

# Klausur zur Vorlesung

## "Grundlagen der technischen Informatik" und "Grundlagen der Rechnerarchitektur" Sommersemester 2001

### 1. Teil: GTI

Der erste Teil (GTI) der Klausur umfasst 6 Aufgaben und hat 9 Seiten, die in maximal 1 Stunde zu bearbeiten sind. **Es sind keine Hilfsmittel zugelassen.** Schreiben Sie Ihre Lösungen nur auf die dafür vorgesehenen Blätter! Verwenden Sie **kein eigenes Konzeptpapier**; notfalls erhalten Sie welches bei der Aufsicht.

**Lassen Sie die Aufgaben und ihre Lösungen zusammengeheftet!**

Schreiben Sie auf jedes Blatt (auch auf das Konzeptpapier) in Blockschrift Ihren **Namen** und Ihre **Matrikelnummer**.

**Bei mehreren präsentierten Lösungen wird die Aufgabe nicht gewertet!** Streichen Sie daher bei Angabe mehrerer Lösungsansätze die nicht zu bewertenden Lösungen durch!

Abschreiben und abschreiben lassen oder Hilfe Dritter führt zum Nichtbestehen der Klausur.

Es sind in diesem ersten Teil insgesamt 60 Punkte erreichbar.

**Viel Erfolg!**

## Musterlösung

Vor- und Nachname: \_\_\_\_\_

*(ohne Gewähr)*

Matrikelnummer: \_\_\_\_\_ Fachbereich: \_\_\_\_\_

Studienfach (ankreuzen):

Informatik	<input type="checkbox"/>
Ingenieur-Informatik (Elektrotechnik)	<input type="checkbox"/>
Ingenieur-Informatik (Maschinenbau)	<input type="checkbox"/>
Ingenieur-Informatik (Informatik)	<input type="checkbox"/>
Wirtschafts-Informatik	<input type="checkbox"/>
Informatik als Nebenfach	<input type="checkbox"/>
Lehramt Informatik	<input type="checkbox"/>

Aufgabe	1	2	3	4	5	6	Summe
Punkte	10	8	11	9	10	12	60
Erreicht							

Summe	GTI:		GRA:		Gesamt:		Note:	
-------	------	--	------	--	---------	--	-------	--

**Aufgabe 1 (10min, 10 Punkte): Quine/McClusky**

Minimieren Sie die boolesche Funktion

$$F = a b \bar{c} d + \bar{a} \bar{b} \bar{c} d + \bar{a} b \bar{c} \bar{d} + a \bar{b} \bar{c} \bar{d} + \bar{a} \bar{b} \bar{c} \bar{d} + a b \bar{c} \bar{d} + \bar{b} c \bar{d} + \bar{a} b \bar{c} d$$

mit Hilfe des **aus der GTI-Vorlesung bekannten** Quine/McClusky Verfahrens. Das Verfahren benötigt hier 2 Iterationen. Geben Sie die Zwischenergebnisse nach den angegebenen Aktionen des Verfahrens in beiden Iterationen an. Bei der Summe der Konsensusse sind **alle** (d.h. **auch eventuell doppelt auftretende**) anzugeben!

Nr.	Aktion	Zwischenergebnisse
	Ausgangsfunktion F (in MinTerm-NF)	$a b \bar{c} d + \bar{a} \bar{b} \bar{c} d + \bar{a} b \bar{c} \bar{d} + a \bar{b} \bar{c} \bar{d} + \bar{a} \bar{b} \bar{c} \bar{d} + a b \bar{c} \bar{d} + \bar{b} c \bar{d} + \bar{a} b \bar{c} d$
1	Summe <b>aller</b> „Simplen Konsensusse“	$b \bar{c} d + a b \bar{c} + \bar{a} \bar{b} \bar{c} + \bar{a} \bar{c} \bar{d} + \bar{a} b \bar{c} + b \bar{c} \bar{d} + \bar{b} \bar{c} \bar{d} + a \bar{b} \bar{d} + \bar{b} c \bar{d}$
	Nach „Streichen von Verlängerungen“ und „Vereinfachen“	$b \bar{c} d + a b \bar{c} + \bar{a} \bar{b} \bar{c} + \bar{a} \bar{c} \bar{d} + \bar{a} b \bar{c} + b \bar{c} \bar{d} + \bar{b} \bar{c} \bar{d} + a \bar{b} \bar{d} + \bar{b} c \bar{d}$
2	Summe <b>aller</b> „Simplen Konsensusse“	$b \bar{c} + b \bar{c} + \bar{a} \bar{c} + \bar{b} \bar{d}$
	Nach „Streichen von Verlängerungen“ und „Vereinfachen“	$b \bar{c} + \bar{a} \bar{c} + \bar{b} \bar{d}$

**Ersatztable:**

Wenn Sie diese Tabelle anstelle der oberen bewertet haben wollen, so streichen Sie die obere Tabelle durch!

Nr.	Aktion	Zwischenergebnisse
	Ausgangsfunktion F (in MinTerm-NF)	
1	Summe <b>aller</b> „Simplen Konsensusse“	
	Nach „Streichen von Verlängerungen“ und „Vereinfachen“	
2	Summe <b>aller</b> „Simplen Konsensusse“	
	Nach „Streichen von Verlängerungen“ und „Vereinfachen“	

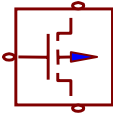
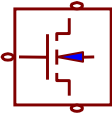
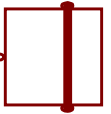
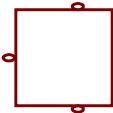
**Aufgabe 2 (8 min, 8 Punkte): CMOS-Technologie**

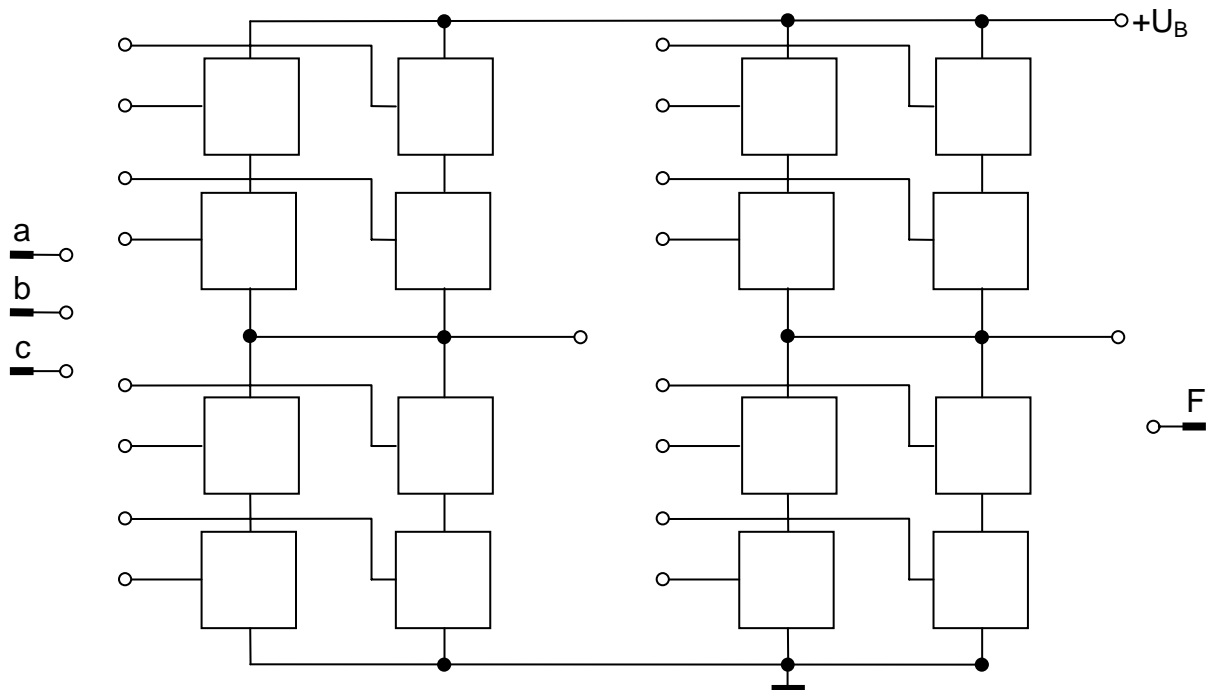
Gegeben sei folgende boolesche Funktion:

$$F = a + \overline{(b \cdot c)}$$

- a) Vervollständigen Sie die Implementierung im unteren Bild mit Hilfe der angegebenen Bauelemente zu einer reinen CMOS Schaltung, wobei nur NAND und NOR implementiert werden können. Es stehen dazu nur zwei Schaltungsgruppen zur Verfügung, mit denen ein NAND und ein NOR oder zwei NAND oder zwei NOR implementiert werden können. Tragen Sie dazu in die freien Kästen **P**, **N**, **L** bzw. **E** für das jeweils einzusetzende Schaltzeichen ein und fügen Sie **fehlende Verbindungen** ein, so dass sich genau die Funktion F ergibt.

*Einzutragende Buchstaben:*

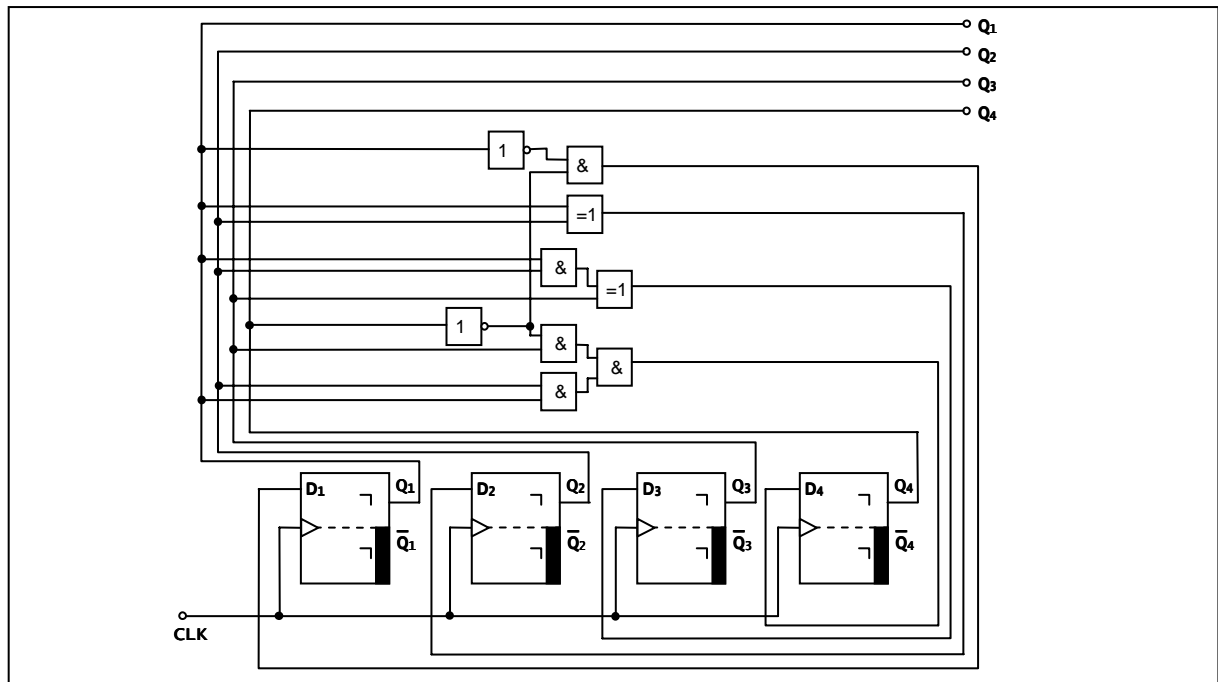
<b>P</b>	PMOS Transistor (Anreicherungstyp, ist durchgeschaltet falls Gate Low )	<b>N</b>	NMOS Transistor (Anreicherungstyp, ist durchgeschaltet falls Gate High )	<b>L</b>	Überbrückung
					
<b>E</b>	leer, unbenutzt				
					



*Hinweis: nicht alle Konnektoren müssen verbunden werden!*

**Aufgabe 3 (11 min, 4 + 5 + 2 Punkte): Flip-Flops und Register**

Gegeben ist der folgende Moore-Automat:

*Hinweis: Beachten Sie, dass es sich um Master-Slave D-Flip-Flops handelt!*

- a) Geben Sie die booleschen Funktionen in Abhängigkeit von  $Q_1$  bis  $Q_4$  für die Übergangsfunktion des Automaten an:

$$D_1 = \bar{Q}_4 \cdot \bar{Q}_1$$

$$D_2 = Q_1 \text{ XOR } Q_2$$

$$D_3 = (Q_1 \cdot Q_2) \text{ XOR } Q_3$$

$$D_4 = Q_1 \cdot Q_2 \cdot Q_3 \cdot \bar{Q}_4$$

- b) Füllen Sie die Tabelle aus:

Nach Takt	$Q_4$	$Q_3$	$Q_2$	$Q_1$
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	0	0	0	0
10	0	0	0	1

- c) Um was für eine Schaltung handelt es sich (bzw. wie kann man die Ausgabe interpretieren)?

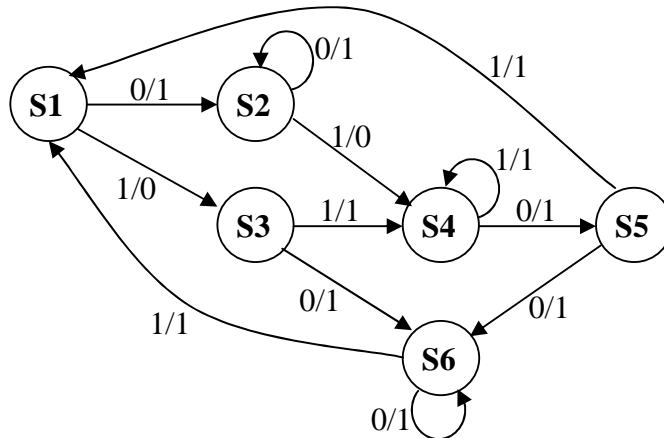
synchroner  
Binärzähler bis 8

*Nach Takt  $i$  = nach  $i$ -ter negativer Flanke des Taktsignals*



**Aufgabe 5 (10 min, 8 + 2 Punkte): Automaten**

Wandeln Sie den gegebenen Mealy-Automaten in einen äquivalenten Zustands-minimalen (reduzierten) Mealy-Automaten um.



S1, S2, S3, S4, S5, S6 sind Zustände;  
1/0 steht für Eingabe X = 1, Ausgabe: Y = 0.

a) Ergänzen Sie bitte das vorgegebene Schema nach dem Ginsburg/Huffmann-Verfahren:

Äquivalenzklassenzuordnung im

	1. Schritt		2. Schritt		3. Schritt		4. Schritt	
$\delta/\lambda$	0	1	0	1	0	1	0	1
S1	S2/1	S3/0	A <sub>1</sub>	A <sub>2</sub>	B <sub>1</sub>	B <sub>2</sub>		
S2	S2/1	S4/0	A <sub>1</sub>	A <sub>2</sub>	B <sub>1</sub>	B <sub>2</sub>		
S3	S6/1	S4/1	A <sub>2</sub>	A <sub>2</sub>	B <sub>3</sub>	B <sub>2</sub>		
S4	S6/1	S4/1	A <sub>2</sub>	A <sub>2</sub>	B <sub>3</sub>	B <sub>2</sub>		
S5	S6/1	S1/1	A <sub>2</sub>	A <sub>1</sub>	B <sub>3</sub>	B <sub>1</sub>		
S6	S6/1	S1/1	A <sub>2</sub>	A <sub>1</sub>	B <sub>3</sub>	B <sub>1</sub>		

Geben Sie hier an, welche Zustände in den Äquivalenzklassen enthalten sind:

$$\begin{aligned}
 A_1 &= \{ \text{S1, S2} \}, & B_1 &= \{ \text{S1, S2} \}, \\
 A_2 &= \{ \text{S3, S4, S5, S6} \}, & B_2 &= \{ \text{S3, S4} \}, \\
 C_1 &= \{ \}, & B_3 &= \{ \text{S5, S6} \}, \\
 C_2 &= \{ \}, & C_3 &= \{ \}, \\
 C_4 &= \{ \} & &
 \end{aligned}$$

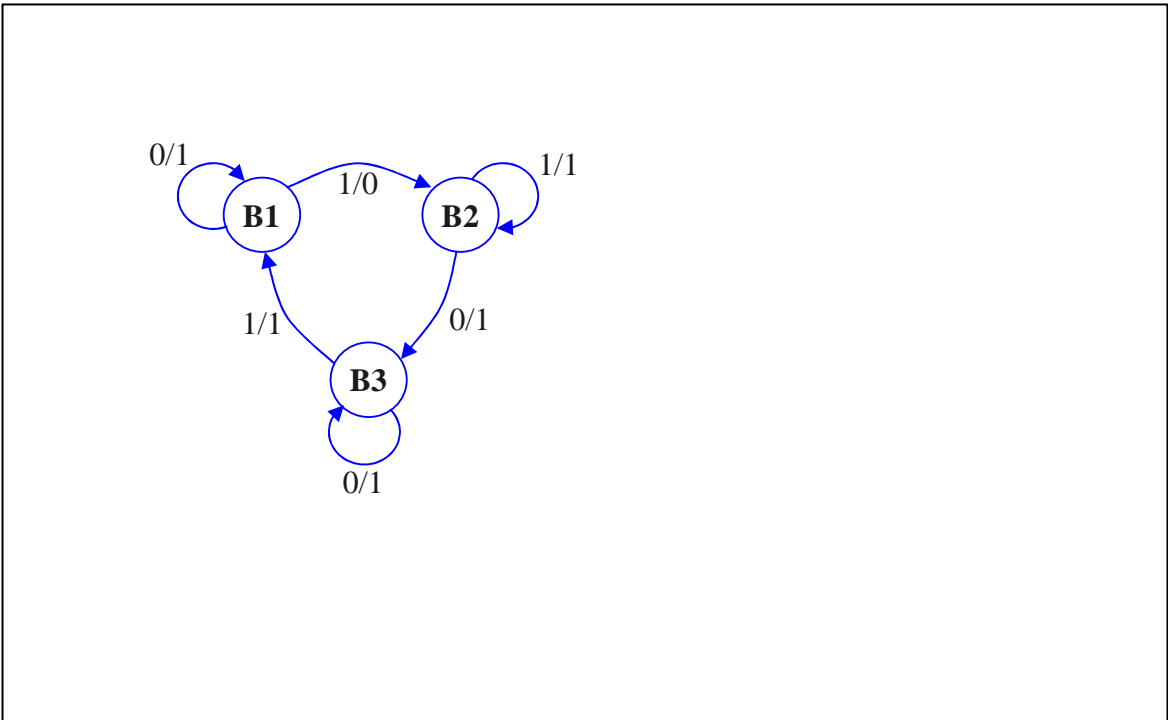
*Hinweis: nicht benötigte Felder(Schritte) und Äquivalenzklassen lassen Sie leer.*

Name:

Matrikel-Nummer:

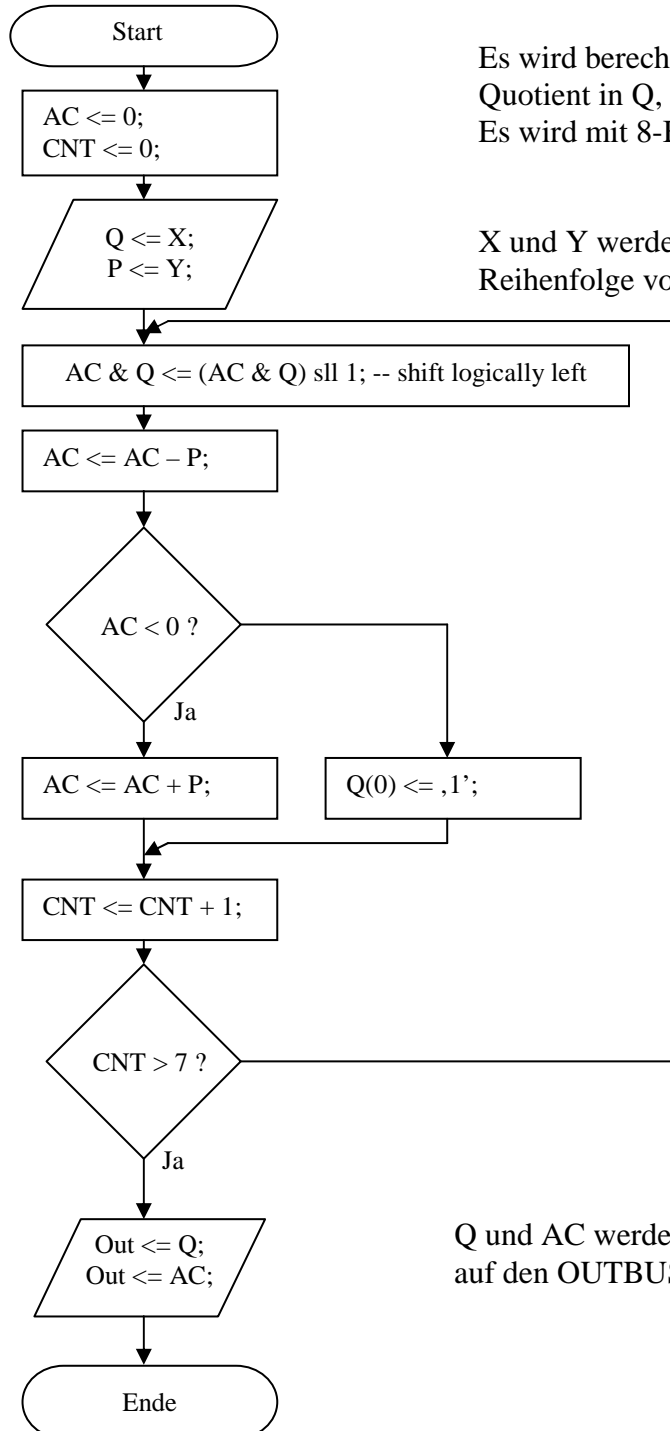
Fortsetzung Aufgabe 5: Automaten

b) Geben Sie den Zustandsübergangsgraphen des minimierten Mealy-Automaten an:



**Aufgabe 6 (12 min, 12 Punkte): Steuerwerk**

Gegeben sind der folgende Algorithmus und ein dafür vorgesehenes Rechenwerk mit seinen Datenpfaden, Signal- und Steuerleitungen.



Es wird berechnet  $Q = X / Y$ ;  
Quotient in Q, Rest in AC;  
Es wird mit 8-Bit Zahlen gerechnet.

X und Y werden am Anfang in dieser  
Reihenfolge vom INBUS gelesen.

Folgende Ausgangssignale stehen zur  
Verfügung:

LdAC: lädt den AC vom INBUS  
LdP: lädt P vom INBUS  
LdQ: lädt Q vom INBUS  
StAC: speichert AC auf den OUTBUS  
StQ: speichert Q auf den OUTBUS  
Reset: setzt AC, Q und CNT auf 0  
SLL: shiftet AC & Q ein Bit nach  
links  
Set: setzt Q(0) auf ,1'  
Inc: inkrementiert CNT um 1  
Add:  $AC \leq AC + P$   
Sub:  $AC \leq AC - P$   
Comp0: setzt das C-Flag, wenn  $AC < 0$ ,  
sonst wird es gelöscht

Folgende Eingabesignale stehen zur  
Verfügung:

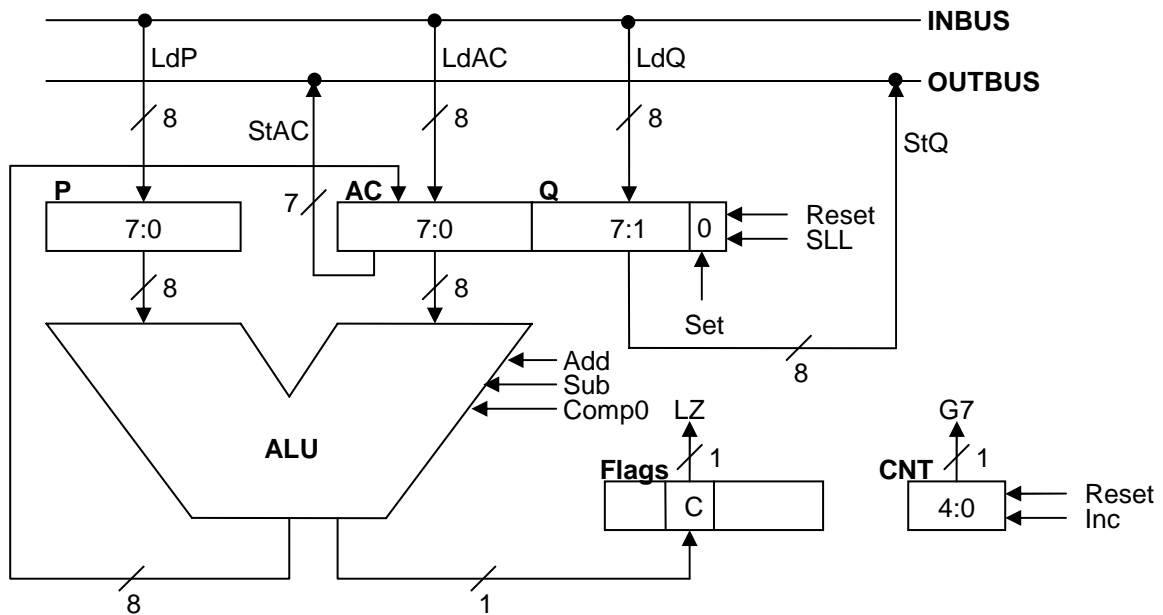
LZ: repräsentiert das C-Flag  
G7: ist ,1' wenn  $CNT > 7$

Q und AC werden in dieser Reihenfolge  
auf den OUTBUS geschrieben.



**Fortsetzung Aufgabe 6: Steuerwerk**

Folgendes Rechenwerk ist gegeben:



Geben Sie den **kleinsten** Moore-Automaten des Steuerwerks an, der den vorgenannten Algorithmus implementiert! Beschriften Sie jede Kante mit einer booleschen Formel und geben Sie tabellarisch an, in welchem Zustand welche Signale aktiviert werden!

Zustandsüberganggraph:

Zustand	aktivierte Signale
S0	Reset
S1	LdQ
S2	LdP
S3	SLL
S4	Sub
S5	Comp0
S6	Add, Inc
S7	Set, Inc
S8	StQ
S9	StAC