

Prof. Dr. Christoph Scholl  
Tobias Seufert

Freiburg, 05. Juli 2023

## Technische Informatik Musterlösung zu Übungsblatt 11

*Hinweis: Auf diesem Blatt befindet sich (mindestens) eine „Bonusaufgabe“. Diese Aufgabe zählt nicht in die Gesamtheit der Aufgaben, bei sinnvoller Bearbeitung wird sie jedoch zur Menge der sinnvoll bearbeiteten Aufgaben gerechnet.*

### Aufgabe 1 (Bonusaufgabe: 2 Punkte)

Sie sollten am 26.06.2023 eine E-Mail zur Vorlesungsevaluation erhalten haben. Füllen Sie den Fragebogen zu dieser Vorlesung bitte bis zum Ende der Evaluation am 09.07.2023 aus.

Lösen Sie diese Aufgabe durch Angabe der folgenden oder einer ähnlichen Aussage: “Ich habe die Vorlesungsevaluation abgeschickt.” Vielen Dank!

### Aufgabe 2 (2 + 2 + 1 + 3 Punkte)

Betrachten Sie den Mealy-Automaten  $M = (\{0, 1\}, \{0, 1\}, \{s_1, \dots, s_6\}, \{s_1\}, \delta, \lambda)$ , der durch folgende Zustands- und Ausgangstafel definiert ist:

$s^t$	$x$	$s^{t+1}$	$y$
$s_1$	0	$s_5$	0
$s_1$	1	$s_2$	0
$s_2$	0	$s_3$	1
$s_2$	1	$s_1$	1
$s_3$	0	$s_2$	0
$s_3$	1	$s_4$	1
$s_4$	0	$s_3$	1
$s_4$	1	$s_1$	1
$s_5$	0	$s_6$	1
$s_5$	1	$s_1$	1
$s_6$	0	$s_5$	0
$s_6$	1	$s_2$	0

Dabei ist  $s^t$  der aktuelle Zustand und  $s^{t+1}$  der Folgezustand. Der Startzustand des Mealy-Automaten sei  $s_1$ .

Konstruieren Sie ausgehend von diesem Mealy-Automaten ein möglichst kleines Schaltwerk mit dem gleichen sequentiellen Verhalten. Gehen Sie dazu wie folgt vor:

- a) Zeichnen Sie für den oben definierten Mealy-Automaten das entsprechende Zustandsdiagramm.
- b) Prüfen Sie das Zustandsdiagramm auf äquivalente Zustände. Fassen Sie zwei Zustände dann zusammen, wenn die in der Vorlesung vorgestellte *hinreichende* Bedingung gilt: Wenn zwei Zustände bei der gleichen Eingabe immer die gleiche Ausgabe liefern und denselben Folgezustand annehmen, dann sind die Zustände äquivalent.

Geben Sie die von Ihnen erkannten Äquivalenzen an und markieren Sie sie in dem Zustandsdiagramm aus a). Zeichnen Sie ein reduziertes Zustandsdiagramm, in dem die äquivalenten Zustände zusammengefasst wurden.

*Hinweis:* Sie sollten in dem reduzierten Zustandsdiagramm mit 4 Zuständen auskommen.

- c) Wählen Sie nun eine Zustandskodierung für die Zustände im reduzierten Zustandsdiagramm und geben Sie die neue Zustands- und Ausgangstafel an. Verwenden Sie in der Zustands- und Ausgangstafel Ihre Zustandskodierung.
- d) Geben Sie eine resultierende Ausgangsfunktion und Übergangsfunktionen an und konstruieren Sie aus dem reduzierten Zustandsdiagramm ein Schaltwerk, das möglichst wenig Flipflops benötigt. Vergessen Sie nicht, den Startzustand der Flipflops anzugeben. *Hinweis:* Leiten Sie für Ihre Funktionen jeweils optimale Boolesche Polynome her.

## Lösung:

- a) s. Abbildung 1

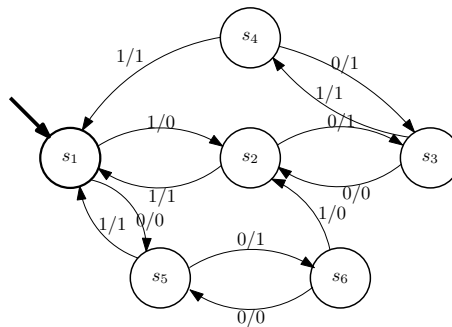


Abbildung 1: Mealy-Automat

- b) s. Abbildung 2
- c)  $s_i = (z_1, z_0)$ ,  $s_{1,6} = (0, 0)$ ,  $s_{2,4} = (0, 1)$ ,  $s_3 = (1, 0)$ ,  $s_5 = (1, 1)$ ,

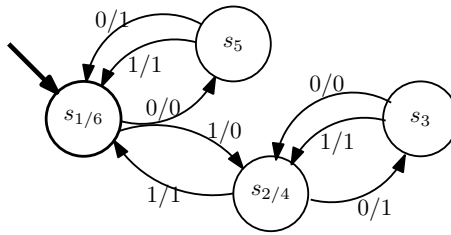


Abbildung 2: Reduzierter Mealy-Automat

$x$	state		next state		$y$
	$z_1^t$	$z_0^t$	$z_1^{t+1}$	$z_0^{t+1}$	
0	0	0	1	1	0
1	0	0	0	1	0
0	0	1	1	0	1
1	0	1	0	0	1
0	1	0	0	1	0
1	1	0	0	1	1
0	1	1	0	0	1
1	1	1	0	0	1

(Das  $x$  und  $y$  in der Tabelle sind die Ein- und Ausgabewerte)

d) s. Abbildung 3

$$z_0^{t+1} = \bar{z}_0^t$$

$$z_1^{t+1} = x + z_1^t = \bar{x} \cdot \bar{z}_1^t$$

$$y = z_0^t + x \cdot z_1^t$$

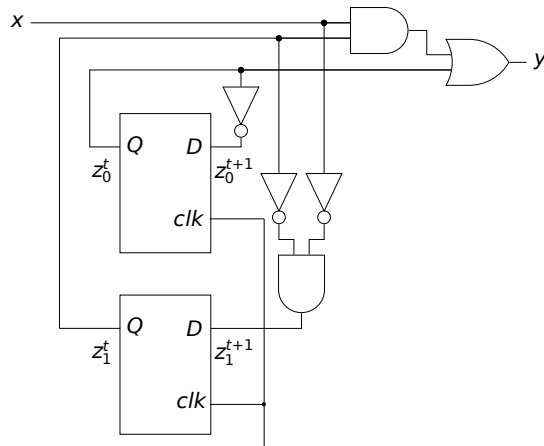


Abbildung 3: Schaltwerk

**Punkte:** Kleine Fehler bei Automatenaufbau (Kanten vergessen/fehlende Beschriftung), je [-0,25]; Folgefehler beachten

### Aufgabe 3 (3 Punkte)

Konstruieren Sie einen Mealy-Automaten, der eine Binärzahl (beliebiger Länge) inkrementiert. Gehen Sie dabei davon aus, dass die Binärzahl mit dem niederwertigsten Bit zuerst gelesen wird (d. h. von rechts nach links). Links von der Eingabe ist zweimal die Markierung (#) angebracht. Die rechte der beiden Markierungen bietet Platz für das Überlaubit in der Ausgabe, da es beim Inkrementieren potentiell zu einem Überlauf kommen kann und die Ein- und Ausgabe von Mealy-Automaten immer gleich lang sind. Die zweite Markierung links von der Überlaufmarkierung kennzeichnet das Ende der Eingabe. Diese Markierung soll nach dem Inkrementieren wieder als letztes Zeichen ausgegeben werden. Die erste Markierung soll durch das Überlaubit (0 oder 1) ersetzt werden.

Beispiel: Aus ##0011 soll #00100 werden, aus ##11111 wird #100000.

Zeichnen Sie das Zustandsdiagramm.

*Hinweis:* Falls die Eingabe nicht dem vorgegebenen Format entspricht, darf ihr Automat eine beliebige Ausgabe erzeugen. Dies gilt insbesondere dann, falls nach dem zweiten #-Symbol noch weitere Eingabesymbole folgen.

**Lösung:**

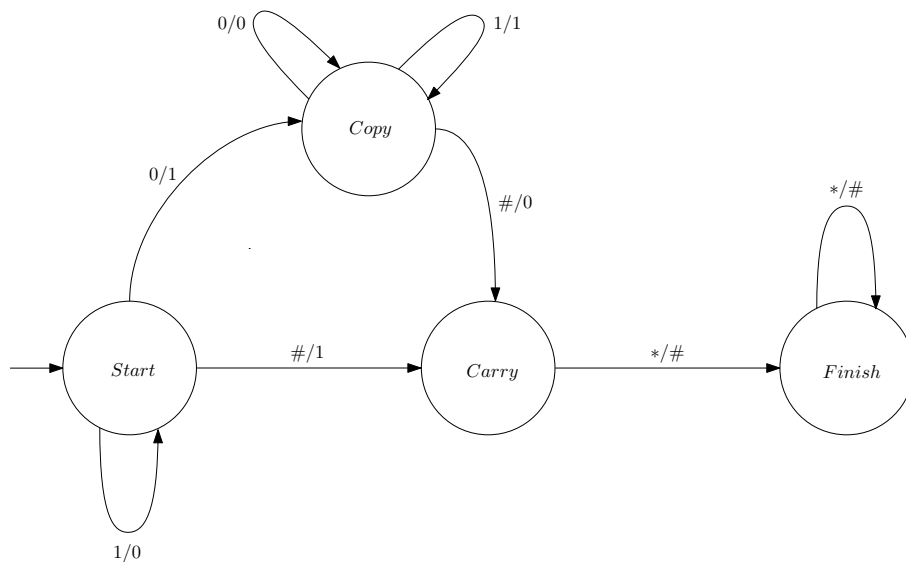


Abbildung 4: Mealy-Automat zum Inkrementieren von Binärzahlen

Zum Inkrementieren müssen vom niederwertigsten Bit an alle 1en in 0en verwandelt werden, bis man entweder zum Wortende oder zur ersten Null gelangt. Im Falle einer Null, wird diese in eine Eins verwandelt und man geht in den Zustand Copy. Hier wird bis zum Wortende alles kopiert. Liest man die erste #, so wird das Übertragsbit ausgegeben: Hat man schon 0en gefunden (Copy), ist das Übertragsbit 0, ansonsten 1. Dann ist man am Wortende angelangt (d. h. man liest zum zweiten Mal #), wird an die Zahl vorne ein # angehängt und man geht zum Endzustand.

Wegen Vollständigkeit muss jeder Zustand je eine Kante für jeden Eingabebuchstaben besitzen. \* kennzeichnet eine Kante, die für alle möglichen Eingabebuchstaben gilt.

#### Aufgabe 4 (Bonusaufgabe: 2 Punkte)

Betrachten Sie den unvollständigen Schaltkreis in Abb. 5. Geben Sie eine Gatter-Implementierung

für die Teilschaltkreise  $f$  und  $g$  an, so dass der gesamte Schaltkreis die Funktionsweise eines Tristate Treibers mit Output-Enable  $/OE$  realisiert. Zur Erinnerung: Der Tristate-Treiber gibt am Ausgang  $y$  den Wert von  $x$  wieder, falls  $/OE = 0$  und  $Z$ , falls  $/OE = 1$ .

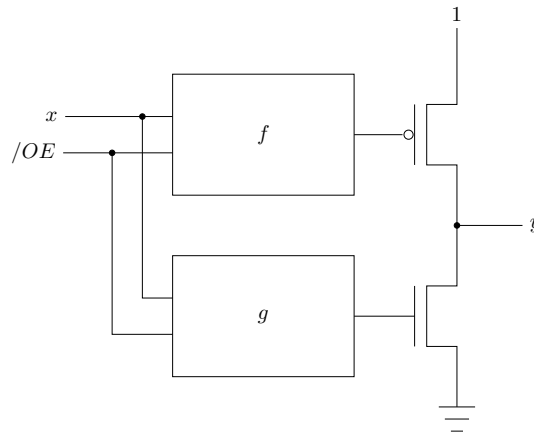


Abbildung 5: Unvollständige Implementierung eines Treibers.

### Lösung:

Die Implementierung ist in Abb. 6 dargestellt. Die Funktion  $f$  und  $g$  können auch wie folgt gezeigt werden:

$x$	$/OE$	$f$	$g$	Erklärung
0	0	1	1	NMOS schaltet durch, zieht $y$ gegen 0.
0	1	1	0	weder P- noch NMOS dürfen leiten.
1	0	0	0	PMOS schaltet durch, zieht $y$ gegen 1.
1	1	1	0	weder P- noch NMOS dürfen leiten.

- $f = \bar{x} \cdot \overline{/OE} + \bar{x} \cdot /OE + x \cdot /OE = \bar{x} + /OE = x \cdot \overline{/OE}$
- $g = \bar{x} \cdot \overline{/OE} = \overline{x + /OE}$

Hinweis:

1P pro Funktion. Jede Kombination von  $f$  und  $g$  ist richtig. Als Lösung nur eine Tabelle angegeben: -1,5P.

Abgabe: 12. Juli 2023, 13<sup>00</sup> über das Übungsportal

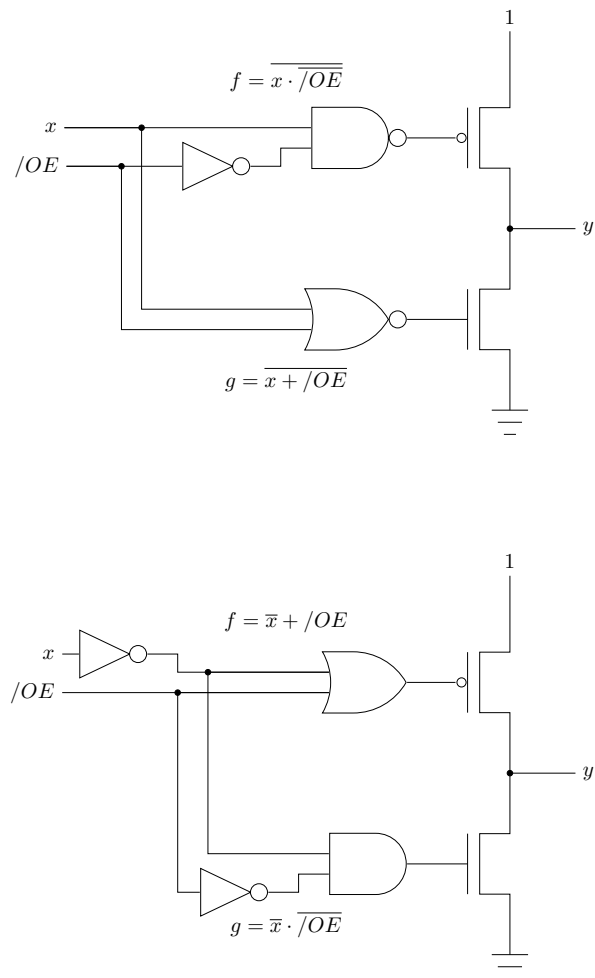


Abbildung 6: Mögliche implementierung des Treibers