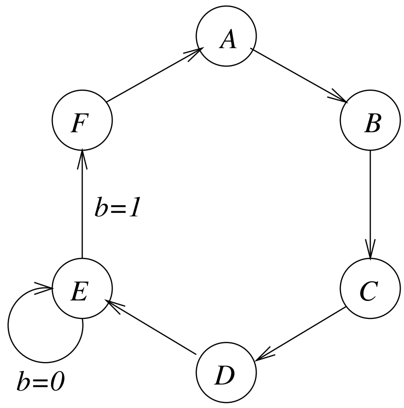
WS 2004/05	Übungen Technische Informatik 1 (T1)	Musterlösung zu Aufgabenblatt 6

Lösung 6.1

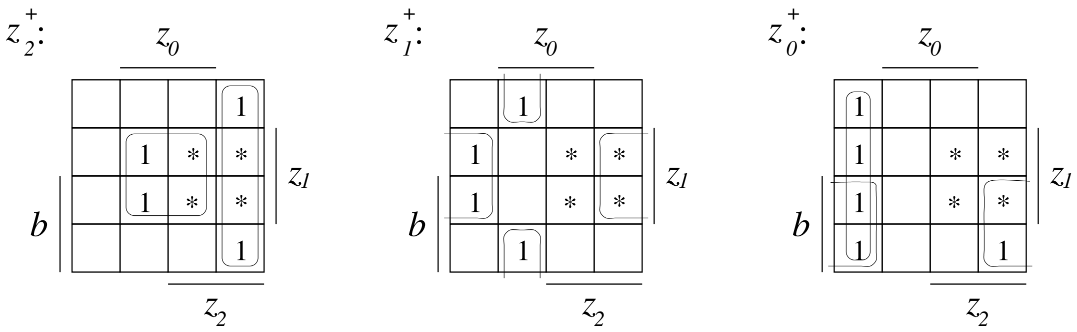
a) Der Zustandgraph für die Fußgängerampelsteuerung sieht folgendermaßen aus:



b) Für eine gewählte Zustandskodierung der Art: $A \Leftrightarrow (000)_2$, $B \Leftrightarrow (001)_2$, $C \Leftrightarrow (010)_2$, $D \Leftrightarrow (011)_2$, $E \Leftrightarrow (100)_2$, $F \Leftrightarrow (101)_2$ ergibt sich für die Übergangsfunktion:

b	z_2	z_1	z_0	z_2^+	z_1^+	z_0^+	*Text*
$*$	0	0	0	0	0	1	$A \Rightarrow B$
$*$	0	0	1	0	1	0	$B \Rightarrow C$
$*$	0	1	0	0	1	1	$C \Rightarrow D$
$*$	0	1	1	1	0	0	$D \Rightarrow E$
0	1	0	0	1	0	0	$E \Rightarrow E$
1	1	0	0	1	0	1	$E \Rightarrow F$
$*$	1	0	1	0	0	0	$F \Rightarrow A$
$*$	1	1	0	$*$	$*$	$*$	
$*$	1	1	1	$*$	$*$	$*$	

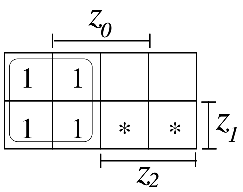
Für die Eingangsvariable b bedeuten *dont' cares* $(*)$ in der Zustandstabelle zwei Einträge im KV-Diagramm (zwei Minterme), d.h. je ein Eintrag für $b = 0$ und ein Eintrag für $b = 1$. Die nicht kodierten Zustände liefern Freiheitsgrade für die Logik-Minimierung. Damit ergeben sich für die Übergangsfunktion folgende KV-Diagramme:



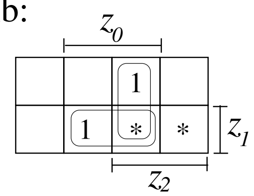
Mit der festgelegten Zustandskodierung ergibt sich aus der Aufgabenspezifikation für die Ausgangsfunktion folgende Wertetabelle:

Zustand	z_2	z_1	z_0	*rot*	*gelb*	*grün*	*stop*	*gehen*
A	0	0	0	1	0	0	1	0
B	0	0	1	1	0	0	0	1
C	0	1	0	1	0	0	1	0
D	0	1	1	1	1	0	1	0
E	1	0	0	0	0	1	1	0
F	1	0	1	0	1	0	1	0
	1	1	0	*	*	*	*	*
	1	1	1	*	*	*	*	*

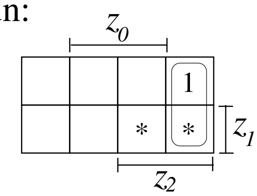
rot:



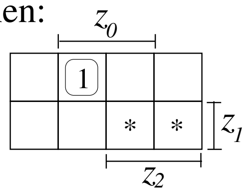
gelb:



grün:



gehen:

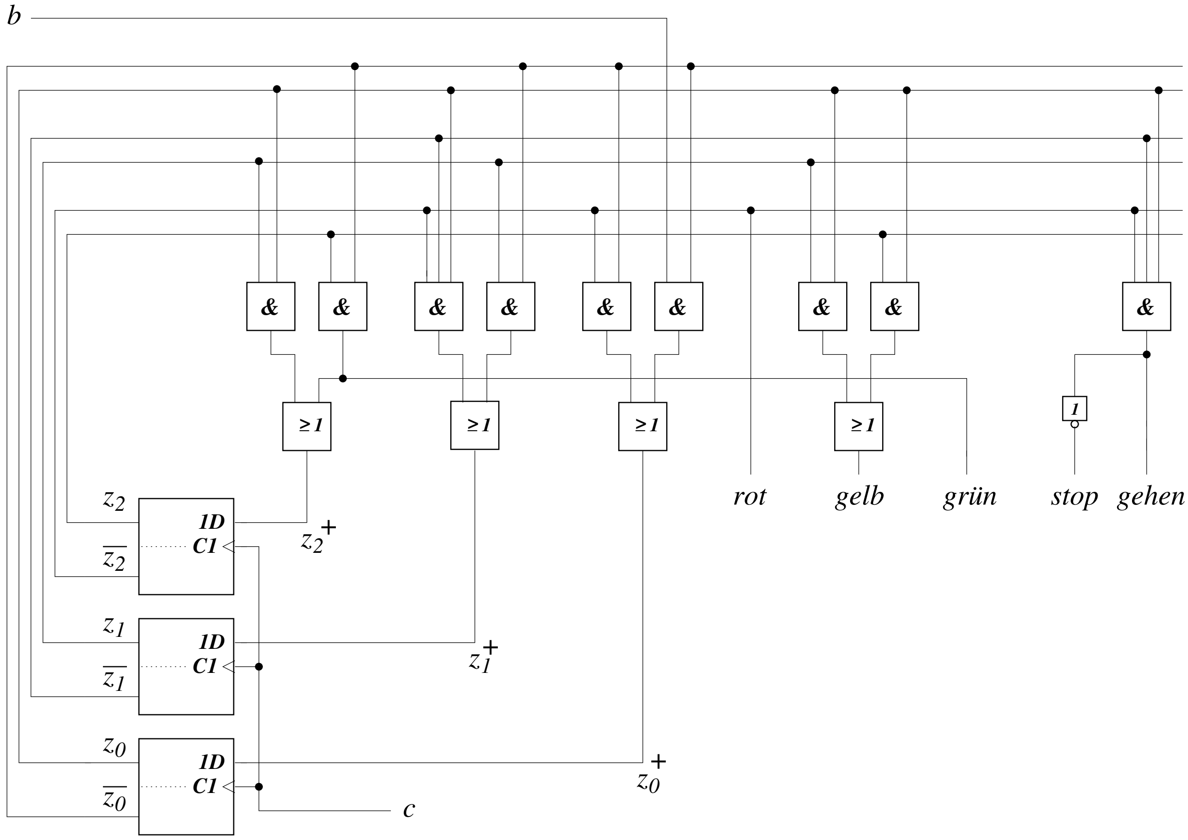


Aus der Minimierung ergeben sich schließlich für die Übergangs- und Ausgangsfunktionen:

$$
\begin{aligned}
z_2^+ &= z_1 z_0 \vee z_2 \overline{z_0} & \quad rot &= \overline{z_2} \\
z_1^+ &= \overline{z_2}\,\overline{z_1} z_0 \vee z_1 \overline{z_0} & \quad gelb &= z_1 z_0 \vee z_2 z_0 \\
z_0^+ &= \overline{z_2}\,\overline{z_0} \vee b \overline{z_0} & \quad grün &= z_2 \overline{z_0}
\end{aligned}
\qquad
\begin{aligned}
gehen &= \overline{z_2}\,\overline{z_1} z_0 \\
stop &= \overline{gehen}
\end{aligned}
$$

Da der Term $z_2\overline{z_0}$ zweifach auftritt, kann dieser zur Bündeloptimierung genutzt werden.

c) Es ergibt sich folgender Moore-Automat :



d) Gerät der Automat auf Grund elektromagnetischer Störungen in einen der beiden nicht kodierten Zustände $(110)_2$ oder $(111)_2$, so wird der Automat in einen Folgezustand übergehen, der durch das Übergangsschaltnetz bestimmt wird. Für den vorliegenden Automaten sind dies:

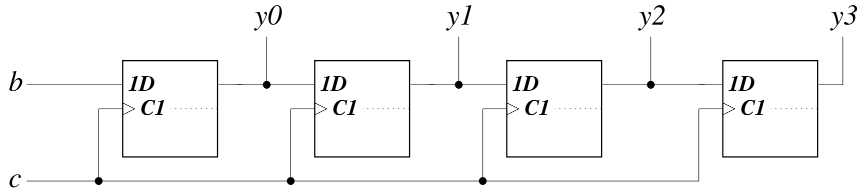
$$(110)_2 \rightarrow (110)_2 \text{ bzw. } (111)_2 \qquad \text{und} \qquad (111)_2 \rightarrow (100)_2$$

Wenn der Automat in den Zustand $(111)_2$ gerät, geht er beim nächsten Taktimpuls in den Zustand $(100)_2 \Leftrightarrow E$ über. Gerät der Automat jedoch in den Zustand $(110)_2$, so bleibt er bei $b = 0$ dort "*hängen*". Bei $b = 1$ wird in den ebenfalls undefinierten Zustand $(111)_2$ übergegangen. Das Aufhängen des Automaten könnte durch Auskodieren der nicht benötigten Zustände erfolgen, so dass diese auf einen definierten Zustand führen.

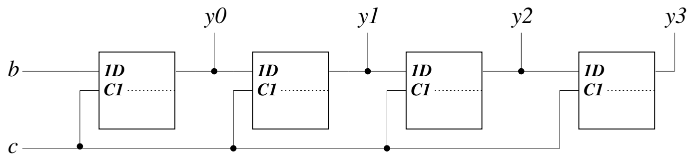
e) Der Anfangszustand des Automaten ist nicht definiert. Der Automat kann nach dem Einschalten einen beliebigen Zustand der 2^3 möglichen Zustände annehmen. Um den Automat in einen definierten Anfangszustand zu bringen, müssten D-Flipflops mit einem zusätzlichen Rücksetzeingang (asynchroner Reset) verwendet werden. Eine andere Möglichkeit wäre, die nicht kodierten Zustände $(110)_2$ und $(111)_2$ auf den Zustand E abzubilden, so dass der Automat – unabhängig davon, in welchem Zustand sich der Automat nach dem Einschalten befindet – in einen definierten Zustand übergeht.

Lösung 6.2

a) Vorderflankengesteuerte D-Flipflops übernehmen mit der Taktvorderflanke (unter Berücksichtigung der Signallaufzeit, d.h. einige Nanosekunden später) den am D-Eingang anliegenden Signalpegel und reichen diese zum Ausgang weiter. Wenn das "*neue*" Signal am Ausgang erscheint, muss der Übernahmevorgang schon abgeschlossen sein. Nur so ist es möglich, mit flankengesteuerten D-Flipflops ein Schieberegister in der folgenden Weise aufzubauen:

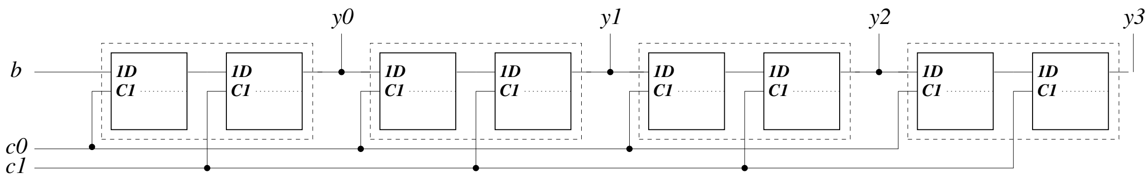


b) Positiv taktzustandsgesteuerte D-Flipflops reichen den am D-Eingang des ersten Kippglieds anliegenden Signalpegel solange zum Ausgang durch, wie der Takteingang logisch 1 ist. Versuchte man, mit derartigen Flipflops ein Schieberegister wie in Aufgabenteil a) zu realisieren, so würde dieses während des aktiven Taktpegels das Eingangssignal des ersten Flipflops bis zum letzten durchreichen, d.h. alle Ausgänge y_0, y_1, y_2, y_3 hätten den gleichen Wert. Eine derartige Schaltung hätte prinzipiell folgendes Aussehen:



Diese Schaltung erfüllt aber keine Schieberegisterfunktion!

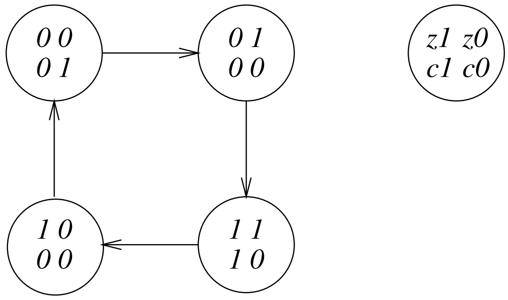
c) Werden pro "Schiebezelle" zwei zustandsgesteuerte D-Flipflops verwendet und mit einem nichtüberlappenden 2-Phasen-Takt angesteuert, so kann das in Aufgabenteil b) beschriebene Problem verhindert werden:



Die Struktur erinnert an JK-Master-Slave Flipflops. Es sei an dieser Stelle noch bemerkt, dass man derartige Schieberegister technologisch sehr effizient - mit wenigen Transistoren in Form von dynamischen[1] D-Flipflops - realisieren kann. Dynamische Schieberegister finden dort Anwendung, wo ein kontinuierlicher Datenstrom zu verarbeiten ist, wie z.B. in der Bildvorverarbeitung.

[1] vgl. DRAMs: die Speicherung erfolgt für eine begrenzte Zeit (einige Millisekunden) durch elektrische Ladungen. Diese müssen regelmäßig wieder aufgefrischt werden (Refresh), was bei dynamischen Schieberegistern automatisch durch ständiges Schieben mit hinreichend hoher Taktfrequenz erfolgt.

d) Aus der Aufgabenstellung ergibt sich für den zu konstruierenden Moore-Automaten folgender Graph:



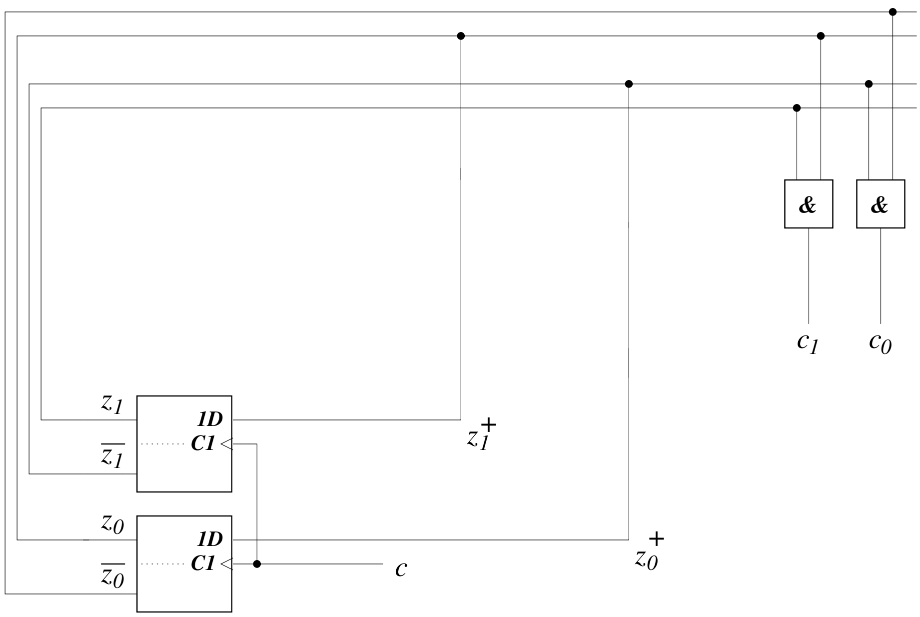
Aus dem dargestellten Graphen können die Übergangs- und Ausgangsfunktionen direkt abgelesen werden. Es ergibt sich daraus die folgende Wertetabelle:

z_1	z_0	z_1^+	z_0^+	c_1	c_0
0	0	0	1	0	1
0	1	1	1	0	0
1	1	1	0	1	0
1	0	0	0	0	0

Aus dieser Wertetabelle lassen sich schließlich die zu implementierenden Booleschen Funktionen direkt ablesen:

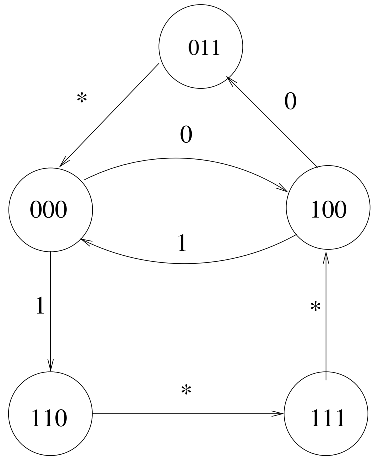
$$
\begin{aligned}
z_1^+ &= \overline{z_1}\, z_0 \vee z_1\, z_0 = z_0 & \qquad c_1 &= z_1\, z_0 \\
z_0^+ &= \overline{z_1}\,\overline{z_0} \vee \overline{z_1}\, z_0 = \overline{z_1} & \qquad c_0 &= \overline{z_1}\,\overline{z_0}
\end{aligned}
$$

Umgesetzt in einen Moore-Automaten erhält man:



e) Aufgrund der einschrittigen Zustandskodierung im Zusammenhang mit dem nur einstufigen Schaltnetz der Ausgangsfunktion sind die Ausgänge c_1 und c_0 frei von Hazards, da sich bei jedem Zustandsübergang nur ein Bit ändert und in dem einstufigen Schaltnetz keine Signallaufzeitdifferenzen auftreten, die an den Ausgängen c_1 und c_0 Hazards zur Folge haben könnten.

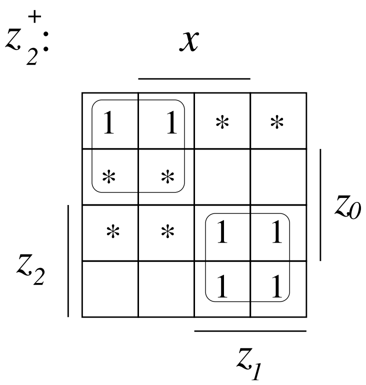
Lösung 6.3 Es ergibt sich folgendes Zustandsdiagramm:

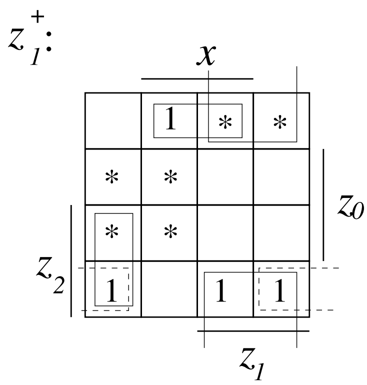


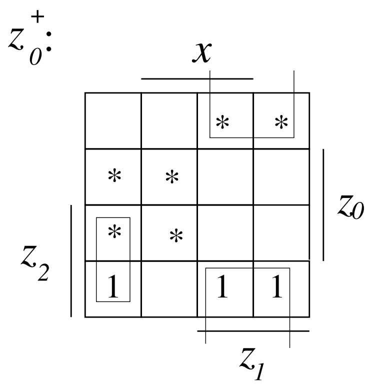
und die Übergangstabelle

x	z_2	z_1	z_0	z_2^+	z_1^+	z_0^+
0	0	0	0	1	0	0
0	0	0	1	*	*	*
0	0	1	0	*	*	*
0	0	1	1	0	0	0
0	1	0	0	0	1	1
0	1	0	1	*	*	*
0	1	1	0	1	1	1
0	1	1	1	1	0	0
1	0	0	0	1	1	0
1	0	0	1	*	*	*
1	0	1	0	*	*	*
1	0	1	1	0	0	0
1	1	0	0	0	0	0
1	1	0	1	*	*	*
1	1	1	0	1	1	1
1	1	1	1	1	0	0

Daraus ergeben sich folgende KV-Diagramme







und damit die möglichen Funktionen (die Schleifen in den KV-Diagrammen lassen sich natürlich auch anders wählen):

$$
\begin{aligned}
z_0 &= \overline{x}\,\overline{z_1}\,z_2 \vee \overline{z_0}\,z_1 \\
z_1 &= \overline{z_0}\,z_1 \vee \overline{x}\,\overline{z_1}\,z_2 \vee x\,\overline{z_0}\,\overline{z_2} \\
z_2 &= z_1\,z_2 \vee \overline{z_0}\,\overline{z_2}
\end{aligned}
$$

Mit diesen Funktionen erhält man dann für die Übergänge, die wir oben als Don't-Cares angenommen hatten:

x	z_2	z_1	z_0	z_2^+	z_1^+	z_0^+
0	0	0	0	1	0	0
0	0	0	1	0	0	0
0	0	1	0	1	1	1
0	0	1	1	0	0	0
0	1	0	0	0	1	1
0	1	0	1	0	1	1
0	1	1	0	1	1	1
0	1	1	1	1	0	0
1	0	0	0	1	1	0
1	0	0	1	0	0	0
1	0	1	0	1	1	1
1	0	1	1	0	0	0
1	1	0	0	0	0	0
1	1	0	1	0	0	0
1	1	1	0	1	1	1
1	1	1	1	1	0	0

Es zeigt sich, dass wir (eher zufälligerweise!) immer in einem vorgegebenen Zustand unseres Automaten landen, so dass es bei dieser Wahl des Übergangsschaltnetzes keine Probleme bzw. Designfehler gibt.
Wenn man in KV-Diagramm für z_1^+ die Schleife für die 1 in der linken unteren Ecke anders legt, ist dem nicht mehr so. Mit der neuen Funktion $z_1 = \overline{z_0}\,z_1 \vee \overline{x}\,\overline{z_0}\,z_2 \vee x\,\overline{z_0}\,\overline{z_2}$ erhält man nämlich folgende Zustandsübergänge:

x	z_2	z_1	z_0	z_2^+	z_1^+	z_0^+	
0	0	0	0	1	0	0	
0	0	0	1	0	0	0	
0	0	1	0	1	1	1	
0	0	1	1	0	0	0	
0	1	0	0	0	1	1	
0	1	0	1	0	0	1	$\leftarrow$
0	1	1	0	1	1	1	
0	1	1	1	1	0	0	
1	0	0	0	1	1	0	
1	0	0	1	0	0	0	
1	0	1	0	1	1	1	
1	0	1	1	0	0	0	
1	1	0	0	0	0	0	
1	1	0	1	0	0	0	
1	1	1	0	1	1	1	
1	1	1	1	1	0	0	

und damit in der markierten Zeile einen Fehler.