

# Technische Grundlagen der Informatik - WS 2019/20

## Aufgabenblatt 6

---

**Abgabe:** Das Übungsblatt mit **Name** und **Übungsgruppe** oben rechts geschrieben darf bis **Mo 27.01.2020 15:00 Uhr** abgegeben werden.

**Achtung:** Übungsblätter die keiner Person und Übungsgruppe zuzuordnen sind, werden nicht korrigiert!

---

**Hinweis:** Die zweite Übung findet in der Woche vom **03.02.2020 - 07.02.2020** statt.

## Aufgabe 1 (Zähler)

Entwerfen Sie einen 3-Bit-Synchronzähler, der folgende Zählfolge aufweisen soll:

$0 \rightarrow 4 \rightarrow 7 \rightarrow 2 \rightarrow 3 \rightarrow 0$  (*Wiederholung*)  $\rightarrow \dots$

Verwenden Sie T-FlipFlops.

1. Geben Sie das Zustandsdiagramm des Zählers an. (4P)
2. Stellen Sie die Übergangstabelle auf. (10P)
3. Geben Sie die minimierten Funktionen der T-Eingänge der FlipFlops an. (4P)
4. Zeichnen Sie das Schaltwerk. Dabei können Sie die einzelnen Funktionen  $T_i$  jeweils zu einer Komponente zusammenfassen. Das Übergangsschaltnetz muss also nicht aus Grundgattern aufgebaut werden. (4P)
5. Um welche Art von Automaten handelt es sich hier? (2P)

## Aufgabe 2 (sequentielle Schaltung und Zustandsdiagramme)

1. Benutzen Sie einen 1 Bit Komparator aus Kapitel 10 (**10. kombinatorische Komponenten, Seite 24**) um einen n Bit Komparator als sequentielle Schaltung zu realisieren. (4P)
2. Entwerfen Sie das Zustandsdiagramm eines Moore-Automaten, welcher kurzzeitige Störungen unterdrückt. Der Ausgang  $y$  soll erst dann 1 werden, nachdem der Eingang  $x$  vier Takte lang ununterbrochen 1 gewesen ist. Außerdem soll  $y$  erst dann wieder 0 werden, nachdem  $x$  vier Takte lang ununterbrochen 0 gewesen ist. (6P)

## Aufgabe 3 (Zustandsminimierung)

1. Minimieren Sie die Zustandstabelle aus Abbildung 1 mit dem Row-Matching Verfahren. (6P)

## Aufgabe 4 (Automatenentwurf)

Entwickeln Sie einen **Mealy**-Automaten mit einem Eingang  $x$  und einem Ausgang  $y$ , welcher den Ausgangswert 1 liefert, wenn die Eingangssequenz 110 oder 101 auftritt. Sich überlappende Sequenzen (z.B. 1101) sollen detektiert werden.

1. Geben Sie ein Zustandsdiagramm mit einer minimalen Anzahl an Zuständen an. (6P)
2. Kodieren Sie die Zustände und geben Sie die kodierte Ausgangstabelle an. (6P)
3. Der Automat soll jetzt mit JK-FlipFlops realisiert werden. Geben Sie hierzu eine Ansteuertabelle an. (6P)

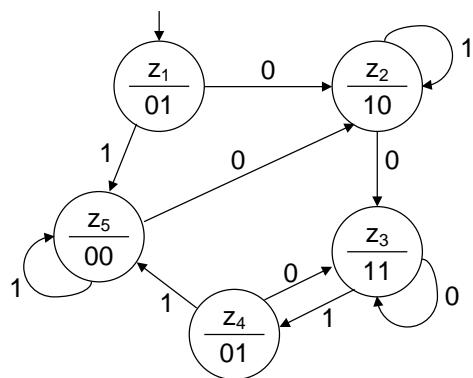
Zustand $Z^t$	Zustand $Z^{t+1}$		Ausgang y	
	$x = 0$	$x = 1$	$x = 0$	$x = 1$
A	B	C	0	0
B	D	E	0	0
C	F	G	0	0
D	H	I	0	0
E	J	K	0	0
F	L	M	0	0
G	N	P	0	0
H	A	A	0	0
I	A	A	0	0
J	A	A	0	1
K	A	A	0	0
L	A	A	0	1
M	A	A	0	0
N	A	A	0	0
P	A	A	0	0

**Abbildung 1:** Zustandstabelle eines Automaten

4. Bestimmen Sie die Funktionen für die Eingänge der FlipFlops. (6P)

### Aufgabe 5 (Moore-Mealy-Umwandlungen)

1. Wandeln Sie folgenden Moore-Automaten in einen Mealy-Automaten um (Abbildung 2). (8P)



**Abbildung 2:** Zustandsdiagramm eines Moore-Automaten

2. Wandeln Sie folgenden Mealy-Automaten in einen Moore-Automaten um (Abbildung 3). (8P)

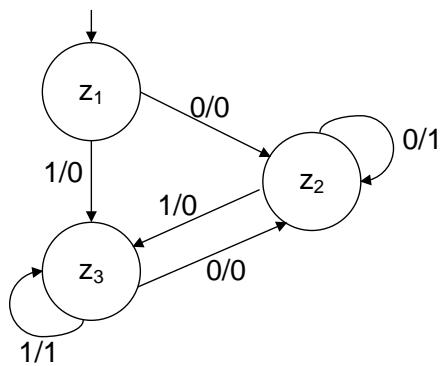


Abbildung 3: Zustandsdiagramm eines Mealy-Automaten

### Aufgabe 6 (Minimierung mit Implikationstabelle)

Gegeben ist die Zustandstabelle aus Abbildung 4. Für diese Zustandstabelle soll eine Zustandsminimierung mit einer Implikationstabelle durchgeführt werden.

Zustand $Z^t$	Zustand $Z^{t+1}$		Ausgang y	
	$x = 0$	$x = 1$	$x = 0$	$x = 1$
A	A	E	1	0
B	C	F	0	0
C	B	H	0	0
D	E	F	0	0
E	D	A	0	0
F	B	F	1	0
G	D	H	0	0
H	H	G	1	0

Abbildung 4: Zustandstabelle eines Automaten

1. Stellen Sie zur Zustandstabelle die Implikationstabelle auf. (8P)
2. Minimieren Sie die Zustände mit der Implikationstabelle. (8P)
3. Geben Sie äquivalente Zustände an und stellen Sie die reduzierte Zustandstabelle auf. (4P)