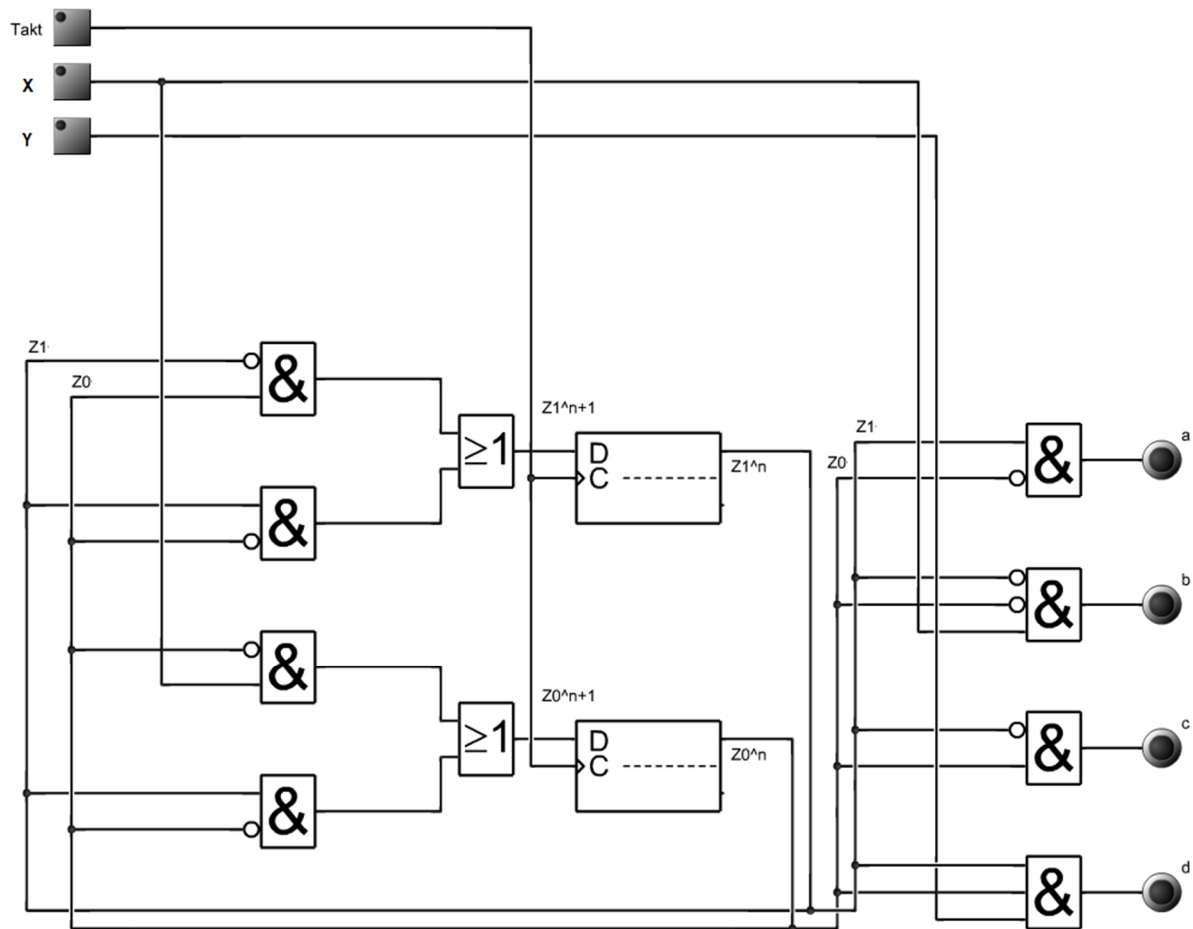


Aufgabe 1: Schaltwerksanalyse

(17 Punkte)

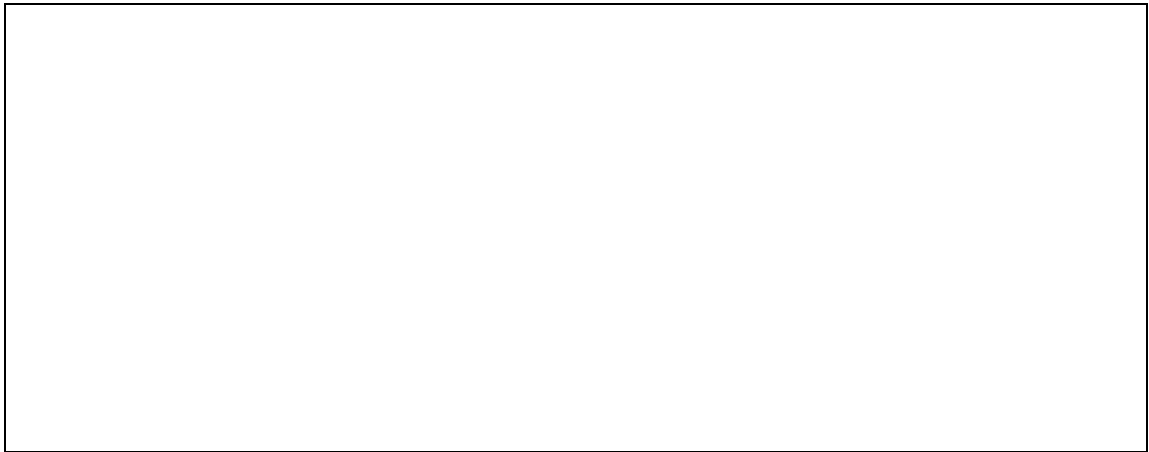
Gegeben Sei folgende Schaltung eines Automaten:



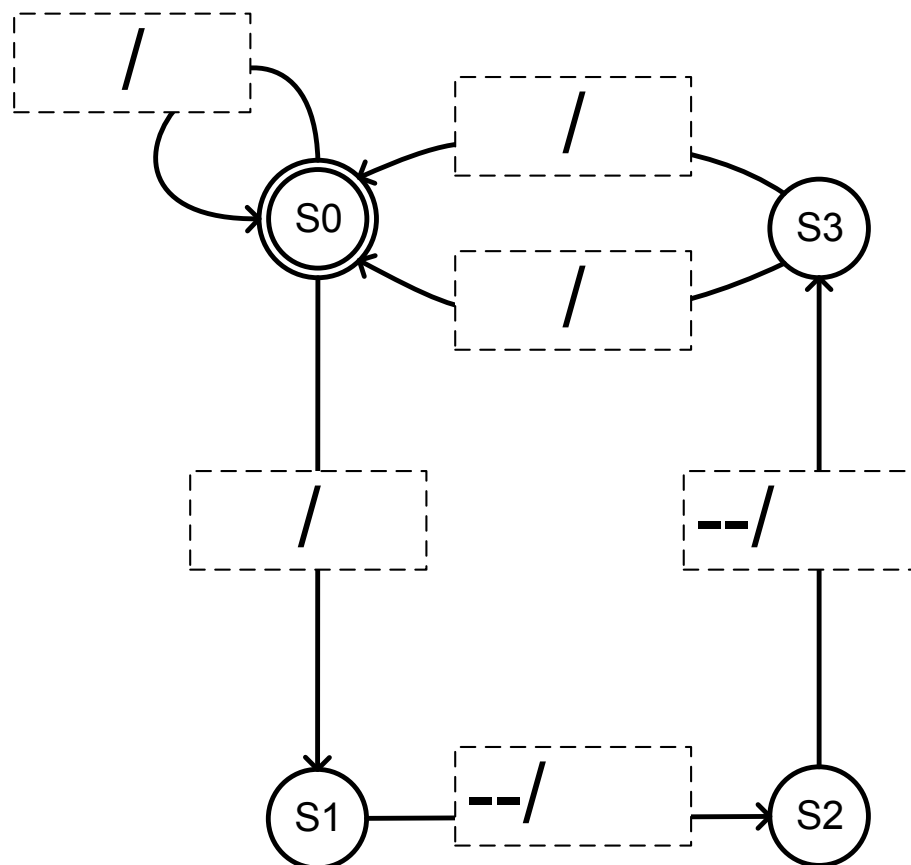
- a) Um welchen Automatentyp handelt es sich? Geben Sie eine kurze Begründung an

- b) Ermitteln Sie die Ansteuergleichungen der D-FlipFlops und die Zustandsübergangsfunktionen.

- c) Ermitteln Sie die Ausgangsgleichungen der Ausgaben a, b, c und d.



- d) Der durch die obige Schaltung gegebene Automat durchläuft sequentiell die Zustände S_0 bis S_3 . Die Zustandskodierung erfolgte durch Umrechnung der Zustandsnummer in die Binar­darstellung ($S_0 - 00_2, \dots, S_3 - 11_2 = Z_1Z_0$). Ergänzen Sie den nachfolgenden Zustands­übergangsgraphen unter der Verwendung der Zustandsübergangsfunktionen, indem Sie die offenen Felder ausfüllen. Es genügt die Angabe der jeweils aktiven Ausgaben! Die Kodierung der Eingaben sei $(X\ Y)$.



- e) Gegeben sei folgende Zustandsübergangstabelle:

$Z_1^{n+1} Z_0^{n+1}$				
	ab			
$Z_1^n Z_0^n$	00	01	10	11
00	00	01	01	01
01	10	10	10	10
10	00	00	10	10
11	xx	xx	xx	xx

Geben Sie die Zustandsübergangsfunktion für Z_1^n in **disjunktiver** und für Z_0^n in **konjunktiver** Minimalform an! Nutzen Sie für die Bestimmung der Übergangsfunktionen folgende KV-Diagramme. Füllen Sie diese komplett aus und kennzeichnen Sie Ihre Minimierungen!

Z_1^{n+1}		$(a,b)^n$			
		00	01	11	10
$(Z_1 Z_0)^n$	00				
	01				
	11				
	10				

Z_0^{n+1}		$(a,b)^n$			
		00	01	11	10
$(Z_1 Z_0)^n$	00				
	01				
	11				
	10				

Zustandsübergangsfunktionen:

$\Sigma_{A1} =$ /17 Pkt

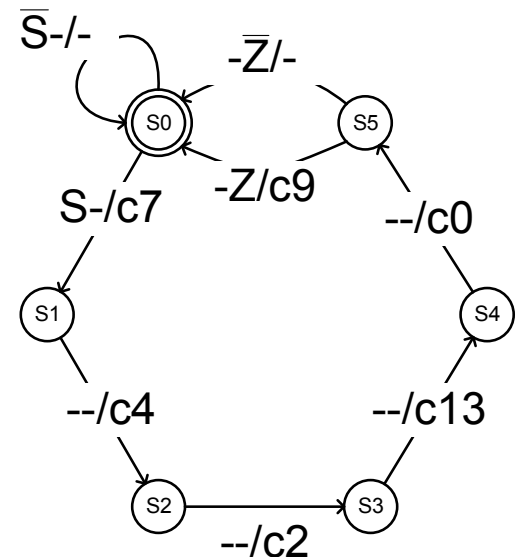
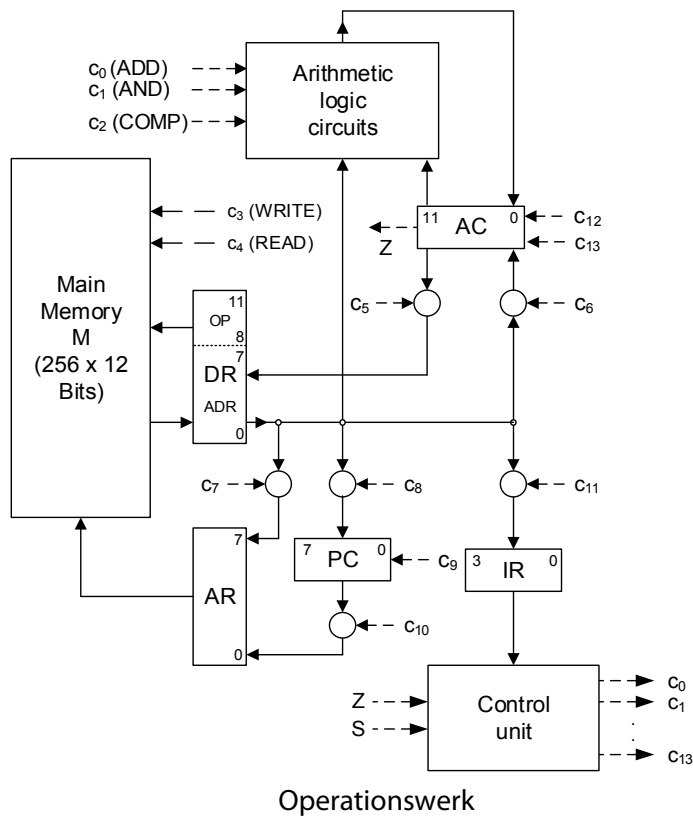
Matrikelnummer: _____

Studiengang: _____

Aufgabe 2: Entwurf auf Registertransfer Ebene

(8 Punkte)

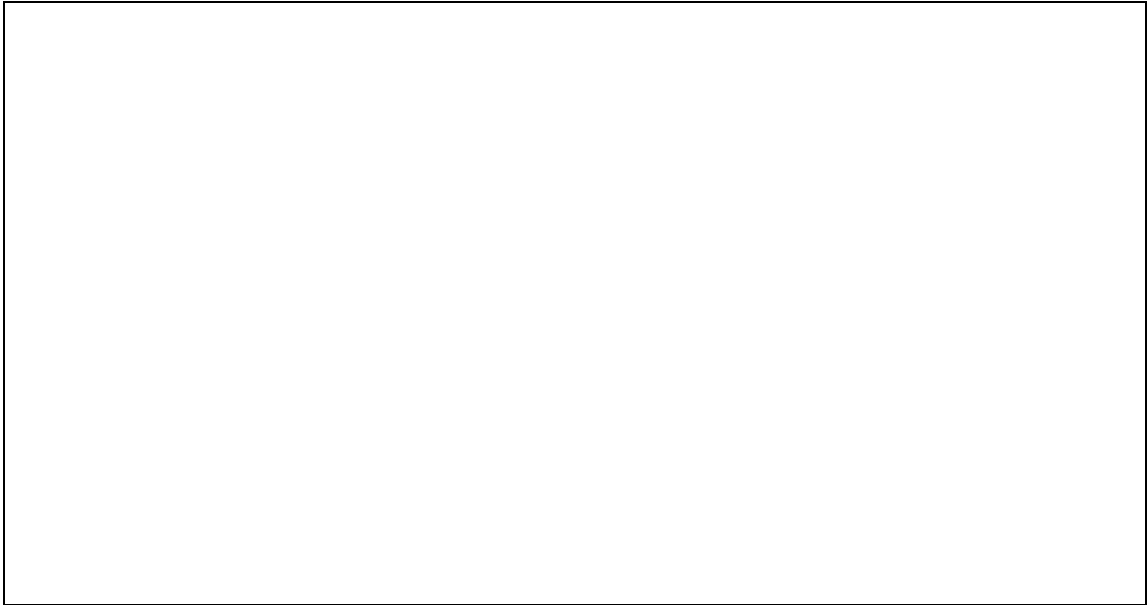
- a) Bei dem nachfolgenden Automaten handelt es sich um ein festverdrahtetes Steuerwerk, welches auf dem gegebenen Operationswerk einer CPU einen Befehl realisiert. Das Z-Flag gibt an, ob $AC = 0$ erfüllt ist ($AC = 0 \Rightarrow Z = 1$).



Die hier benötigten Steuersignale sind:

c_0	$AC \leftarrow AC + DR$
c_2	$AC \leftarrow \text{not } AC$
c_4	READ M
c_7	$AR \leftarrow DR(7:0)$
c_9	$PC \leftarrow PC + 1$
c_{13}	$AC \leftarrow AC + 1$

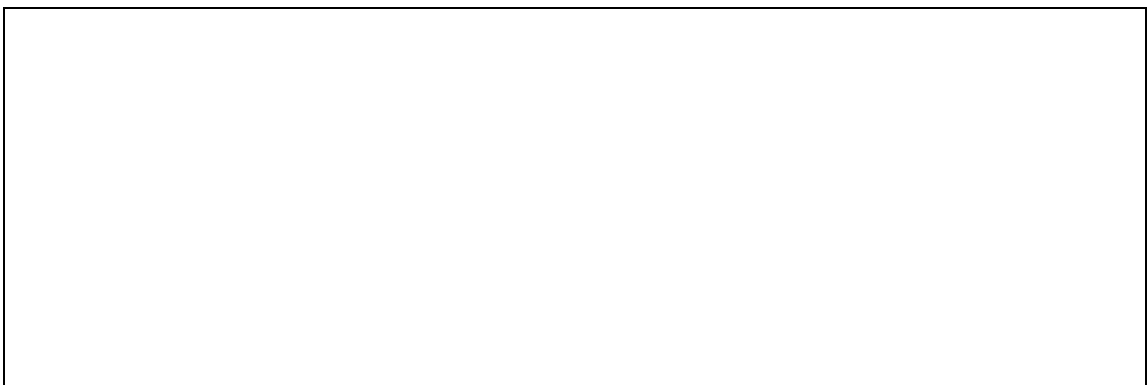
Beschreiben Sie das durch den Automaten definierte Verhalten als RT-Programm bzw. in RT-Notation. Achten Sie auf das Timing!

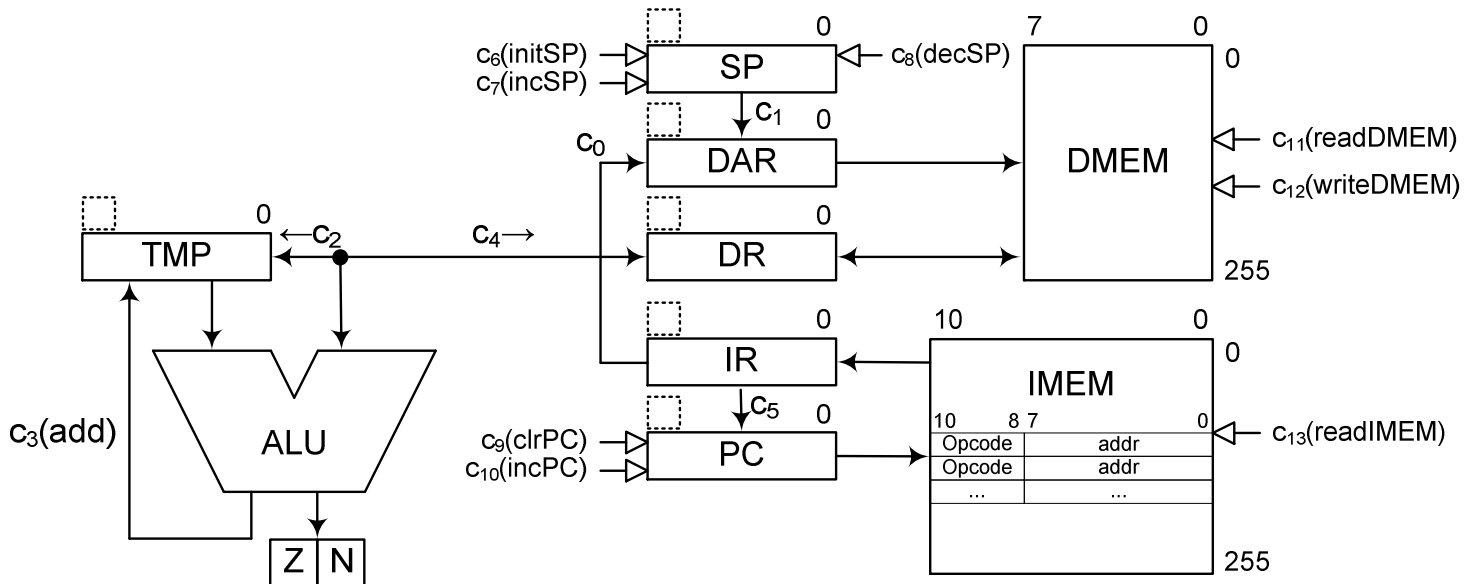


- b) Geben Sie die semantische Bedeutung (KEINE zeilenweise Beschreibung des RT-Codes) des Befehls an!



- c) Dieses Verhalten kann mit Hilfe eines Automaten mit gleichen Ein- und Ausgaben jedoch zwei Zuständen weniger realisiert werden. Bitte geben Sie eine Begründung unter Zuhilfenahme Ihres RT-Codes an! Welche Vorteile bringt das? Nennen Sie zwei!



$\Sigma_{A2} =$ /8 Pkt**Aufgabe 3: Mikroprogrammierte CPU-Kontrolleinheit (20 Punkte)**

Das oben vorgegebene Blockschaltbild zeigt das Operationswerk der in dieser Aufgabe zu untersuchenden mikroprogrammierten CPU.

Die ALU beherrscht ausschließlich die Addition (add). Das Z und N-Flag werden von der ALU automatisch erzeugt und haben jeweils 1 Bit Breite. Es gilt: $Z=1$, wenn die letzte Operation der ALU eine 0 lieferte, sonst $Z=0$. Weiter gilt $N=1$, wenn die letzte Operation der ALU ein negatives Ergebnis lieferte, sonst $N=0$.

Eine Registertransferbeschreibung einiger Befehle der CPU ist Ihnen vorgegeben:

Opcode	Befehl	Beschreibung
0	PUSH addr	Legt den Speicherinhalt unter der Adresse addr auf den Stack ab.
1	POP addr	Entfernt das oberste Stackelement und legt den Wert im Speicher unter der Adresse addr ab.
2	ADD addr	Entfernt das oberste Stackelement und addiert es auf den Wert unter der Adresse addr .
3	BRGR addr	Setzt den Programmablauf an Adresse addr fort, wenn $TMP > 0$ war.

- Ergänzen Sie im Blockschaltbild die fehlenden Indizes der Registerbreiten in den vorgesehenen Kästen.
- Ergänzen Sie in dem auf der nächsten Seite vorgegebenen RT-Programm die frei gelassenen Registerbreiten und -indizes sowie die Deklaration der Speicher. Ergänzen Sie zudem in den vorgesehenen Feldern (rechts) die **in der Abbildung vorgegebene** Kontrollsignale.
- Beschreiben Sie die Funktion des Stacks anhand der folgenden Fragestellungen:

- In welche Richtung (Speicherende oder Speicheranfang) wächst der Stack?
- Auf welche Adresse zeigt der Stackpointer SP nach jeder Operation?


```
declare register SP(   ), DAR(   ), DR(   ), IR(   ) PC(   ), TMP(   ) , Z, N
declare memory DMEM(   ), IMEM(   )
```

```
INIT:          PC <- 0, SP <- 255;
```

```
FETCH:  read IMEM;
```

```
DAR <- IR(   ), PC <- PC + 1 | switch IR(   ){
    case 0: goto PUSH
    case 1: goto POP
    case 2: goto ADD
    case 3: goto BRGR
    default: goto FETCH
};
```

```
PUSH:          read DMEM;
```

```
SP <- SP - 1, write DMEM | goto FETCH;
```

```
POP:           SP <- SP + 1;
```

```
DAR <- SP;
```

```
read DMEM;
```

```
DAR <- IR(   );
```

```
write DMEM | goto FETCH;
```

```
ADD:           SP <- SP + 1;
```

```
DAR <- SP;
```

```
read DMEM;
```

```
TMP <- DR, DAR <- IR(   );
```

```
read DMEM;
```

```
    TMP <- DR + TMP;
```

```
DR <- TMP;
```

```
write DMEM | goto FETCH;
```

```
BRGR:  if N = 0 AND Z = 0 then PC <- IR(   ) fi |
goto FETCH;
```

- d) Implementieren Sie die folgenden Befehle selbst und **geben Sie auch die Kontrollsignale an**. Sie müssen den FETCH nicht weiter anpassen.

ADDS: Entfernt die beiden obersten Stackelemente und legt ihre Summe wieder auf dem Stack ab.

BRLT addr: Setzt den Programmablauf an Adresse addr fort, wenn das Ergebnis der letzten Operation echt kleiner als Null war.

- e) Erstellen Sie nun ein vertikales mikroprogrammiertes Steuerwerk auf Basis des **vorgegebenen Programms** aus Aufgabe b). **Halten Sie sich, falls möglich, an das vorgegebene Timing.** Es stehen Ihnen ausschließlich die folgenden Condition Select Signale zur Verfügung:

Condition Select	Funktion
000	Nicht springen
001	Springe zu der dekodierten Opcode-Adresse (Mapping ROM)
100	Springe, falls Z = 0
101	Springe, falls N = 0
111	Springe immer

Hinweise:

- Es ist in den folgenden Tabellen ausreichend, nur die 1en auszufüllen. Wir nehmen an, dass alle nicht ausgefüllten Felder auf 0 gesetzt sind.
- Sie müssen die Tabellen ggf. nicht voll ausfüllen, einige Zeilen können leer bleiben.
- Verwenden Sie das Ihnen aus der Vorlesung bekannte Mapping ROM.

Gehen Sie wie folgt vor:

- (i) Bestimmen Sie eine minimale Kodierung der Kontrollsignale aus b).

C ₁₃	C ₁₂	C ₁₁	C ₁₀	C ₉	C ₈	C ₇	C ₆	C ₅	C ₄	C ₃	C ₂	C ₁	C ₀	Kodierung			

- (ii) Erstellen Sie nun das vertikale Mikroprogramm in der vorgegebenen Tabelle. **Ergänzen Sie den mit den gegebenen Condition Selects umgesetzten RT-Code** für BRGR.

Matrikelnummer: _____

Studiengang: _____

addr					CS			Next_addr					c_vert				RT-Code
0	0	0	0	0													INIT: PC <- 0, SP <- 255;
0	0	0	0	1													FETCH: read IMEM;
0	0	0	1	0													DAR <- IR(), PC <- PC + 1 switch IR ()
0	0	0	1	1													PUSH: read DMEM;
0	0	1	0	0													SP <- SP - 1, write DMEM goto FETCH;
0	0	1	0	1													POP: SP <- SP + 1;
0	0	1	1	0													DAR <- SP;
0	0	1	1	1													read DMEM;
0	1	0	0	0													DAR <- IR();
0	1	0	0	1													write DMEM goto FETCH;
0	1	0	1	0													ADD: SP <- SP + 1;
0	1	0	1	1													DAR <- SP;
0	1	1	0	0													read DMEM;
0	1	1	0	1													TMP <- DR + TMP;
0	1	1	1	0													DR <- TMP;
0	1	1	1	1													write DMEM goto FETCH;
1	0	0	0	0													BRGR:
1	0	0	0	1													
1	0	0	1	0													
1	0	0	1	1													
1	0	1	0	0													
1	0	1	0	1													
1	0	1	1	0													
1	0	1	1	1													
1	1	0	0	0													

(iii) Geben Sie das Mapping ROM an.

Opcode	Sprungadresse

 $\Sigma_{A3} =$ /20 Pkt

Aufgabe 4: Allgemeine Fragen

(5 Punkte)

Beantworten Sie die folgenden Fragen.

- a) Was ist der Unterschied zwischen einem Schaltnetz und einem Schaltwerk?

- b) Worin liegen die Vorteile und Nachteile eines Carry-Look-Ahead-Addierers gegenüber einem gewöhnlichen Ripple-Carry-Adder?

- c) Worin unterscheiden sich das RS-Flipflop und das JK-Flipflop?

- d) Was ist der Unterschied zwischen horizontaler zu vertikaler Mikroprogrammierung?

- e) Welche Operationen muss ein Program Counter (PC) üblicherweise unterstützen?

$\Sigma_{A4} =$ /5 Pkt