

**Modul Computersysteme
Prüfungsklausur SS 2016**

Lösungsvorschläge

**Prof. Dr. J. Keller
LG Parallelität und VLSI
Prof. Dr.-Ing. W. Schiffmann
LG Rechnerarchitektur**

Aufgabe 1 Schaltfunktionen (11 Punkte):

a) Gegeben ist das folgende Karnaugh-Diagramm einer Schaltfunktion f in den vier Variablen X_1 bis X_4 .

	X_1				
X_2	0	0	0	0	
	1	0	0	1	X_4
	1	0	1	1	
	1	0	1	0	
	X_3				

Vervollständigen Sie die Wertetabelle! (4 P.)

Hinweis: Wie im Kurstext verwenden wir die verkürzende Schreibweise für Karnaugh-Diagramme, bei der gerade die Spalten bzw. Zeilen mit einer Variablen markiert werden, bei denen die entsprechende Variable den Wert 1 hat. Zum Beispiel gilt in den obersten beiden Zeilen $X_2 = 1$, und in den beiden äußeren Spalten $X_3 = 0$.

X_1	X_2	X_3	X_4	$f(X_1, \dots, X_4)$
0	0	0	0	
0	0	0	1	
0	0	1	0	
0	0	1	1	
0	1	0	0	
0	1	0	1	
0	1	1	0	
0	1	1	1	
1	0	0	0	
1	0	0	1	
1	0	1	0	
1	0	1	1	
1	1	0	0	
1	1	0	1	
1	1	1	0	
1	1	1	1	

b) Geben Sie die Anzahl der Primimplikanten der Funktion f aus Teil a) an. (4 P.)

c) Nennen Sie einen der Primimplikanten der Funktion f aus Teil a). (1 P.)

d) Kann es eine Schaltfunktion in vier Variablen geben, deren Formelgröße höchstens 3 beträgt?
Begründen Sie Ihre Antwort (2 P.)

Hinweis: Falls Ihre Antwort „JA“ lautet, reicht als Begründung ein Beispiel! Falls Ihre Antwort „NEIN“ lautet, reicht als Begründung ein Gegenbeispiel nicht aus!

Lösung: a) Die Wertetabelle lautet:

X_1	X_2	X_3	X_4	$f(X_1, \dots, X_4)$
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	0
1	0	0	0	1
1	0	0	1	1
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	1
1	1	1	0	0
1	1	1	1	0

b) Die Anzahl der Primimplikanten ist 5, s. Teilaufgabe c).

c) Die Primimplikanten lauten $\bar{X}_1 \bar{X}_2 X_3$, $\bar{X}_1 \bar{X}_2 X_4$, $X_1 \bar{X}_2 \bar{X}_3$, $\bar{X}_3 X_4$.

d) Ja! Als Beleg dient die Beispielfunktion $X_1 \wedge X_2 \vee X_3 \wedge X_4$.

Aufgabe 2 Binärzahlen, Arithmetik (6 Punkte):

Rechnen Sie die Zahlendarstellungen um:

$$[10001001]_2 = \boxed{}$$

$$[01001100]_2 = \boxed{}$$

$$\langle 1001101 \rangle_2 = \boxed{}$$

$$\text{bin}_6(19) = \boxed{}$$

$$\text{twoc}(-1) = \text{mit 8 Stellen einschl. Vorzeichen} \boxed{}$$

$$\text{twoc}(-10) = \text{mit 5 Stellen einschl. Vorzeichen} \boxed{}$$

Lösung:

$$[10001001]_2 = -128 + 8 + 1 = -119$$

$$[01001100]_2 = 64 + 8 + 4 = 76$$

$$\langle 1001101 \rangle_2 = 64 + 8 + 4 + 1 = 77$$

$$\text{bin}_6(19) = \text{bin}_6(16 + 2 + 1) = 010011$$

$$\text{twoc}(-1) = 11111111 \text{ mit 8 Stellen einschl. Vorz.}$$

$$\text{twoc}(-10) = \text{twoc}(-16 + 4 + 2) = 10110 \text{ mit 5 Stellen einschl. Vorz.}$$

Aufgabe 3 Schaltnetze (8 Punkte + 2 Zusatzpunkte):

a) Erstellen Sie die Wertetabelle für einen Halbaddierer. (2 P.)

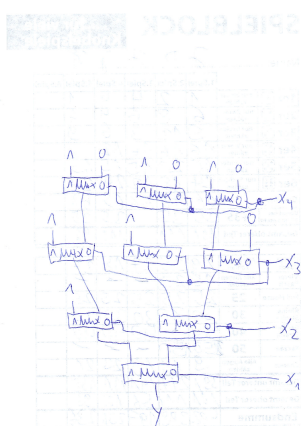
<i>a</i>	<i>b</i>	<i>c</i>	<i>s</i>
0	0		
0	1		
1	0		
1	1		

Lösung:*Die Wertetabelle lautet*

<i>a</i>	<i>b</i>	<i>c</i>	<i>s</i>
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

b) Erstellen Sie die Wertetabelle der Schaltfunktion, die von folgendem Schaltnetz berechnet wird. (6 P.)

Hinweis: Überlegen Sie zunächst, was im Schaltnetz passiert, wenn $X_1 = 1$ und $X_2 = 1$ (Unterste vier Zeilen der Wertetabelle). Überlegen Sie danach, was im Schaltnetz passiert, wenn $X_1 = 1$ und $X_2 = 0$, usw.



X_1	X_2	X_3	X_4	Y
0	0	0	0	
0	0	0	1	
0	0	1	0	
0	0	1	1	
0	1	0	0	
0	1	0	1	
0	1	1	0	
0	1	1	1	
1	0	0	0	
1	0	0	1	
1	0	1	0	
1	0	1	1	
1	1	0	0	
1	1	0	1	
1	1	1	0	
1	1	1	1	

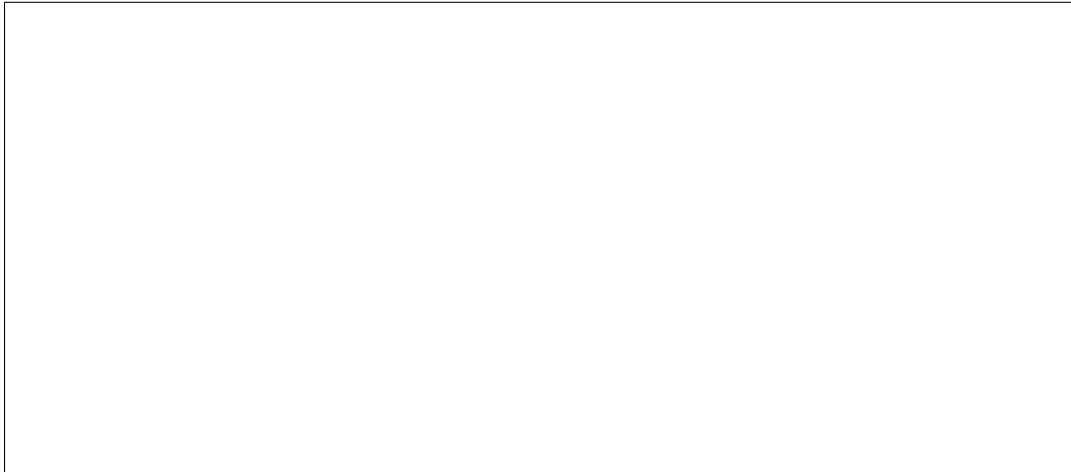
Lösung:

X_1	X_2	X_3	X_4	Y
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

Diese Schaltfunktion nimmt genau dann den Wert 1 an, wenn mindestens zwei der Variablen den Wert 1 haben.

c) Möglichkeit für 2 Zusatzpunkte:

Beschreiben Sie, wie in dem in b) gezeigten Schaltnetz die oberste Reihe von Multiplexern vereinfacht werden könnte.

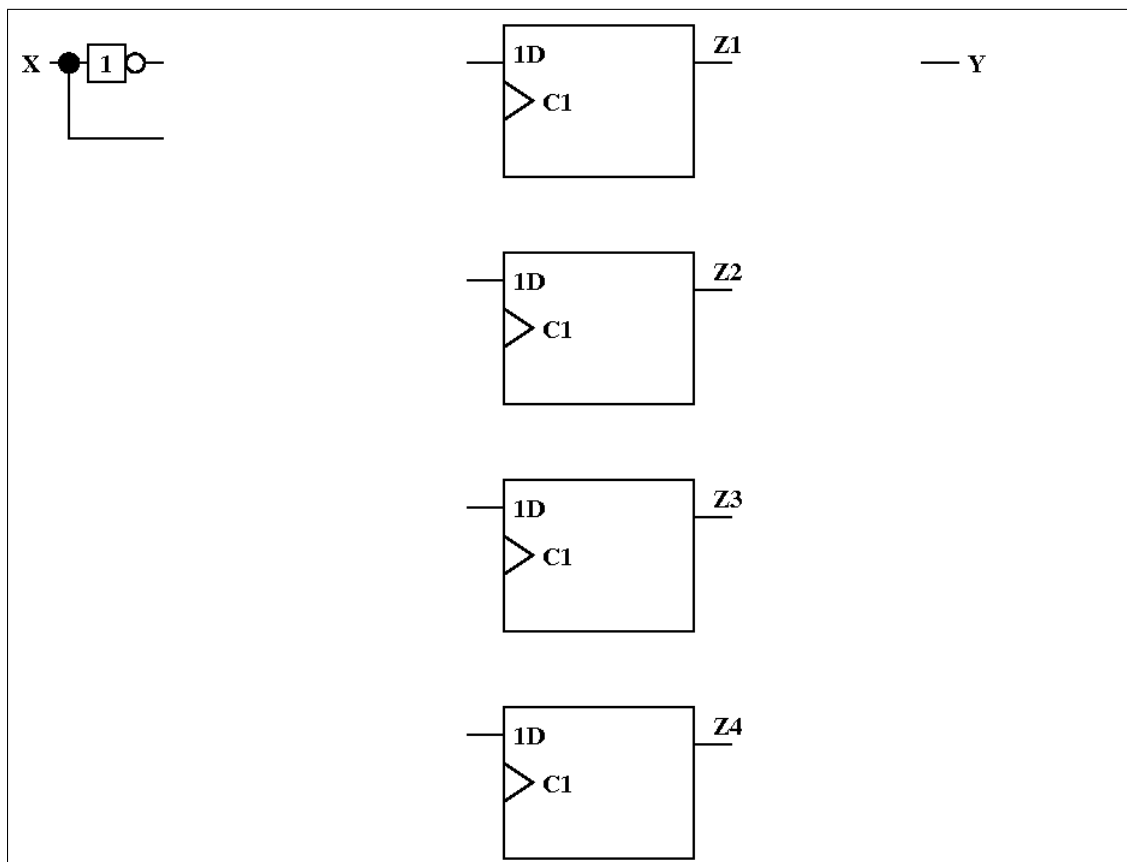
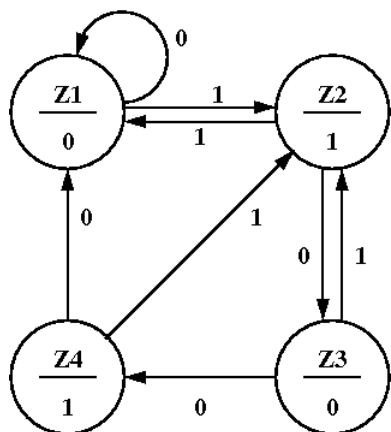


Lösung:

Man kann die drei Multiplexer der obersten Reihe einfach weglassen, und ihre Ausgänge mit X_4 verbinden, denn sie berechnen die Schaltfunktion $1 \wedge X_4 \vee 0 \wedge \bar{X}_4 = X_4$.

Aufgabe 4 Schaltwerkssynthese (9 Punkte):

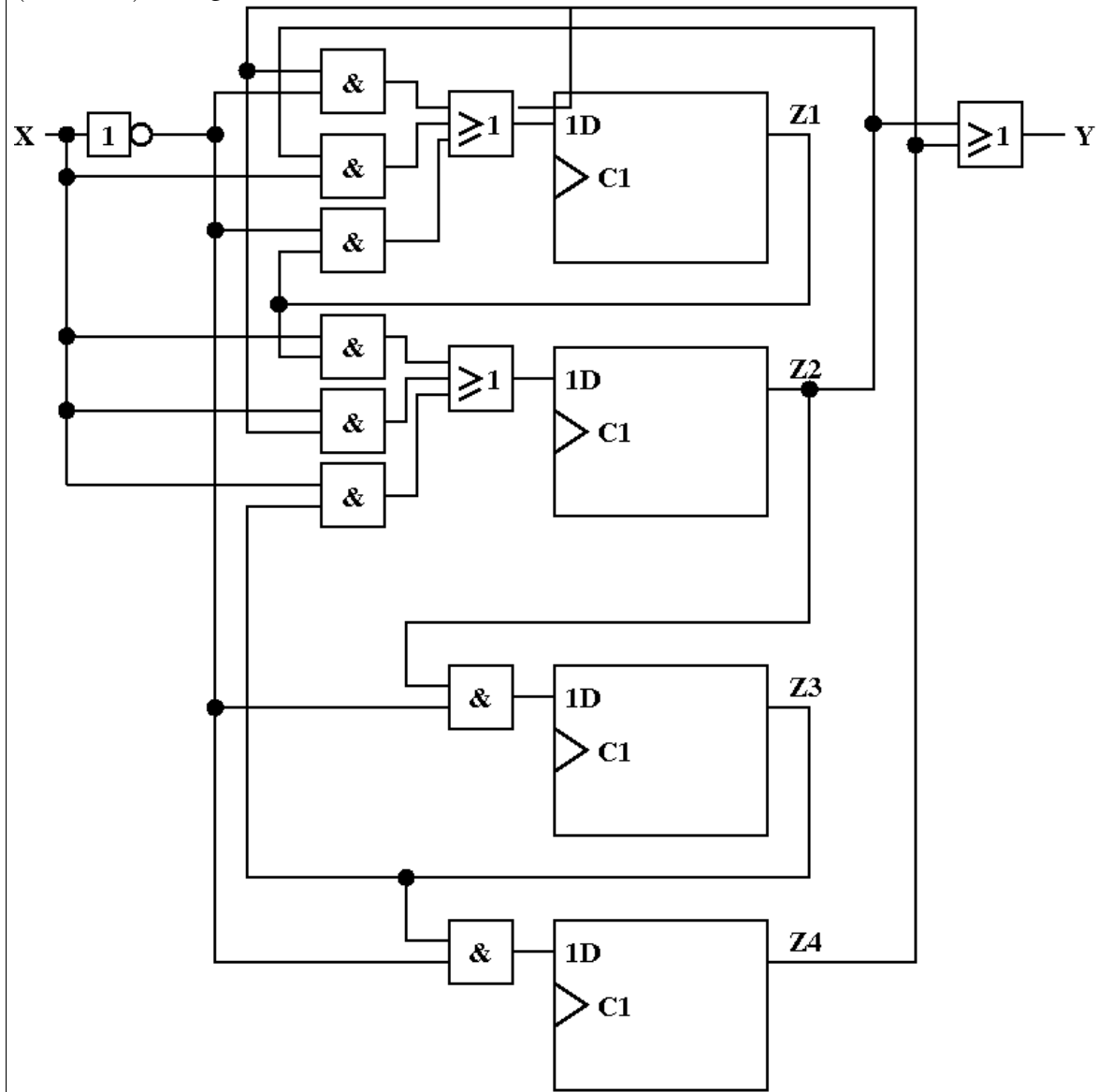
a) Gegeben sei ein Automat mit 4 Zuständen, Eingangssignal X und Ausgangssignal Y durch das folgende Zustandsdiagramm. Erstellen Sie ein Schaltwerk für diesen Automaten mit Hot-One-Codierung der Zustände unter Nutzung von D-Flipflops. (5 P.)



HINWEIS: dieses Bild muss durch das Bild aus der Prüfungsklausur WS2015/16 Aufgabe 4a unten ersetzt werden (das Bild müsste Patrick oder Jörg haben)! Dabei müssen die Zustandsnummern Z1 bis Z4 lauten!

Lösung:

Das vollständige Schaltwerk benötigt neben den 4 Flipflops 12 Gatter, wobei keine Optimierung vorgenommen wurde. Alternativ könnte nämlich auch $Z2^+ = X$ und $Z1^+ = (Z1 \vee Z4) \wedge \bar{X}$ gewählt werden.



b) Sie implementieren einen Automaten mit 8 Zuständen durch ein Schaltwerk mit Hot-One-Codierung. Die Gatter für die Zustandsübergangsfunktion und die Ausgabefunktion haben Sie bereits realisiert. Allerdings stellen Sie fest, dass Sie nur noch 3 Flipflops besitzen. Zeichnen Sie auf, wie Sie mit diesen Flipflops, einem 3-Bit Encoder und einem 3-Bit Decoder die 8 Flipflops der Hot-One-Codierung „simulieren“ können. Begründen Sie Ihre Schaltung. (4 P.)

Lösung: Man verbindet die Eingänge der 3 Flipflops mit den Ausgängen des Encoders, und die Ausgänge der 3 Flipflops mit den Eingängen des Decoders. Diese Schaltung hat 8 Eingänge und 8 Ausgänge so wie die 8 Flipflops. Da es sich um eine Hot-One-Codierung handelt, ist stets genau ein Eingang mit einer 1 belegt, so dass der Encoder einen gültigen Zustand liefert, der in 3 Flipflops gespeichert werden kann. Der binär gespeicherte Zustand wird durch den Decoder wieder in eine Hot-One-Codierung verwandelt, bei der genau eines der 8 Zustandssignale den Wert 1 hat.

Aufgabe 5 Zustandsminimierung (7 Punkte):

Gegeben ist ein Schaltwerk durch seine Zustandstabelle.

Zustand Z	Eingabe X	Ausgabe Y	Folgezustand Z^+
Z1	0	0	Z2
Z1	1	1	Z4
Z2	0	0	Z2
Z2	1	0	Z1
Z3	0	1	Z1
Z3	1	0	Z5
Z4	0	0	Z5
Z4	1	0	Z6
Z5	0	0	Z4
Z5	1	1	Z1
Z6	0	0	Z3
Z6	1	0	Z5

a) Handelt es sich um einen Moore- oder einen Mealy-Automaten? (1 P.)

Lösung: *Es handelt sich um einen Mealy-Automaten, da y von der Eingabe abhängt.*

b) Erstellen Sie eine Liste aller Zustandspaare, die die Gleichung

$$(3.25) \forall x \in I : f(x, z_i) = f(x, z_j)$$

erfüllen, wobei f die Ausgangsfunktion des Automaten und I die Menge der möglichen Eingaben ist. (2 P.)

Lösung: Die Zustandspaare $(1,5)$, $(2,4)$, $(2,6)$ und $(4,6)$ erfüllen Gleichung (3.25).

c) Erstellen Sie für jedes Zustandspaar aus Teil b) die Liste der Folgezustandspaare, d.h. erstellen Sie die Tabelle Stufe 0 für eine Zustandsminimierung. (2 P.)

Lösung: Die Tabelle lautet:

(1,5)	(1,4) (2,4)
(2,4)	(1,6) (2,5)
(2,6)	(1,5) (2,3)
(4,6)	(3,5) (5,6)

Man sieht aus dieser Tabelle übrigens direkt, dass es bei diesem Automaten keine äquivalenten Zustände gibt.

d) Kann es ein Schaltwerk mit 5 Zuständen, 2-Bit-Eingabe und 1-Bit-Ausgabe geben, bei dem kein Paar von Zuständen die obige Gleichung (3.25) erfüllt? Begründen Sie Ihre Antwort. (2 P.)

Lösung: Ja, ein solches Schaltwerk ist möglich. Bei jeder der vier möglichen Eingaben gibt es zwei mögliche Ausgaben, d.h. insgesamt gibt es pro Zustand acht mögliche Ein-/Ausgabe-Kombinationen eines Zustands. Da es weniger als 8 Zustände gibt, kann man jedem Zustand eine andere Ein-/Ausgabe-Kombination geben, so dass kein Paar von Zuständen Gleichung (3.25) erfüllt.

Aufgabe 6 Komplexe Schaltwerke (9 Punkte + 2 Zusatzpunkte):

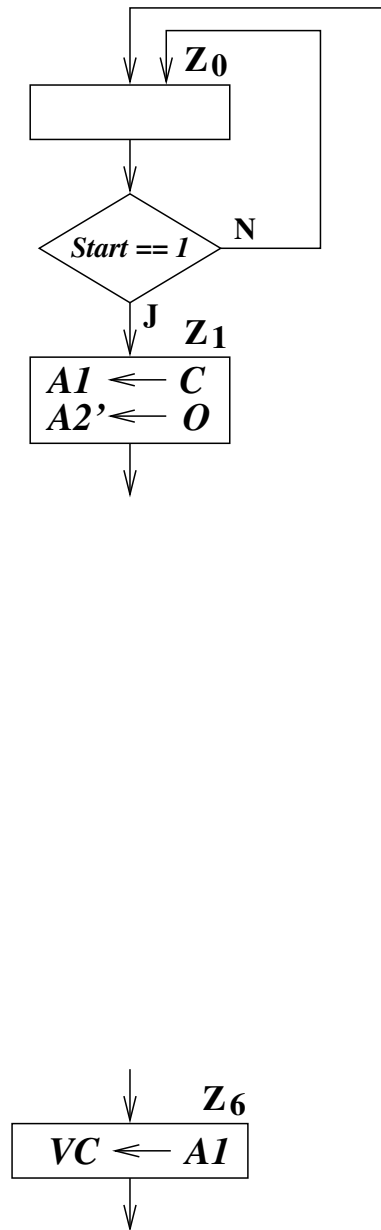
Gesucht ist ein ASM-Diagramm zur Durchführung einer sog. **Cäsar**-Kodierung auf Großbuchstaben. Die Cäsar-Kodierung beruht auf der Verschiebung eines gegebenen Klartextzeichens um einen festen *Offset*. Bei einem Offset von drei beispielsweise wird aus dem Buchstaben 'A' der Buchstabe 'D', aus dem Buchstaben 'B' wird der Buchstabe 'E', usw.

Um die Kodierung im Rechner leicht ausführen zu können, steht Ihnen ein Transformer-Schaltnetz TRAN zur Verfügung, das ein Klartextzeichen in sein Zahlenäquivalent entsprechend der Stellung im Alphabet umwandelt. Aus dem Buchstaben 'A' beispielsweise wird die Zahl 1, aus dem Buchstaben 'B' wird die Zahl 2, usw. Da das Alphabet aus 26 Buchstaben besteht, wird jeweils in eine Zahl zwischen 1 und 26 gewandelt. Mit Hilfe eines weiteren Schaltnetzes RTRAN (Rückwärts-Transformer) kann aus dem Zahlenäquivalent das dazugehörige Zeichen zurückgewonnen werden. Im Operationswerk werden die folgenden Elemente zur Verfügung gestellt:

- ein Transformer TRAN
- ein Rückwärts-Transformer RTRAN
- ein Addierer ADD
- ein Subtrahierer SUB
- ein Komparatoren (Vergleicher, ζ)
- ein Zustandsregister zur Speicherung eines Buchstabens
- zwei Zustandsregister zur Speicherung von Zahlen

Nachdem das Startsignal empfangen wurde, werden im Folge-Takt der zu kodierende Buchstabe sowie der Offset in Register geladen. Sie können davon ausgehen, dass nur Buchstaben 'A' bis 'Z' und nur Offsets 0 bis 25 eingegeben werden. In den weiteren Takten soll der eingegebene Buchstabe kodiert, die Cäsar-Verschiebung ausgeführt und die Kodierung wieder aufgelöst werden. Das Ergebnis soll anschließend auf dem Ausgang *VC* ausgegeben werden. Erhält das Schaltwerk beispielsweise 'A' und 3 als Eingaben, soll es 'D' ausgeben.

- a) Ergänzen Sie das folgende ASM-Diagramm für den oben beschriebenen Cäsar-Kodierer. Dabei dürfen keine bedingten Ausgangsboxen benutzt werden. (5 P.) Die Variable C steht für den zur Kodierung übergebenen Buchstaben, die Variable O für den gewählten Offset (zahl).



Lösung:

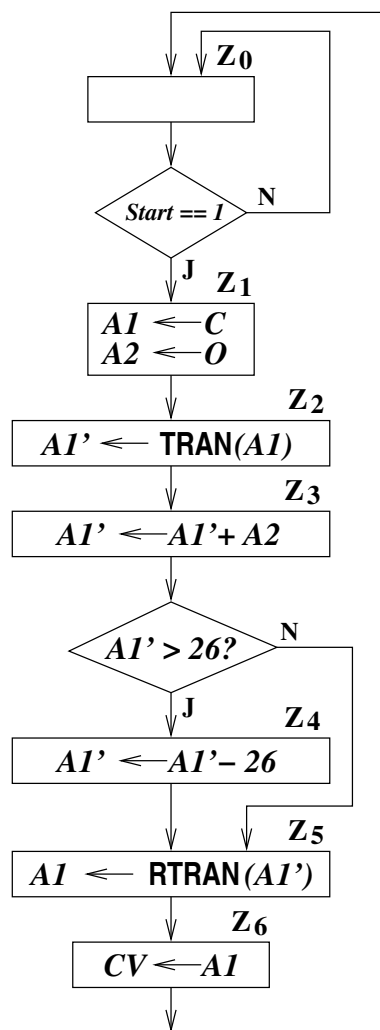
a) ASM-Diagramm des Cäsar-Kodierers

Um die Zeichenverschiebung auszuführen, wird das in Register A1 eingelesene Zeichen A1 zuerst in das entsprechende Zahlenäquivalent gewandelt und in Register A1' gespeichert. Jetzt muss zum Zahlenäquivalent in Register A1' der in Register A2' gespeicherte Offset addiert werden.

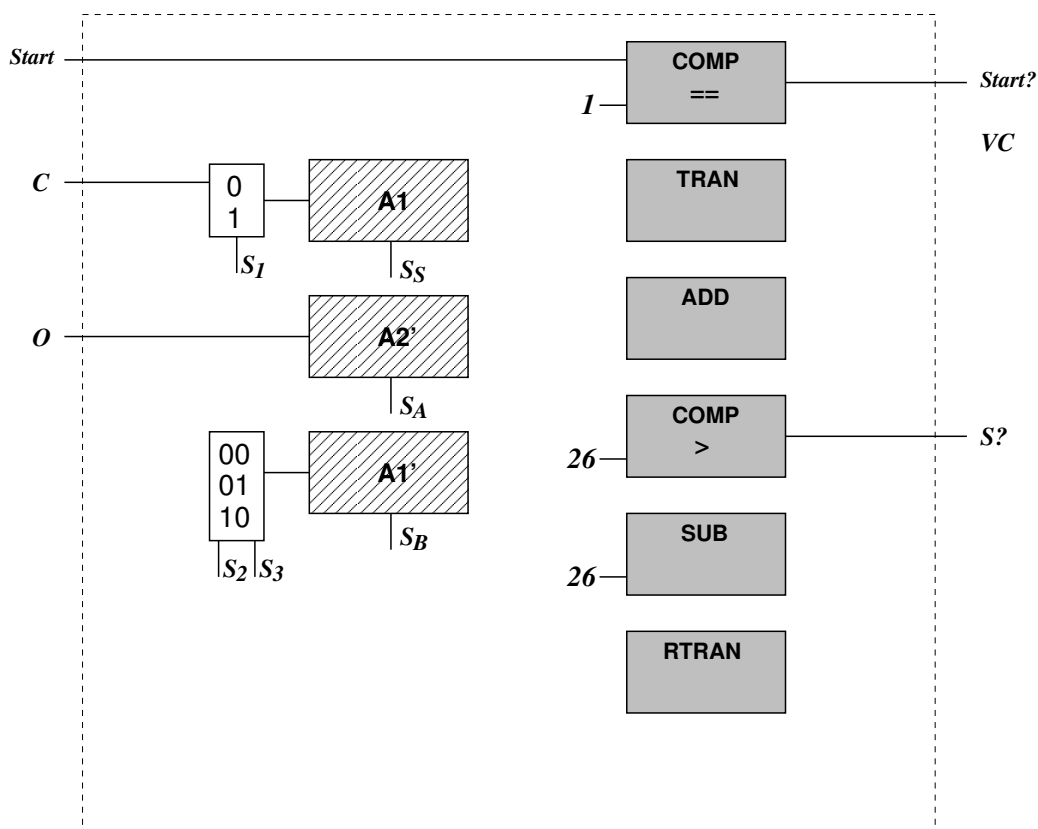
Damit ein Zahlenäquivalent größer als 26 am Ende des Alphabets auf ein Zahlenäquivalent am Anfang des Alphabets geschoben wird, ist eine Abfrage des Register A1' notwendig.

Bei einem Zahlenäquivalent größer als 26 muss die Zahl 26 vom Register A1' subtrahiert werden um auf das korrekte Zeichen im Anfang des Alphabetes zu verschieben. Schließlich muss das in Register A1' gespeicherte Zahlenäquivalent in das entsprechende Zeichen gewandelt werden.

Es sind also insgesamt 4 Operationen, also 4 Anweisungsboxen und eine Entscheidungsbox erforderlich.



b) Ergänzen Sie das folgende Schaltbild des Operationswerks für das beschriebene Additions-/Subtraktionswerk. (4 P.)



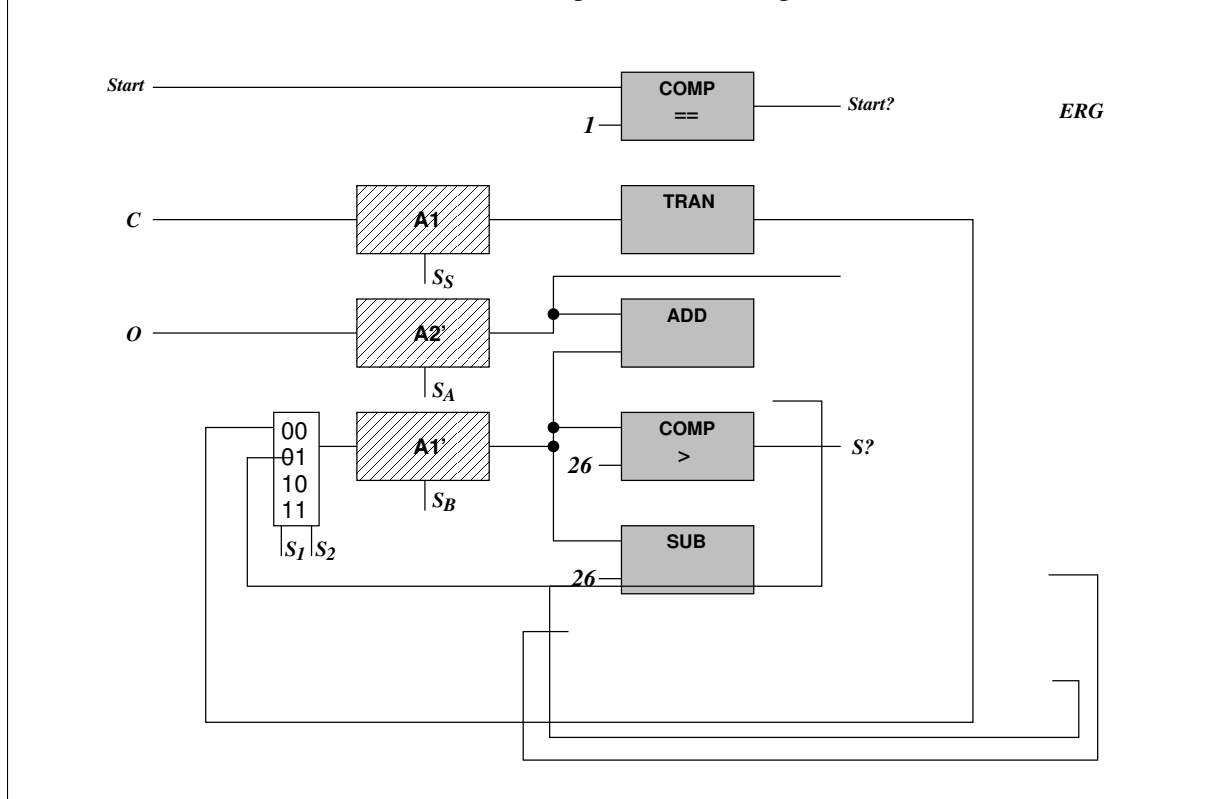
Lösung: b) Entwurf des Operationswerks

Für die Wandlung in Zustand Z_2 wird eine Verbindung von Register A1 auf den Transformer und von da aus weiter auf den MUX vor Register A1' benötigt.

Für die Addition in Zustand Z_3 wird je eine Verbindung von Register A1' und A2' auf den Addierer und von da aus weiter auf den MUX vor Register A1' benötigt.

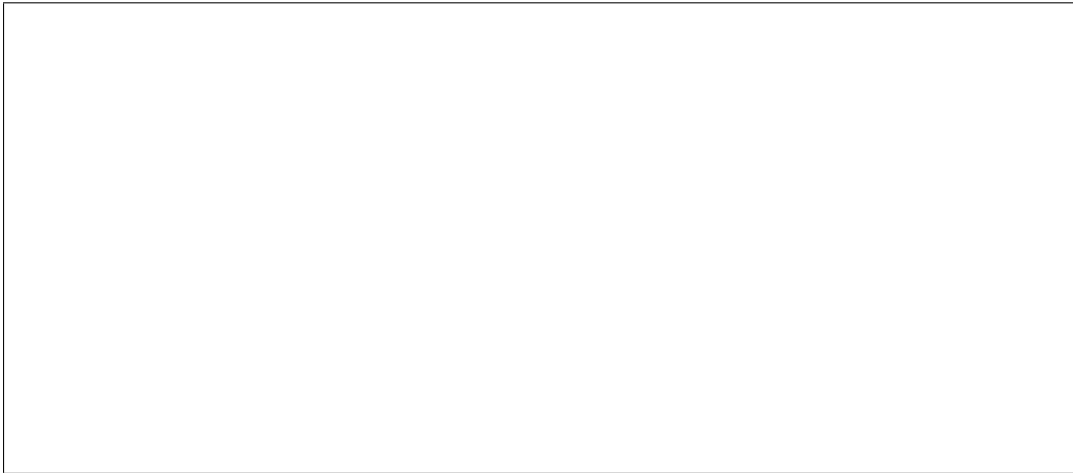
Für die Subtraktion in Zustand Z_4 wird eine Verbindung von Register A1' auf den Subtrahierer und von da aus weiter auf den MUX vor A1' benötigt.

Für die Wandlung in Zustand Z_5 wird eine Verbindung von Register A1' auf den Transformer und von da aus weiter auf den MUX vor Register A1 benötigt.



c) Möglichkeit für 2 Zusatzpunkte:

Könnte man auf den Komparator verzichten, wenn als einzugebende „Buchstaben“ sowohl Groß-Buchstaben A bis Z als auch Kleinbuchstaben a bis z, die Ziffern 0 bis 9 und die Zeichen '+' und '-' zugelassen wären, und die Buchstaben von 0 an codiert werden? Begründen Sie Ihre Antwort.



Lösung: c) Die Anzahl der eingegebenen Zeichen ist nun 64, d.h. sie werden von 0 bis 63 codiert. Benutzt man einen 6-Bit-Addierer, um den Offset zu addieren, so ist es ausreichend, die Summenbits 0 bis 5 weiterzuleiten. Diese sind wieder im Bereich 0 bis 63. Die Entscheidungsbox und damit der Komparator werden nicht mehr benötigt.