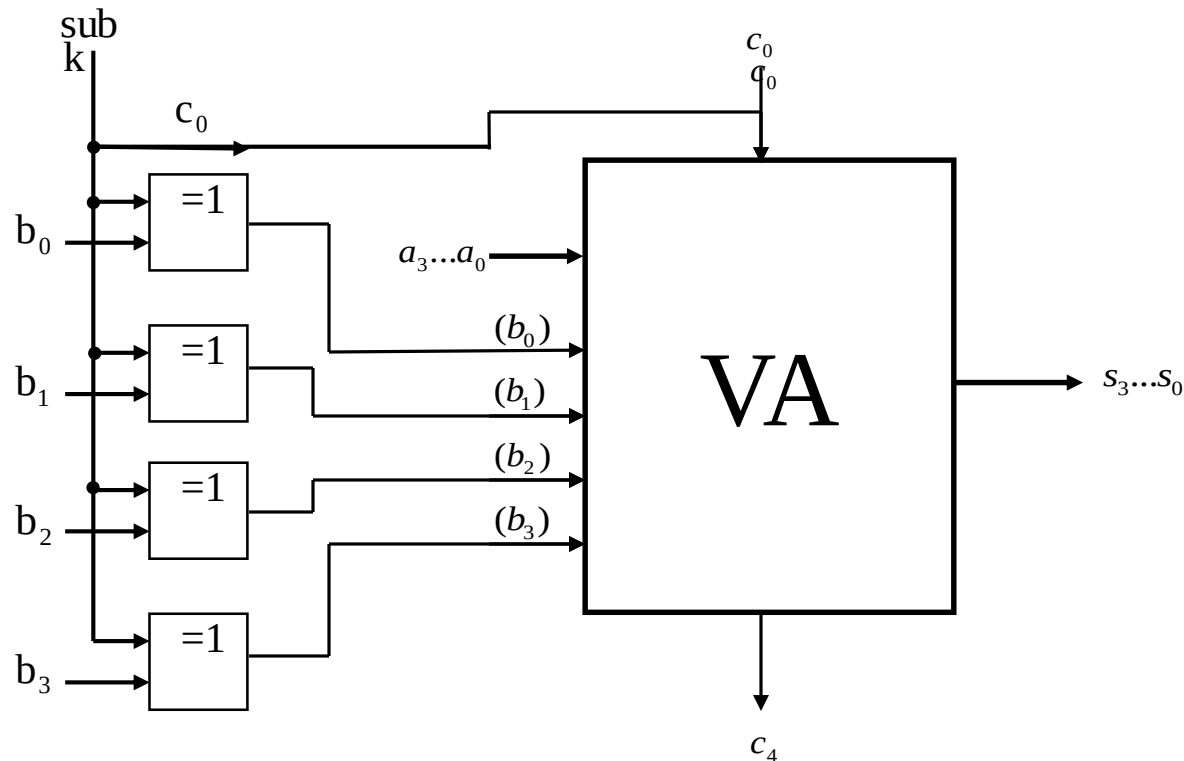
4. Aufgabe – 4.1

■ Übertragen des Verfahrens:



4. Aufgabe – 4.1

■ Übertragen des Verfahrens:



4. Aufgabe – 4.2

- 4.2 Konstruieren Sie eine Schaltung, die für den Fall, dass a und b jeweils in einer K2-Darstellung gegeben sind, einen Überlauf der Ergebnisse erkennt.
- die Summe aus einer negativen und einer positiven Zahl in der K2-Darstellung kann nie zu einem Überlauf führen.
 - Überlauf ist gegeben, wenn:
 1. beide Zahlen positiv sind, das Ergebnis jedoch negativ
 2. beide Zahlen negativ sind, das Ergebnis jedoch positive
 - Eine negative Zahl kann leicht an einer 1 an der höchstwertigen Stelle ($n-1$) erkannt werden.

4. Aufgabe – 4.2

■ Funktionstabelle:

beide Zahlen positiv, Ergebnis negativ

beide Zahlen negativ, Ergebnis positiv

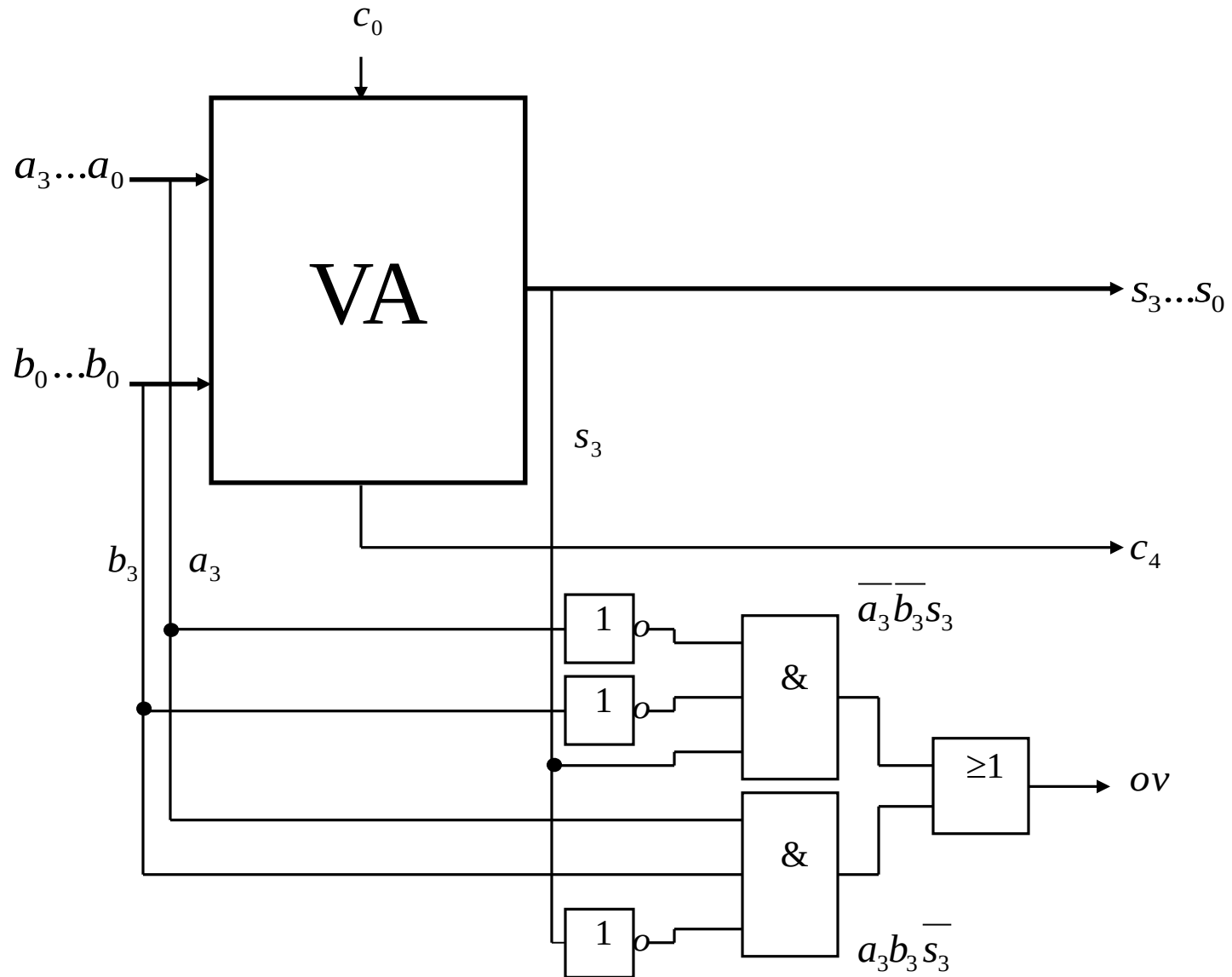
a_{n-1}	b_{n-1}	s_{n-1}	ov
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

$$ov = \overline{a_{n-1}} \overline{b_{n-1}} s_{n-1} + a_{n-1} b_{n-1} \overline{s_{n-1}}$$

Für den 4-Bit-Addierer:

$$ov = \overline{a_3} \overline{b_3} s_3 + a_3 b_3 \overline{s_3}$$

4. Aufgabe – 4.2



Fabian Kempf

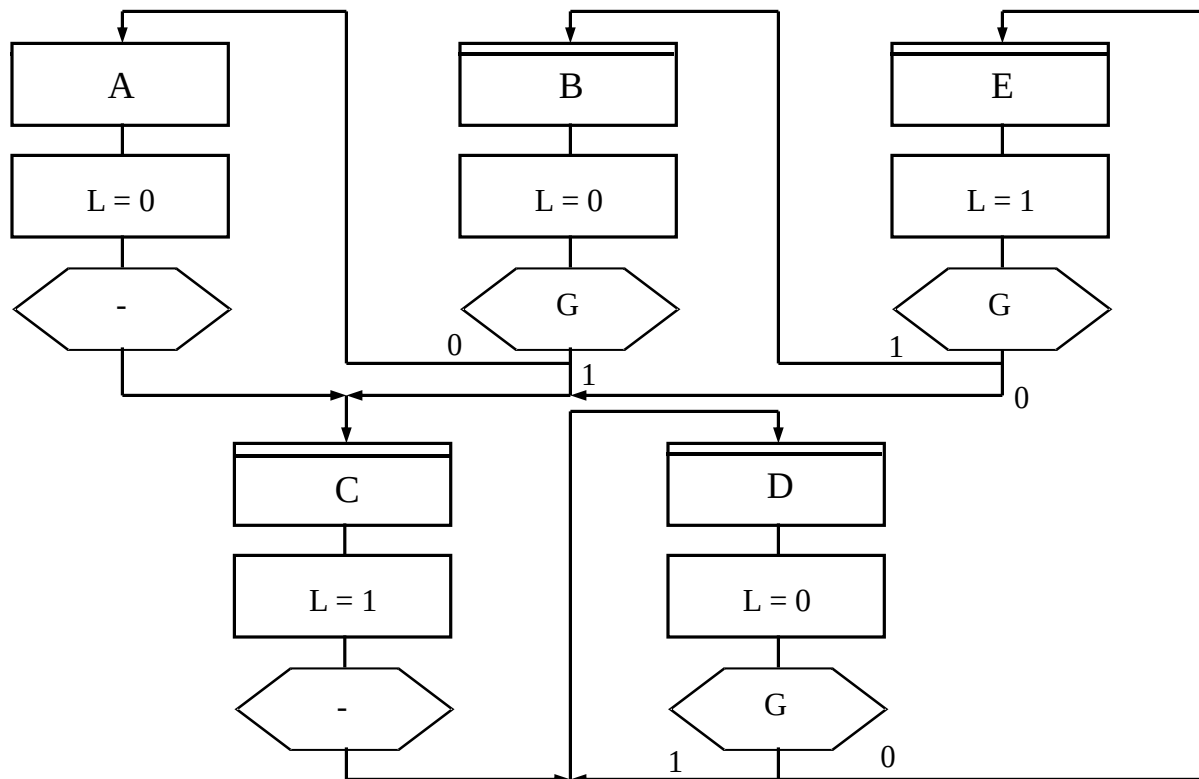
kempf@kit.edu

Karlsruher Institut für Technologie (KIT)

Aufgabe F2: Automaten

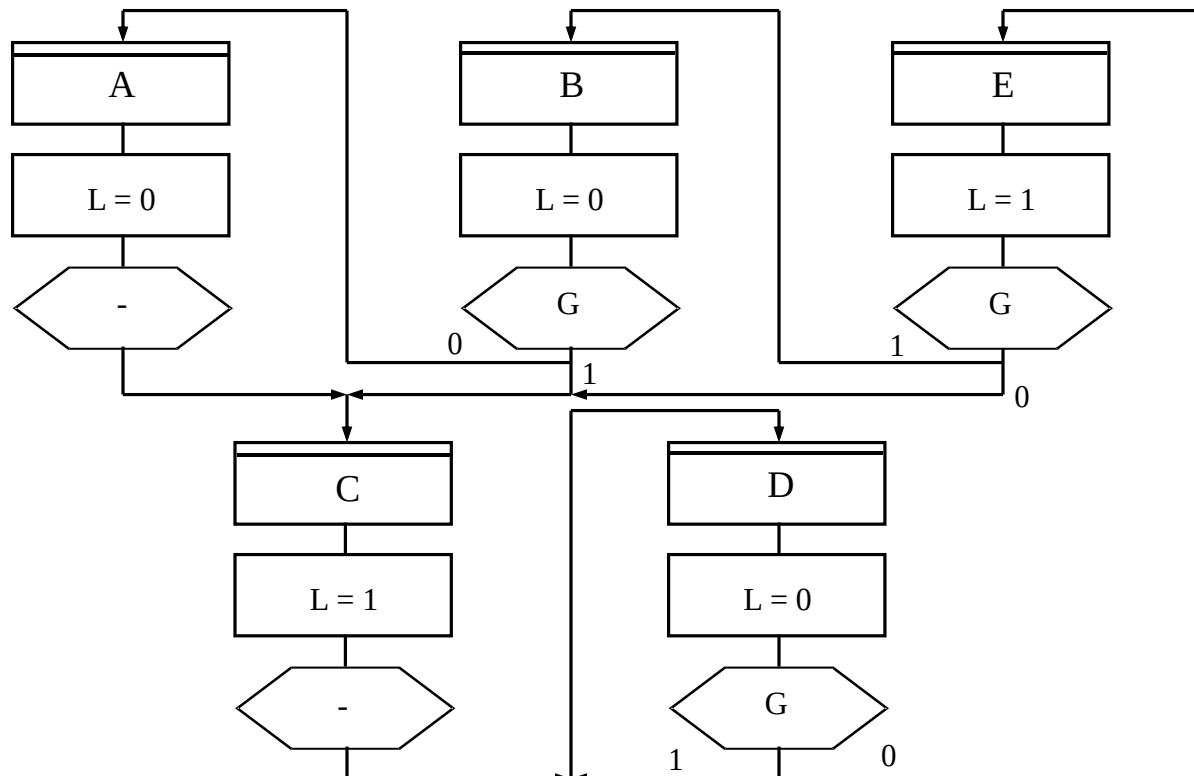
Aufgabe F2

Zur Realisierung eines Automaten haben zwei GDT-Studenten ein Ablaufdiagramm (Abbildung) bzw. eine Ablauftabelle (Tabelle) entwickelt. Leider wurde die Ablauftabelle ein Opfer der aus einer früheren Klausuraufgabe bekannten Cafeteria-Kaffeeleckten und ist deshalb nur unvollständig wiedergegeben.



Aufgabe F2

- **F2.1** Handelt es sich bei dem in gezeigten Automaten um einen Moore- oder einen Mealy-Typ?



Moore, da Ausgabe nur von Zuständen abhängt !!!

■ **F2.2** Wie viele Flip-Flops würden Sie für die Realisierung des Automaten als synchrones Schaltwerk mindestens benötigen?

- wir haben 5 Zustände im Automaten
- für eine binäre Kodierung würden wir also

$$\lceil \lg 5 \rceil = 3 \text{ Bits benötigen !!!}$$

Aufgabe F2

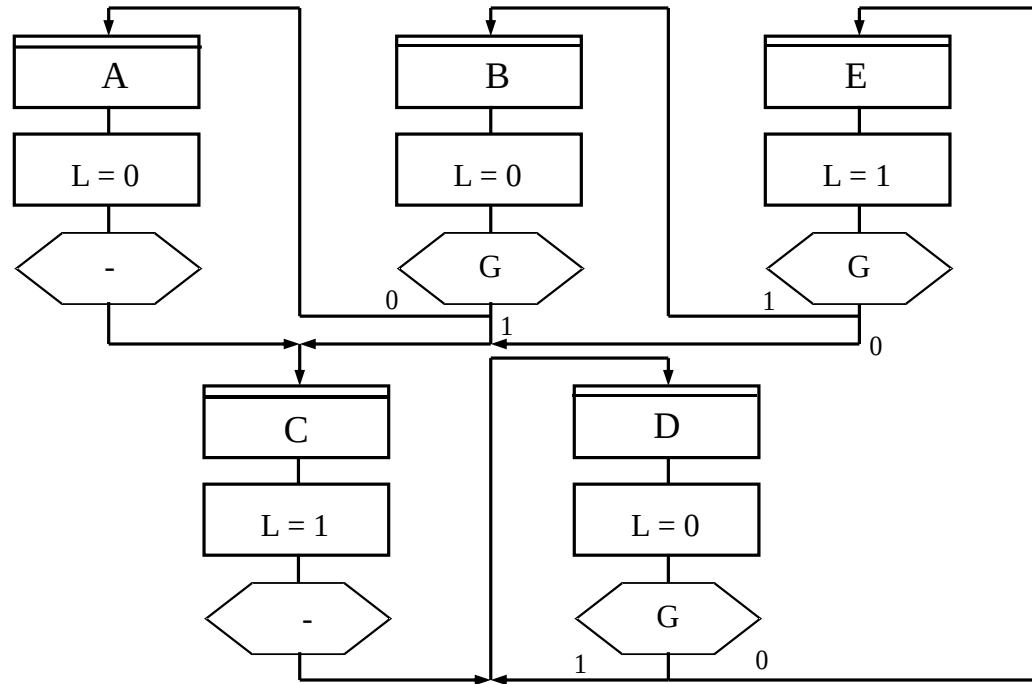
- **F2.3** Vervollständigen Sie nun die Ablaftabelle T1. Gehen Sie davon aus, dass die Ablaftabelle und das Ablaufdiagramm dasselbe Automatenverhalten beschreiben. Versuchen Sie eine geeignete Zuordnung zwischen den Bezeichnungen A, B, C, D, E, G und L und den Bezeichnungen R, U, W, Z, T, X und Y zu treffen. Geben Sie diese Zuordnung in einer Tabelle an.

S^n	X	S^{n+1}	Y
R	-		0
U	0 1	Z	
W	0 1	W	
Z	-	1	
T	0 1		

Aufgabe F2

S^n	X	S^{n+1}	Y
R	-		0
U	0		
	1	Z	
W	0		
	1	W	
Z	-		
T	0		
	1		

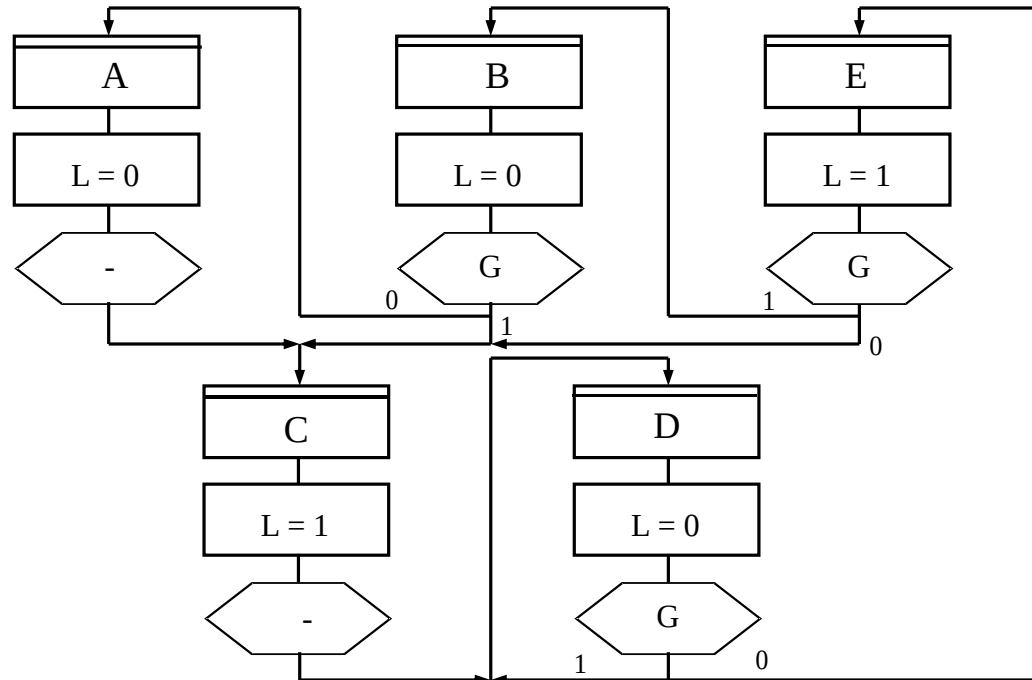
Diagramm	Tabelle
A	
B	
C	
D	
E	
G	
L	



Aufgabe F2

S^n	X	S^{n+1}	Y
R	-		0
U	0		
	1	Z	
W	0		
	1	W	
Z	-		
T	0		
	1		

Diagramm	Tabelle
A	
B	
C	
D	
E	
G	X
L	

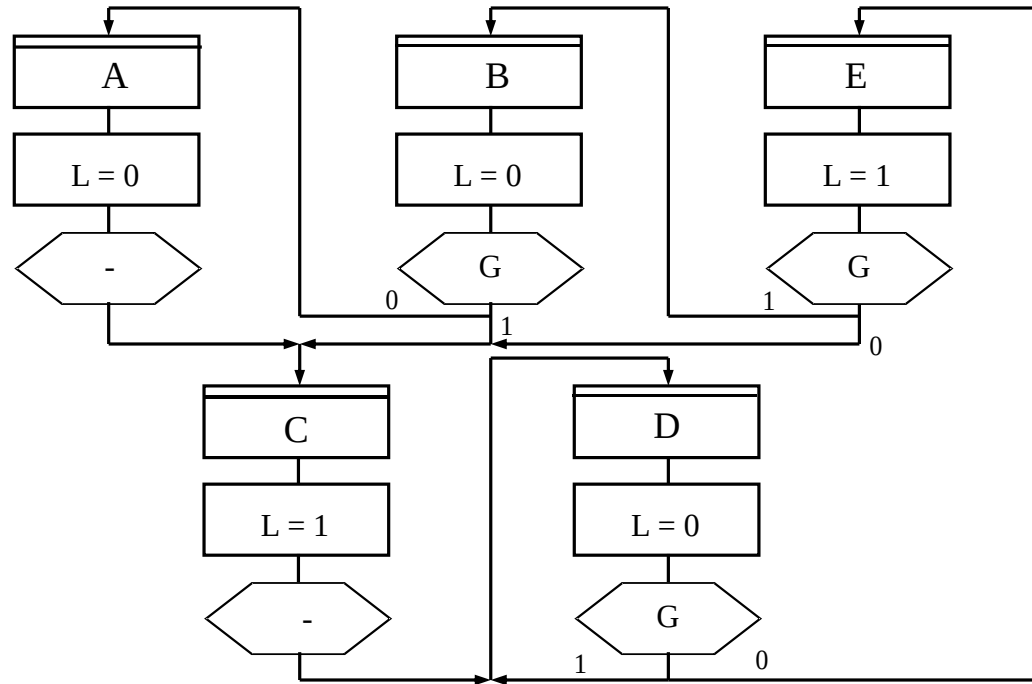


- G ist die Eingabe des Ablaufdiagramms
 $\Rightarrow G \equiv X$

Aufgabe F2

S^n	X	S^{n+1}	Y
R	-		0
U	0		
	1	Z	
W	0		
	1	W	
Z	-		
T	0		
	1		

Diagramm	Tabelle
A	
B	
C	
D	
E	
G	X
L	Y



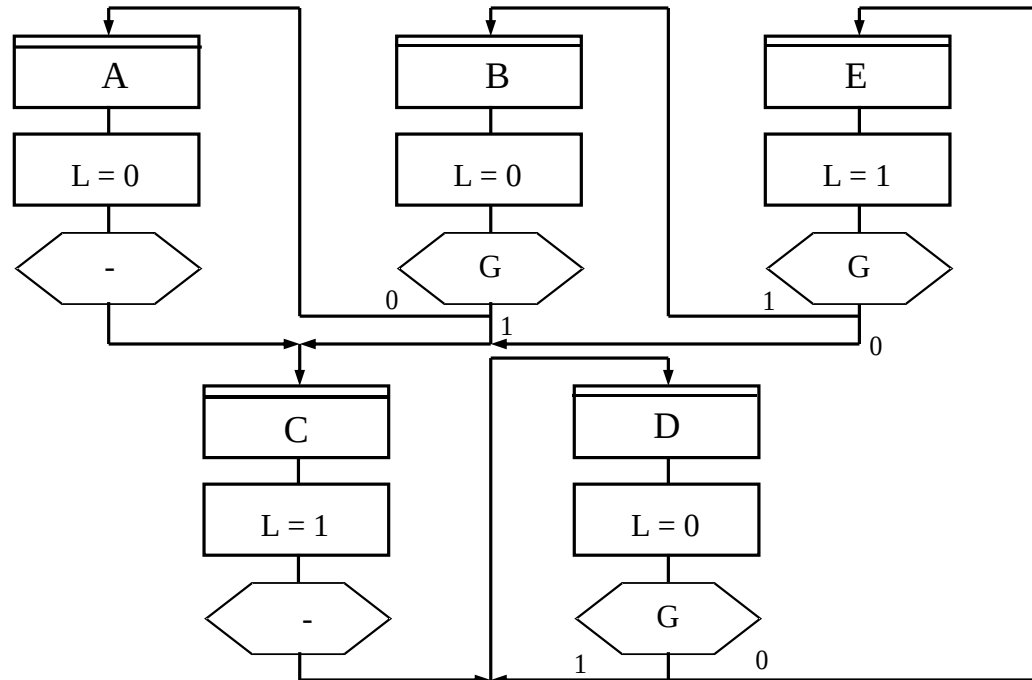
- L ist die Ausgabe des Ablaufdiagramms

$$\Rightarrow L \equiv Y$$

Aufgabe F2

S^n	X	S^{n+1}	Y
R	-		0
U	0		
	1	Z	
W	0		
	1	W	
Z	-		
T	0		
	1		

Diagramm	Tabelle
A	
B	
C	
D	W
E	
G	X
L	Y

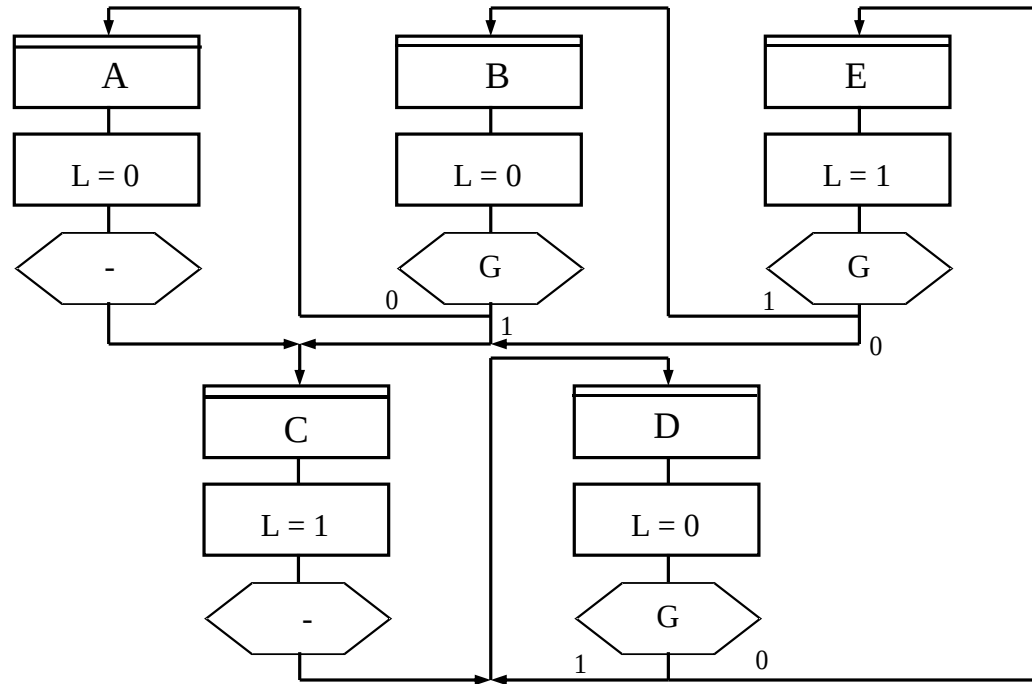


- D hat nur sich selbst und E als Nachfolgezustand!
 $\Rightarrow D \equiv W$

Aufgabe F2

S^n	X	S^{n+1}	Y
R	-		0
U	0		
	1	Z	
W	0		
	1	W	
Z	-		
T	0		
	1		

Diagramm	Tabelle
A	R
B	
C	
D	W
E	
G	X
L	Y

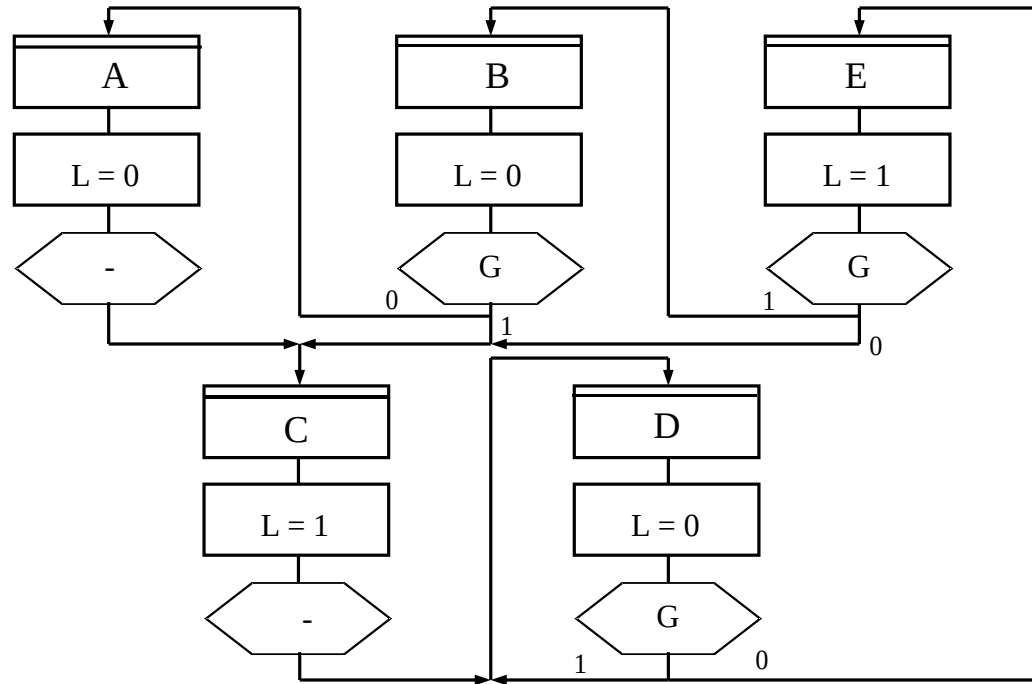


- A besitzt einen Nachfolgezustand und die Ausgabe = 0
 $\Rightarrow A \equiv R$

Aufgabe F2

S^n	X	S^{n+1}	Y
R	-		0
U	0		
	1	Z	
W	0		
	1	W	
Z	-		
T	0		
	1		

Diagramm	Tabelle
A	R
B	
C	Z
D	W
E	
G	X
L	Y

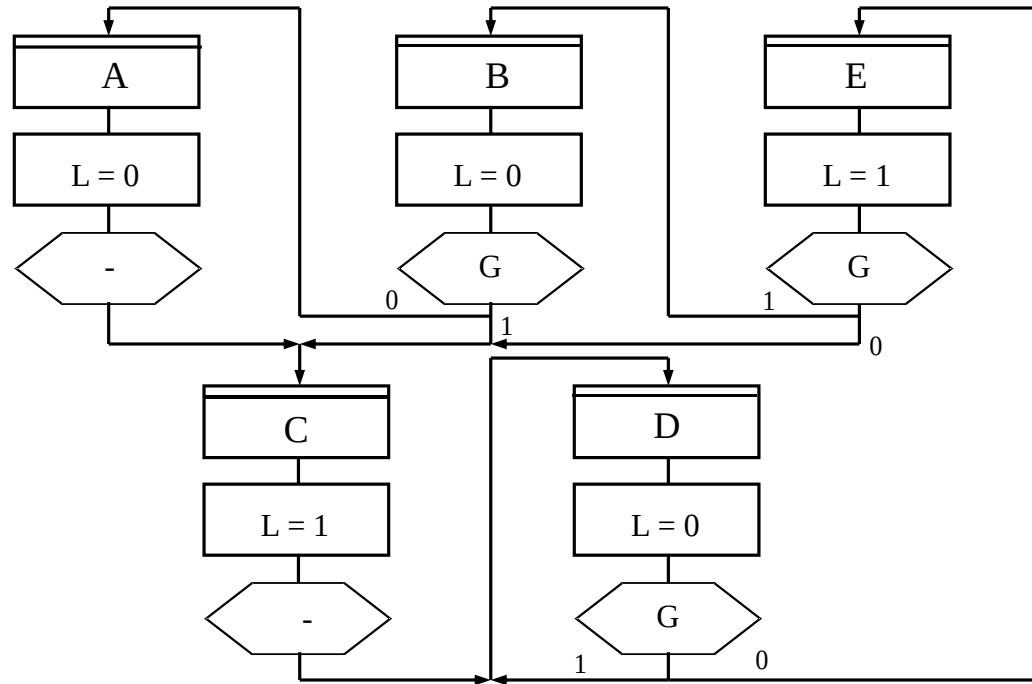


- C ist der zweite Zustand mit einem Nachfolgezustand
 $\Rightarrow C \equiv Z$

Aufgabe F2

S^n	X	S^{n+1}	Y
R	-		0
U	0		
	1	Z	
W	0		
	1	W	
Z	-		
T	0		
	1		

Diagramm	Tabelle
A	R
B	U
C	Z
D	W
E	
G	X
L	Y

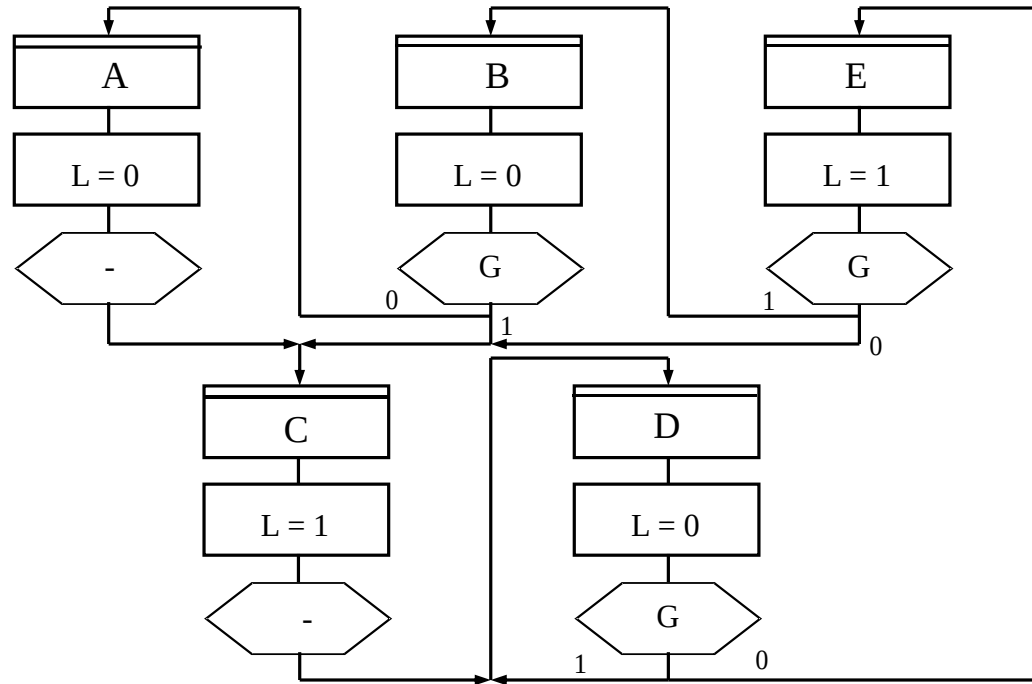


- Mit der Eingabe $G = 1$ erfolgt der Übergang von B nach $C \equiv Z \Rightarrow B \equiv U$

Aufgabe F2

S^n	X	S^{n+1}	Y
R	-		0
U	0		
	1	Z	
W	0		
	1	W	
Z	-		
T	0		
	1		

Diagramm	Tabelle
A	R
B	U
C	Z
D	W
E	T
G	X
L	Y

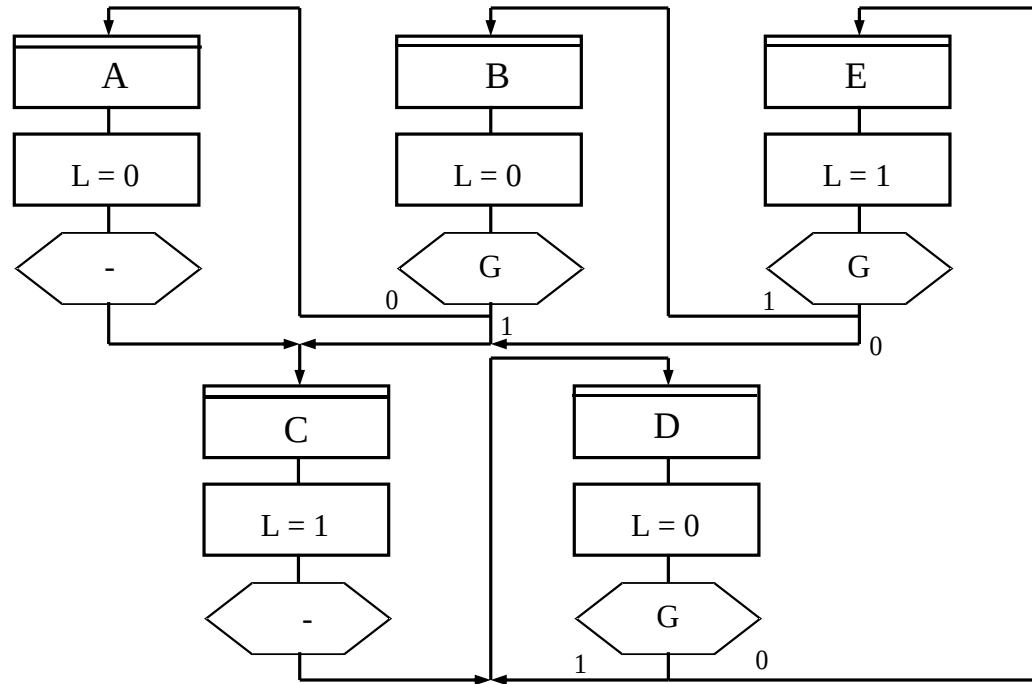


- E und T bleiben über $\Rightarrow E \equiv T$

Aufgabe F2

S^n	X	S^{n+1}	Y
R	-	Z	0
U	0		
	1	Z	
W	0		
	1	W	
Z	-		
T	0		
	1		

Diagramm	Tabelle
A	R
B	U
C	Z
D	W
E	T
G	X
L	Y

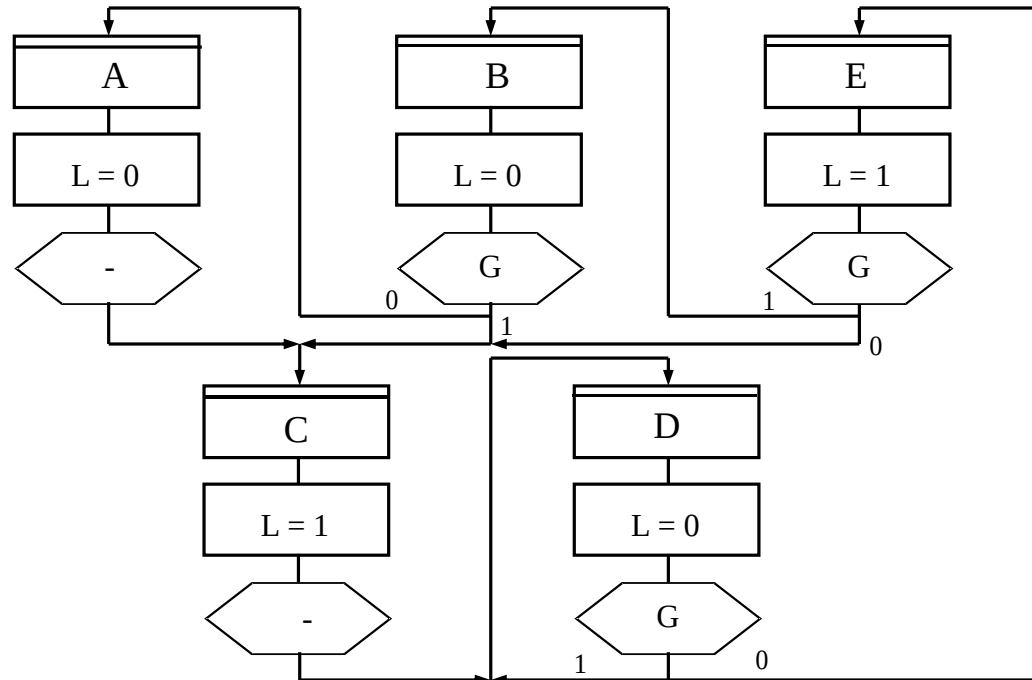


- die Ablaftabelle vervollständigen:
der Übergang von $A \Rightarrow C$ entspricht dem Übergang von $R \Rightarrow Z$

Aufgabe F2

S^n	X	S^{n+1}	Y
R	-	Z	0
U	0	R	
	1	Z	
W	0		
	1	W	
Z	-		
T	0		
	1		

Diagramm	Tabelle
A	R
B	U
C	Z
D	W
E	T
G	X
L	Y

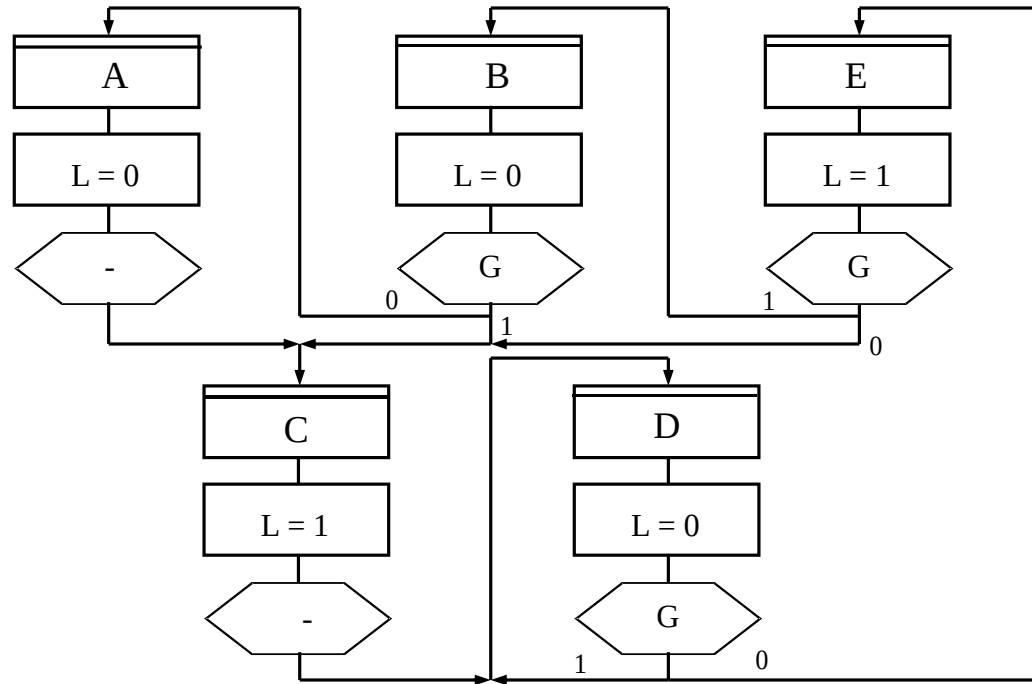


- die Ablaftabelle vervollständigen:
der Übergang von $B \Rightarrow A$ mit $G = 0$ entspricht dem
Übergang von $U \Rightarrow R$

Aufgabe F2

S^n	X	S^{n+1}	Y
R	-	Z	0
U	0 1	R Z	0
W	0 1	W	
Z	-		
T	0 1		

Diagramm	Tabelle
A	R
B	U
C	Z
D	W
E	T
G	X
L	Y

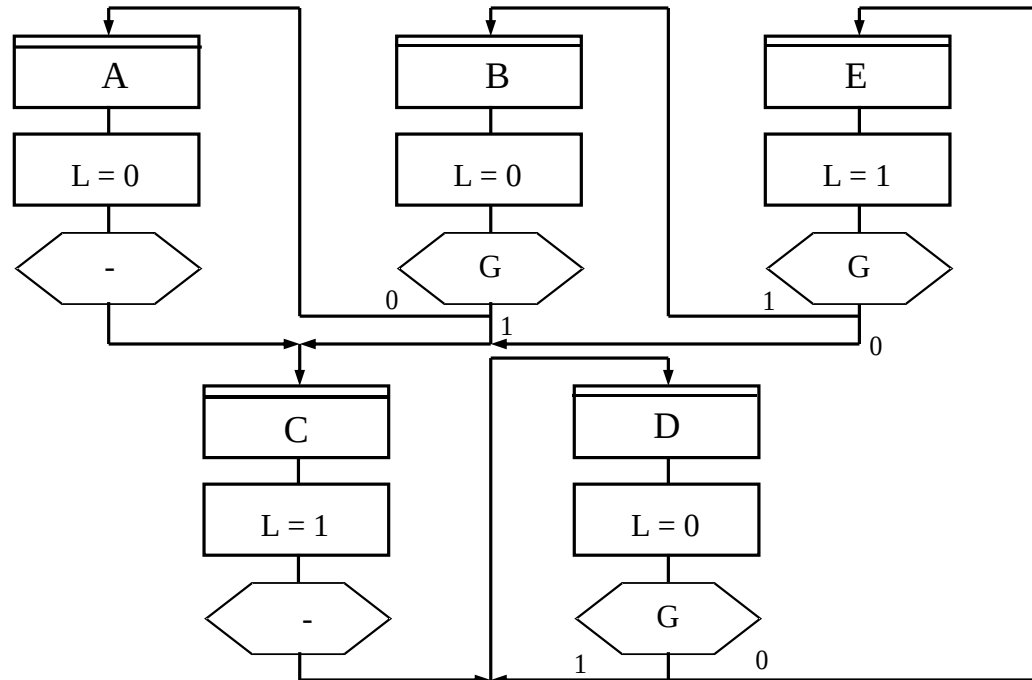


- die Ablauf-tabelle vervollständigen:
der Zustand B erzeugt die Ausgabe L = 0

Aufgabe F2

S^n	X	S^{n+1}	Y
R	-	Z	0
U	0	R	0
	1	Z	
W	0	T	
	1	W	
Z	-		
T	0		
	1		

Diagramm	Tabelle
A	R
B	U
C	Z
D	W
E	T
G	X
L	Y

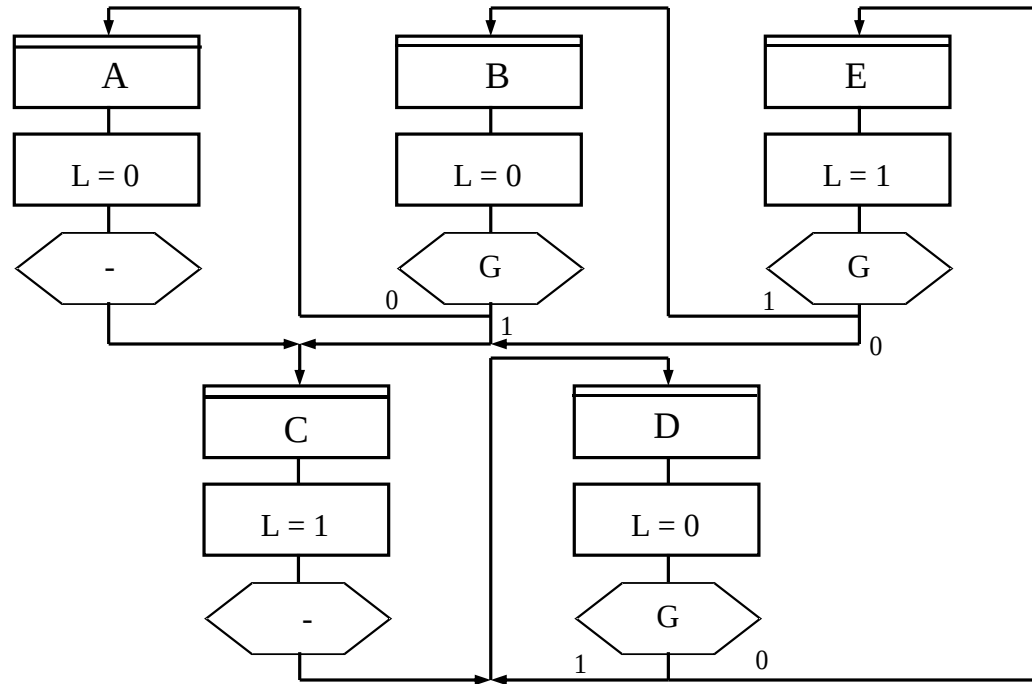


- die Ablaftabelle vervollständigen:
der Übergang von $D \Rightarrow E$ entspricht dem
Übergang von $W \Rightarrow T$

Aufgabe F2

S^n	X	S^{n+1}	Y
R	-	Z	0
U	0	R	0
	1	Z	
W	0	T	0
	1	W	
Z	-		
T	0		
	1		

Diagramm	Tabelle
A	R
B	U
C	Z
D	W
E	T
G	X
L	Y

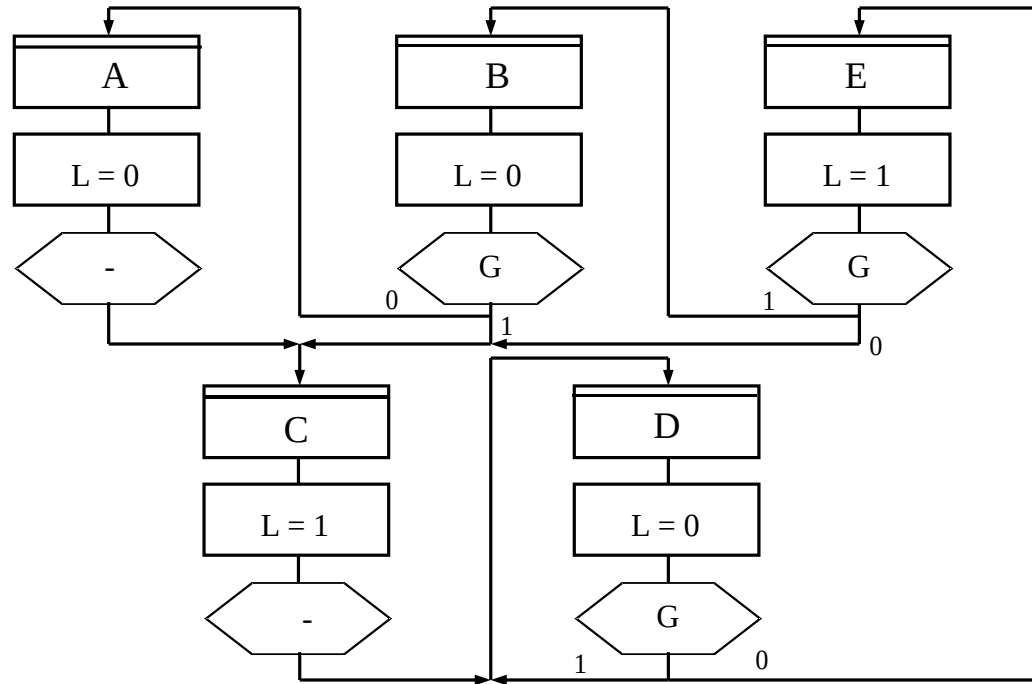


- die Ablaftabelle vervollständigen:
der Zustand D erzeugt die Ausgabe $L = 0$

Aufgabe F2

S^n	X	S^{n+1}	Y
R	-	Z	0
U	0	R	0
	1	Z	
W	0	T	0
	1	W	
Z	-	W	
T	0		
	1		

Diagramm	Tabelle
A	R
B	U
C	Z
D	W
E	T
G	X
L	Y

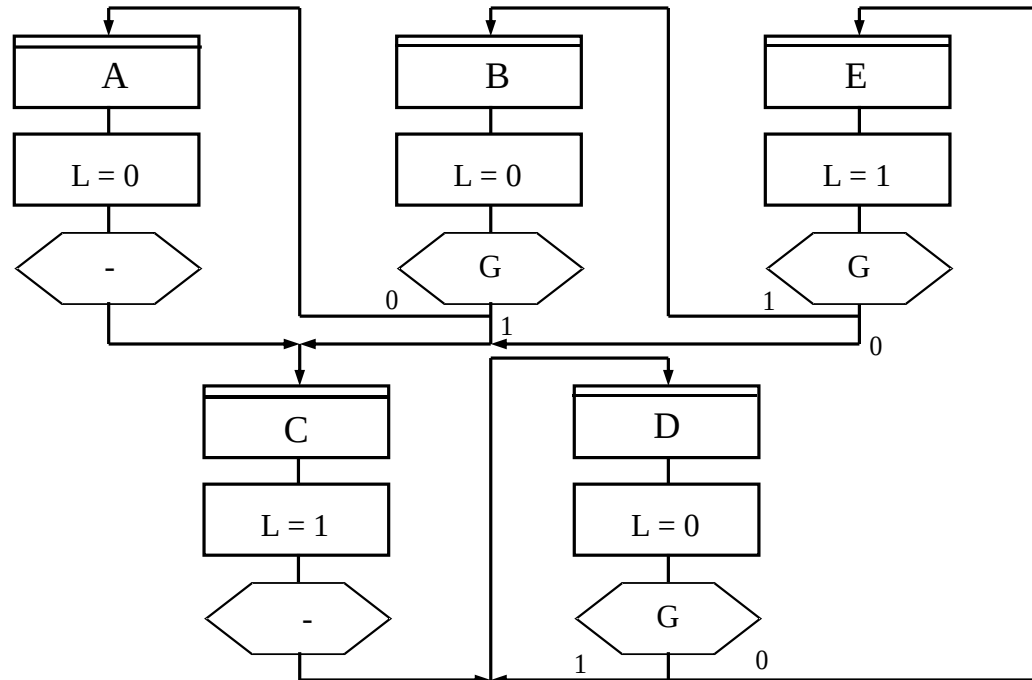


- die Ablaftabelle vervollständigen:
der Übergang von $C \Rightarrow D$ entspricht dem
Übergang von $Z \Rightarrow W$

Aufgabe F2

S^n	X	S^{n+1}	Y
R	-	Z	0
U	0	R	0
	1	Z	
W	0	T	0
	1	W	
Z	-	W	1
T	0		
	1		

Diagramm	Tabelle
A	R
B	U
C	Z
D	W
E	T
G	X
L	Y

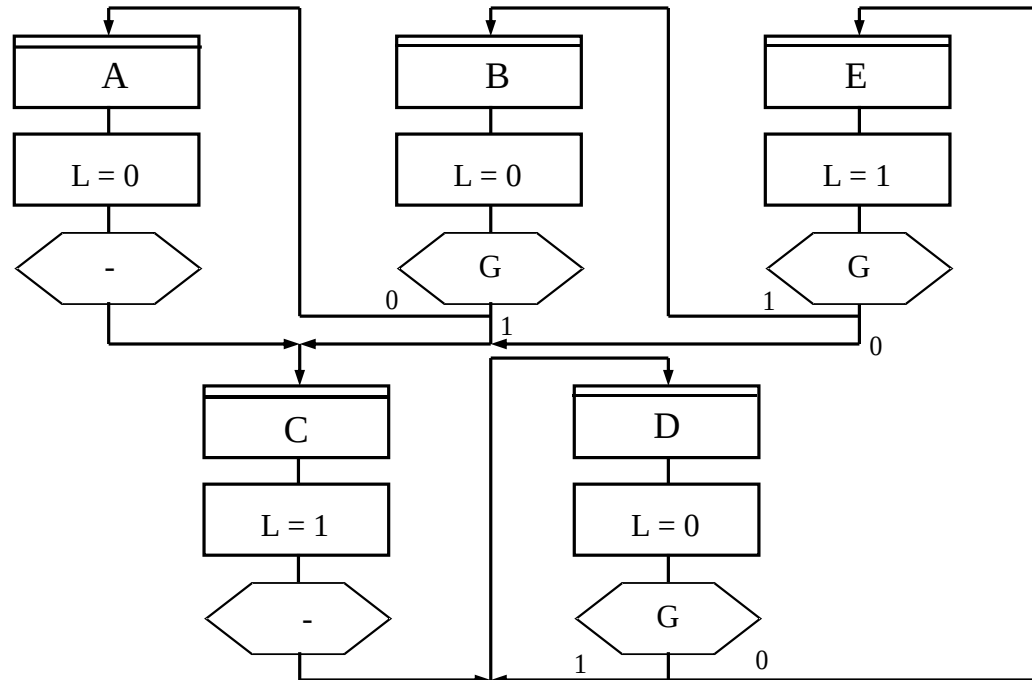


- die Ablauftabelle vervollständigen:
der Zustand C erzeugt die Ausgabe L = 1

Aufgabe F2

S^n	X	S^{n+1}	Y
R	-	Z	0
U	0	R	0
	1	Z	
W	0	T	0
	1	W	
Z	-	W	1
T	0	Z	
	1		

Diagramm	Tabelle
A	R
B	U
C	Z
D	W
E	T
G	X
L	Y

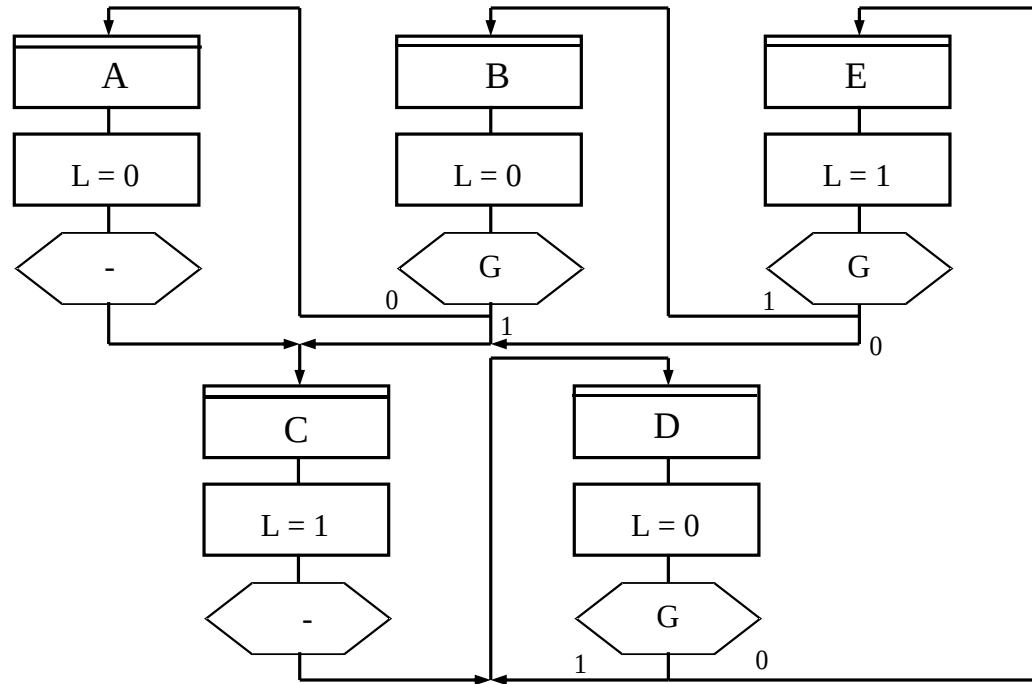


- die Ablaftabelle vervollständigen:
der Übergang von $E \Rightarrow C$ mit $G = 0$ entspricht dem Übergang von $T \Rightarrow Z$

Aufgabe F2

S^n	X	S^{n+1}	Y
R	-	Z	0
U	0	R	0
	1	Z	
W	0	T	0
	1	W	
Z	-	W	1
T	0	Z	
	1	U	

Diagramm	Tabelle
A	R
B	U
C	Z
D	W
E	T
G	X
L	Y

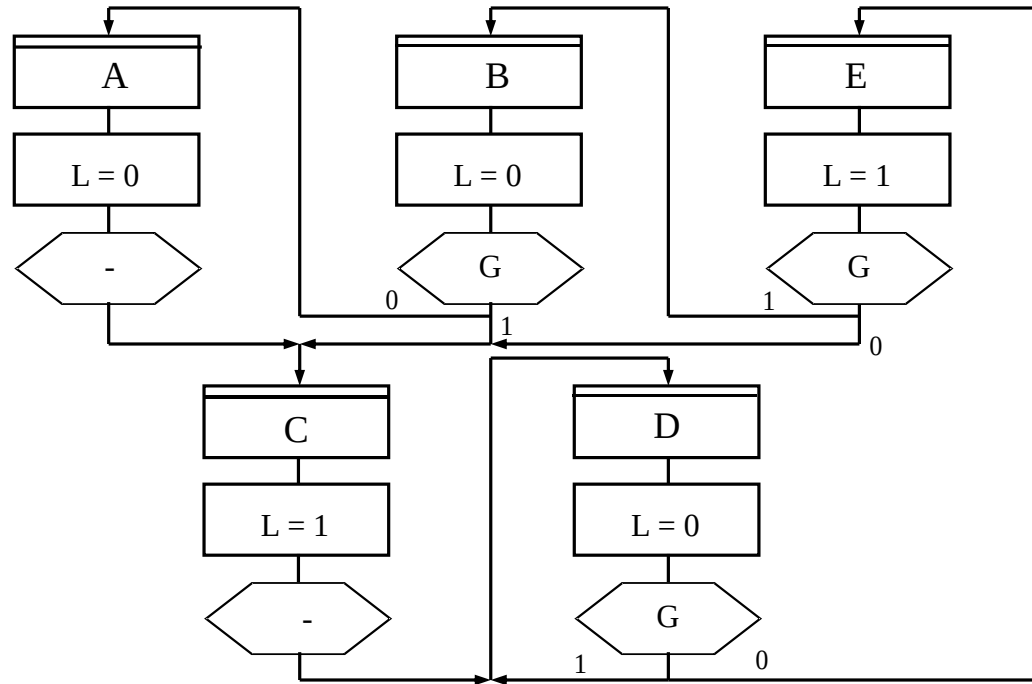


- die Ablaftabelle vervollständigen:
der Übergang von $E \Rightarrow B$ mit $G = 1$ entspricht dem Übergang von $T \Rightarrow U$

Aufgabe F2

S^n	X	S^{n+1}	Y
R	-	Z	0
U	0	R	0
	1	Z	
W	0	T	0
	1	W	
Z	-	W	1
T	0	Z	1
	1	U	

Diagramm	Tabelle
A	R
B	U
C	Z
D	W
E	T
G	X
L	Y



- die Ablaftabelle vervollständigen:
der Zustand E erzeugt die Ausgabe L = 1