RHEINISCH-WESTFÄLISCHE TECHNISCHE HOCHSCHULE AACHEN

LEHRSTUHL INFORMATIK 11 SOFTWARE FÜR EINGEBETTETE SYSTEME

Professor Dr.-Ing. Stefan Kowalewski
Dipl.-Inform. Dominik Franke
Dipl.-Inform. Sebastian Biallas

Aachen, 07. Mai 2010
SWS: V4/Ü2, ECTS: 7

Einführung in die Technische Informatik

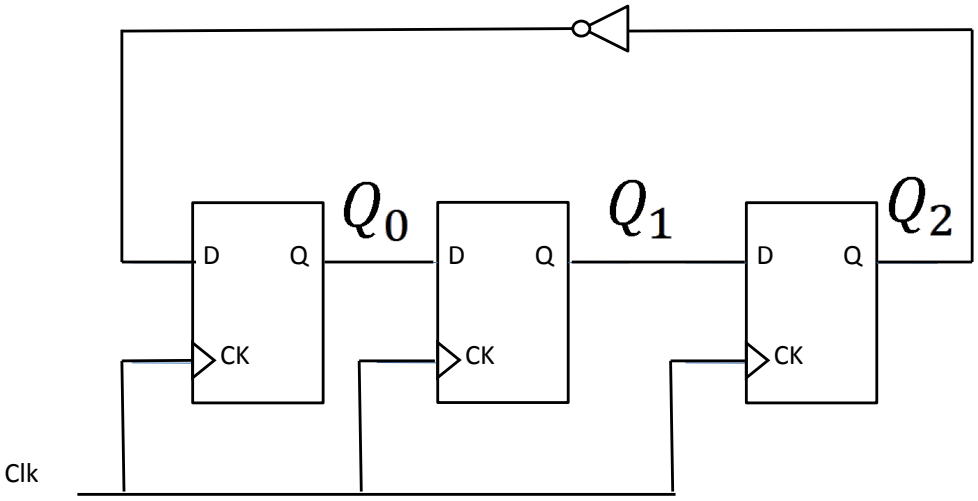
WS 2010/2011

Blatt 8: Musterlösung

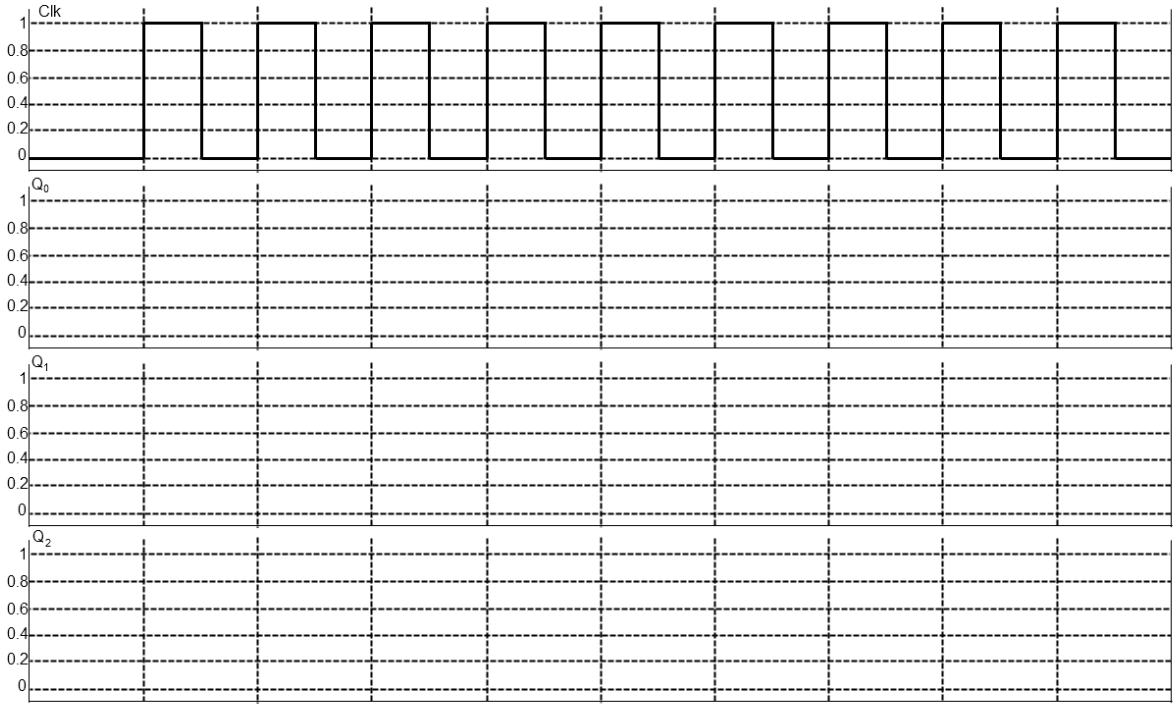
ACHTUNG: Die Musterlösung ist ein zusätzliches Serviceangebot. Sie erhebt weder Anspruch auf Vollständigkeit noch auf Korrektheit.

Aufgabe 1: (⋆) Signalverlauf

Gegeben sei die folgende Schaltung:



a) Vervollständigen Sie den folgenden Signalverlauf. Gehen Sie dabei davon aus, dass zu Beginn Q_0, Q_1 und Q_2 den Wert 0 haben.



b) Die Variablen Q_2', Q_1' und Q_0' bezeichnen die neuen Werte von Q_2, Q_1, Q_0 nach einem Takt.
Ergänzen Sie die folgende Tabelle:

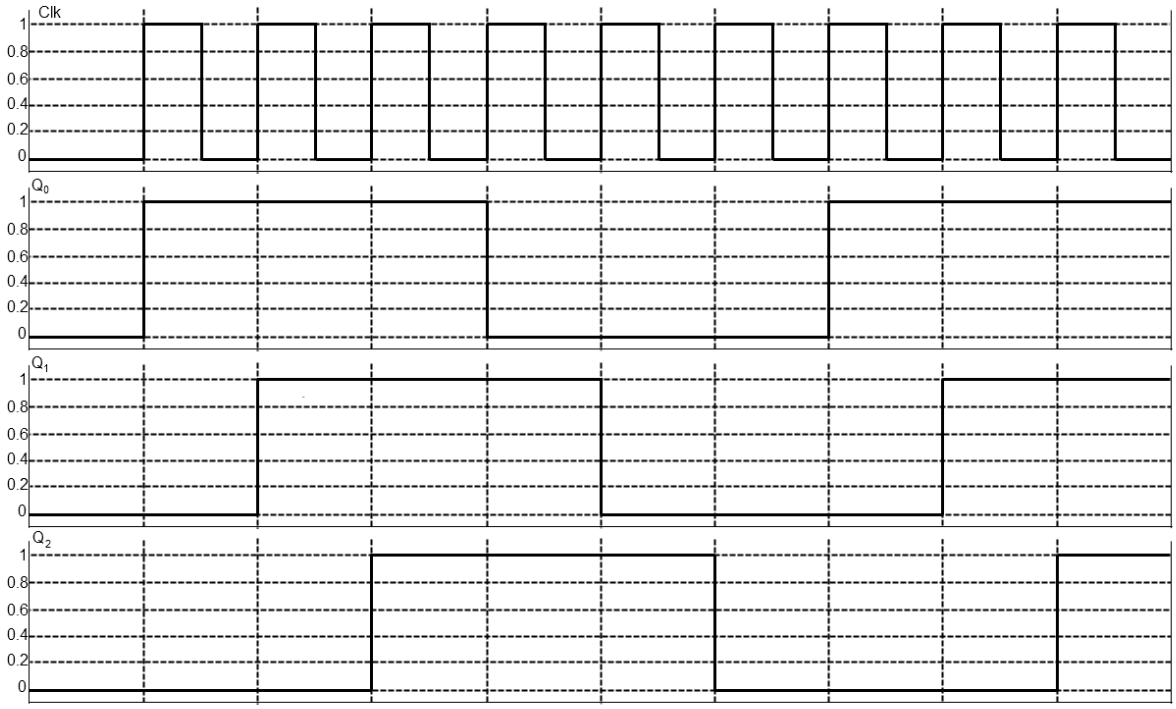
Aktuelle Zustände			Nächste Zustände		
Q_2	Q_1	Q_0	Q_2'	Q_1'	Q_0'
0	0	0			
0	0	1			
0	1	0			
0	1	1			
1	0	0			
1	0	1			
1	1	0			
1	1	1			

c) Mit jedem Takt werden die Belegungen von Q_2, Q_1 und Q_0 geändert. Abhängig von der Anfangsbelegung von Q_2, Q_1 und Q_0 sind verschiedene Verläufe möglich. Geben Sie alle möglichen Verläufe als Zustandsdiagramme an.

Hinweis: Gemeint sind Läufe der Form $Q_2Q_1Q_0 \rightarrow Q_2'Q_1'Q_0' \rightarrow Q_2''Q_1''Q_0'' \rightarrow \ldots$. Verwenden Sie hierzu die Ergebnisse der Teilaufgabe b).

Lösungsvorschlag

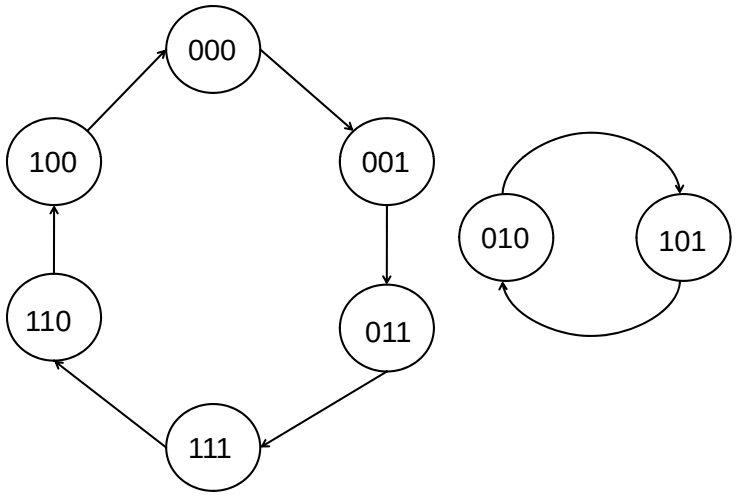
a)



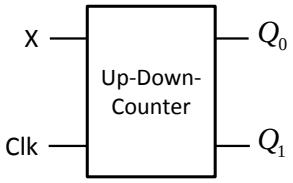
b)

Aktuelle Zustände			Nächste Zustände		
Q_2	Q_1	Q_0	Q'_2	Q'_1	Q'_0
0	0	0	0	0	1
0	0	1	0	1	1
0	1	0	1	0	1
0	1	1	1	1	1
1	0	0	0	0	0
1	0	1	0	1	0
1	1	0	1	0	0
1	1	1	1	1	0

c)



Aufgabe 2: ($\star$) **Flip-Flops**
In dieser Aufgabe sollen Sie einen 2-Bit Up-Down-Counter mit Hilfe von zwei D-Flip-Flops konstruieren. Das untere Bild zeigt den gesuchten Zähler als Black-Box (also ohne das Innenleben zu offenbaren).



a) Das Verhalten des Zählers kann mit einer Tabelle wie der folgenden angegeben werden. Dabei ist Q_1 das msb[1] und Q_0 das lsb der dargestellten Zahl. Die Ausgaben Q_1' und Q_0' sind die neuen Zustände von Q_1 und Q_0 nach einem Takt und sind abhängig von X, Q_1 und Q_0. Bei $X = 0$ soll der Zähler hoch- und bei $X = 1$ runterzählen.

Vervollständigen Sie die folgende Tabelle:

Aktuelle Zustände			Nächste Zustände	
X	Q_1	Q_0	Q_1'	Q_0'
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

[1]Most Significant Bit und Least Significant Bit.

b) Finden Sie das Minimalpolynom zu Q_1' und Q_0'.
Hinweis: Zur Vereinfachung können Sie Karnaugh-Diagramme benutzen.

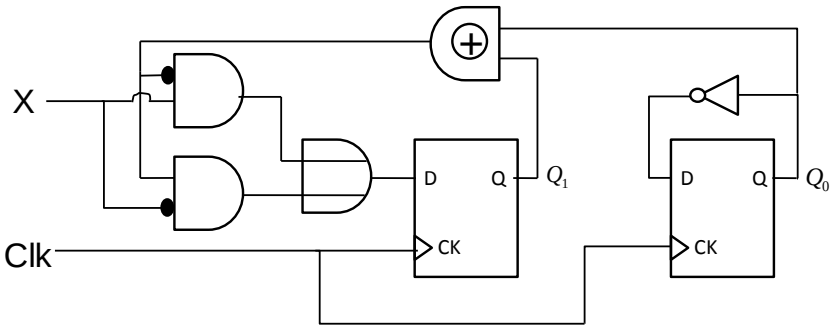
c) Realisieren Sie den beschriebenen Zähler mit Hilfe von zwei D-Flip-Flops und den nötigen Gattern (zur Verfügung stehen beliebig viele: AND, OR, NOT, XOR, NAND).

Lösungsvorschlag

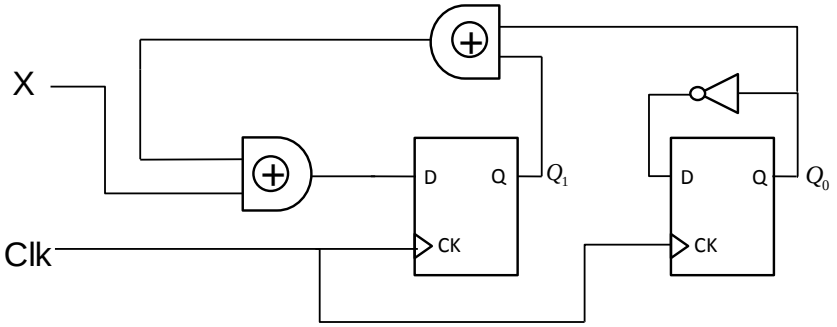
a) (In Globalübung vorgerechnet.)

b) (In Globalübung vorgerechnet.)

c) Die ersten zwei Terme in der Gleichung zu Q_1' lassen sich zu $X\overline{(Q_1 \oplus Q_0)}$ umformen. Die letzten beiden Terme in der Gleichung lassen sich zu $\overline{X}(Q_1 \oplus Q_0)$ umformen. Insgesamt ergibt sich als mögliche Realisierung:

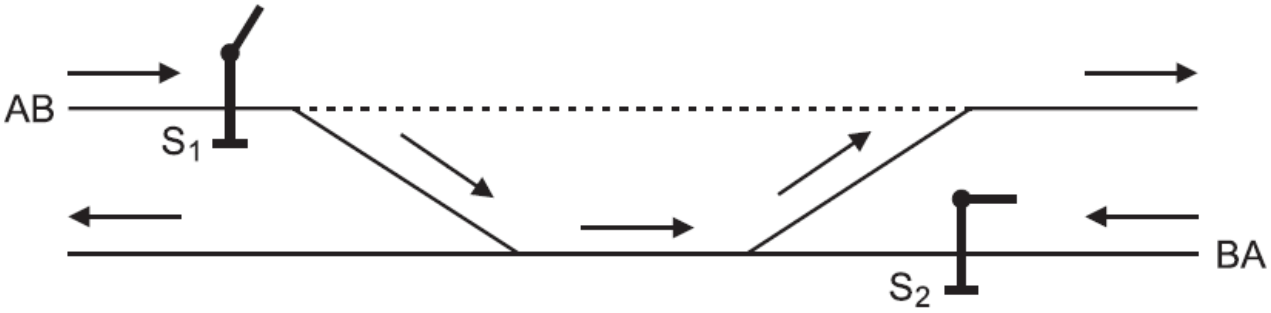


Diese Schaltung lässt sich mit einem weiteren XOR-Gatter umformen zu:



Aufgabe 3: Schaltungsentwurf

Zwischen zwei Orten A und B gibt es eine Schienenverbindung mit zwei Gleiswegen AB und BA. Dabei nutzen von A nach B fahrende Züge den Gleisweg AB, während von B kommende Züge den Gleisweg BA befahren. Bauarbeiten auf einem Teilabschnitt von AB führen jedoch zu einer Ausnahmesituation, die in der folgenden Abbildung illustriert ist:



Danach müssen AB befahrende Züge für einen kurzen Teilabschnitt auf den Gleisweg BA ausweichen. Der Zugverkehr auf diesem Teilabschnitt soll durch zwei einfache Signalanlagen S1 und S2 geregelt werden, die entweder 'freie Fahrt' oder 'unbedingter Halt' signalisieren können. Weiterhin sollte die Regelung derart erfolgen, dass die beiden Signalanlagen in ihren Signalen als Paar (S_1, S_2) aufgefasst stets in einem Zyklus die Zustände $(0,0), (1,0), (0,0), (0,1)$ und wieder $(0,0)$ durchlaufen, wobei 'freie Fahrt' mit 1 und 'unbedingter Halt' mit 0 codiert ist.

a) Entwerfen Sie ein (Steuer-)Schaltwerk, das die beiden Signalanlagen in der beschriebenen Weise und unter Verwendung der angegebenen Codierung steuert.

b) Demonstrieren Sie die Arbeitsweise Ihres Steuerwerks, indem Sie die Inhalte aller vorhandenen Delays vor und nach jedem Takt für einen kompletten Zyklus protokollieren.

c) Erweitern Sie Ihr Steuerwerk um eine Steuerleitung Z, deren Wert darüber bestimmt, ob der regelnde Betrieb stattfindet oder ob die beiden Signalanlagen ständig auf 'unbedingter Halt' gesetzt sind.

Hinweis: Teil c) lässt sich mit zwei zusätzlichen Gattern realisieren.

Lösungsvorschlag

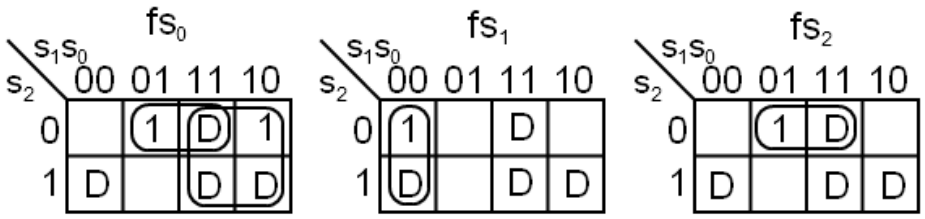
a) Die Schwierigkeit der Aufgabe besteht darin, dass ein aktuelles Signalpaar (0, 0) in Abhängigkeit von der vorherigen Signalstellung sowohl in das Signalpaar (1, 0) als auch in das Signalpaar (0, 1) überführt werden kann bzw. muss.

Man könnte 'intern' z. B. das zweite (0, 0)-Paar durch (1, 1) ersetzen und mit einer Art Ringzähler arbeiten, der in einem Zyklus $(0,0) \rightarrow (1,0) \rightarrow (1,1) \rightarrow (0,1) \rightarrow (0,0) \rightarrow ...$ 'zählt'. Eine nachgeschaltete 'Umwandlungslogik' muss dann das Paar (1, 1) in (0, 0) umrechnen, während die anderen Zählerausgaben durchgeleitet werden. Ein Kritikpunkt an dieser Lösung im Hinblick auf die Aufgabenstellung ist, dass sie die beiden Signalanlagen nicht direkt in der beschriebenen Weise und unter Verwendung der angegebenen Codierung steuert, sondern die Zustandscodierung von den Steuersignalen trennt. Dass hier drei der vier Zustandscodierungen mit den Steuersignalen übereinstimmen können, ändert nichts an der hier prinzipiell anderen Vorgehensweise. Im Allgemeinen muss dann auch bedacht werden, dass die 'Umwandlungslogik' zuverlässig sein muss, um sicheren Signalbetrieb zu gewährleisten. (In der Vorlesung sind Hazards nicht besprochen worden.)

Wir nutzen zur Lösung eine Schaltfunktion $f : B^3 \rightarrow B^3$, die über ein zusätzliches Bit S_0 beide (0, 0)-Varianten unterscheiden kann. Dabei wird f nur partiell genutzt. Das zusätzliche Bit entspricht dann einem 'internen' Steuersignal S_0. Wir notieren S_0^{alt}, S_1^{alt} und S_2^{alt} als Signalwerte vor einem Signalwechsel und S_0^{neu} , S_1^{neu} , S_2^{neu} als entsprechende Signalwerte danach. Den Signalübergang von z. B. S_0^{alt} nach S_0^{neu} beschreibt die Funktion fs_0 (fs_1 und fs_2 entsprechend). Damit ergibt sich die folgende Wertetabelle, in der wir die ausgezeichneten Signalpaare (S_1, S_2) des in der Aufgabenstellung gegebenen Zyklus mit Z_0, Z_1, Z_2, Z_3 bezeichnen. Die Funktionswerte S_1^{neu} und S_2^{neu} sind durch den einzuhaltenden Zyklus vorgegeben. Allerdings sind zur Festlegung von S_0^{neu} teilweise auch Alternativen möglich, d. h. es gibt mehrere Lösungen, die bis auf 'symmetrische' Vertauschungen identisch sind.

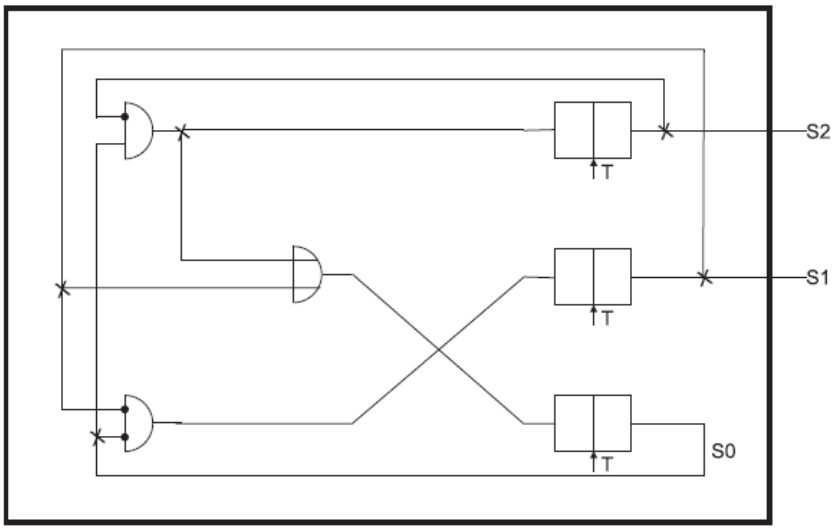
'altes' Signalpaar	S_0^{alt}	S_1^{alt}	S_2^{alt}	S_0^{neu}	S_1^{neu}	S_2^{neu}	'neues' Signalpaar
Z_0	0	0	0	0	1	0	Z_1
tritt niemals auf	0	0	1	D	D	D	
Z_1	0	1	0	1	0	0	Z_2
tritt niemals auf	0	1	1	D	D	D	
Z_2	1	0	0	1	0	1	Z_3
Z_3	1	0	1	0	0	0	Z_0
tritt niemals auf	1	1	0	D	D	D	
tritt niemals auf	1	1	1	D	D	D	

Mit Hilfe von Karnaugh-Diagrammen und unter Ausnutzung der Don't-Care-Werte ergeben sich für die drei Booleschen Funktionen S_0^{neu}, S_1^{neu}, S_2^{neu} der gesuchten Schaltfunktion f die folgenden Minimalpolynome:



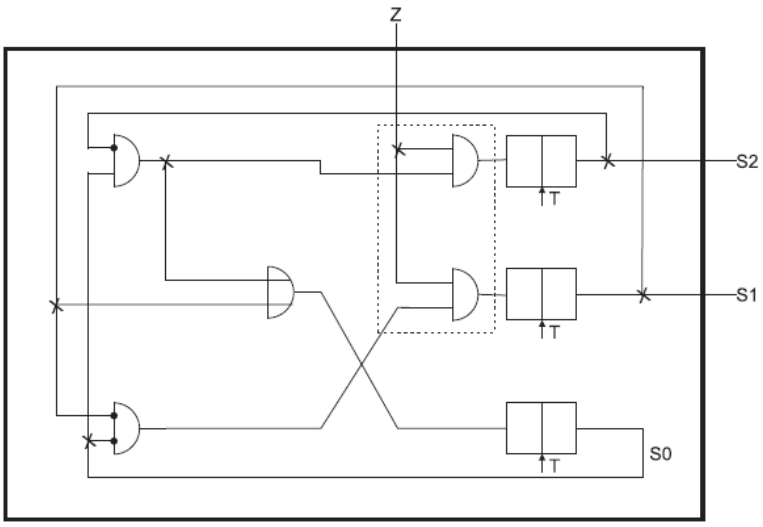
Daraus folgt: $S_0^{neu} = S_0^{alt} \cdot \overline{S_2^{alt}} + S_1^{alt} \quad S_1^{neu} = \overline{S_0^{alt}} \cdot \overline{S_1^{alt}} \quad S_2^{neu} = S_0^{alt} \cdot \overline{S_2^{alt}}$

Damit ergibt sich das gesuchte Schaltwerk unter Verwendung von drei 1-Bit-Registern wie in der folgenden Abbildung:



b) Wir notieren $|S_0|S_1S_2|$ und starten initial mit $S_0 = 0$ sowie $S_1 = S_2 = 0$, d.h. beide Signalanlagen signalisieren 'unbedingter Halt'.
$|0|00| \xrightarrow{Takt} |0|10| \xrightarrow{Takt} |1|00| \xrightarrow{Takt} |1|01| \xrightarrow{Takt} |0|00| \xrightarrow{Takt} ...$

c) Erweitertes Steuerwerk: Die nachfolgende Abbildung zeigt eine mögliche Realisierung.



Eine andere Möglichkeit wäre, die ausgehenden Steuersignale S_2S_1 durch ein UND mit Z zu verbinden.

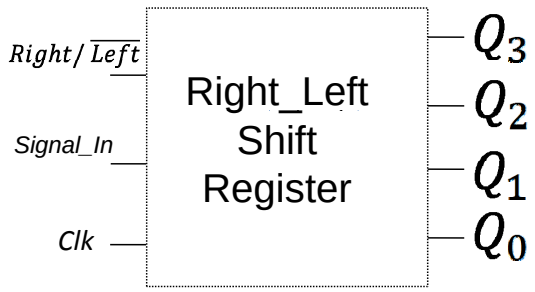
Noch eine andere Möglichkeit, für die sogar nur ein einziges UND-Gatter nötig ist, wäre, den an den Flip-Flops eingehenden Takt T durch ein UND mit Z zu verbinden. Der Vorteil hierbei ist, dass bei Wiederaufnahme des Signalzyklus dort weitergemacht werden kann, wo aufgehört wurde. Dies ist zwar auch bei der skizzierten Schaltung der Fall, aber es ist nicht direkt einsehbar.

Die beiden genannten Alternativen haben den Vorteil, dass sie auch dann realisierbar sind, wenn das Bauteil aus Aufgabenteil a) als Black-Box vorliegt. Man muss nur den Takteingang bzw. die Ausgänge manipulieren. Bei der skizzierten Schaltung muss man dagegen das Innenleben modifizieren.

Aufgabe 4: Schieberegister

Ziel dieser Aufgabe ist es, ein Schieberegister für 4-Bit Zahlen zu realisieren. Desweiteren soll die Möglichkeit gegeben werden, die Richtung des Shifts vorzugeben, wie auch das nachrückende Bit zu spezifizieren.

Im Folgenden ist das zu realisierende Schaltwerk als Black-Box dargestellt:



a) Realisieren Sie das Schaltwerk. Sie benötigen dazu vier D-Flip-Flops, die die Werte von $Q_3Q_2Q_1Q_0$ speichern. Kümmern Sie sich zunächst nicht um die Initialisierung dieser Werte, nehmen Sie sie als gegeben an.

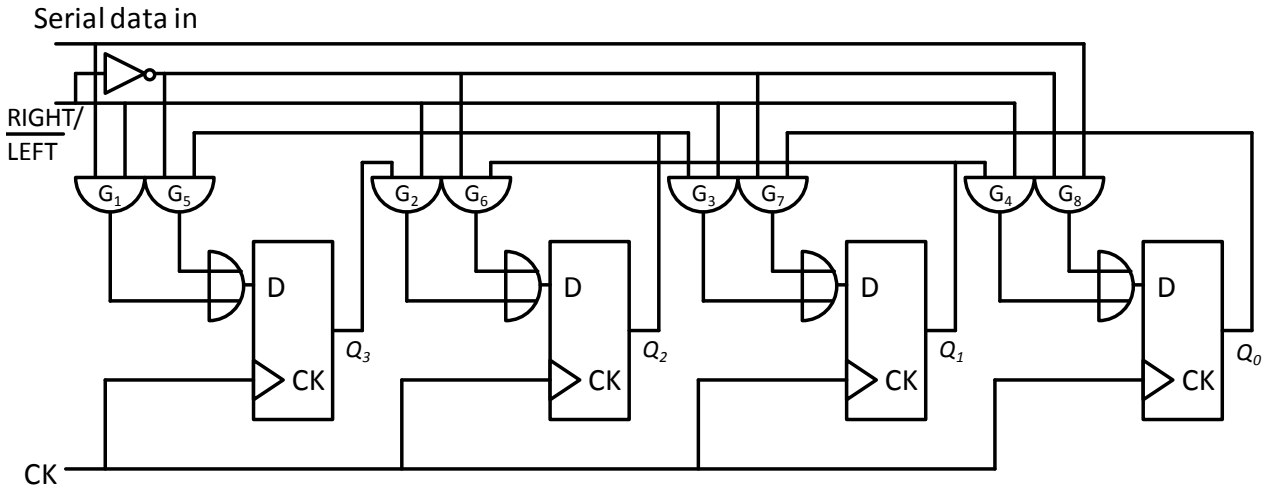
   Ihr Schaltwerk muss im Wesentlichen folgendes leisten: Bei einem Taktsignal wird im Falle von $Right/\overline{Left} = 1$ nach rechts geshiftet und bei $Right/\overline{Left} = 0$ nach links geshiftet. Dabei wird $Signal\_In$ auf die freiwerdende Stelle nachgeschoben. Ein Beispiel für $Right/\overline{Left} = 1$ und $Signal\_In = 1$ wäre $0000 \rightarrow 1000$. Ein Beispiel für $Right/\overline{Left} = 0$ und $Signal\_In = 1$ wäre $0000 \rightarrow 0001$. Ein anderes Beispiel für $Right/\overline{Left} = 0$ und $Signal\_In = 0$ wäre $0010 \rightarrow 0100$.

b) Wie können Sie mit Ihrem Schaltwerk das Register auf einen vorgegeben Wert initialisieren?

c) Wie müssen Sie Ihr Schaltwerk erweitern/modifizieren, wenn Sie noch einen zusätzlichen Eingang $Mode$ berücksichtigen sollen, der festlegt, ob $Signal\_In$ benutzt wird, oder ob 'im Kreis' geshiftet wird? Dabei legt $Right/\overline{Left}$ weiterhin die Richtung des Shiftens fest.

Lösungsvorschlag

a) Nachfolgend ist eine mögliche Realisierung abgebildet. Bei der Abhandlung der einzelnen Flip-Flops sind im Wesentlichen zwei Fälle zu betrachten. Entweder das Flip-Flop befindet sich 'im Inneren', wie Q_2 und Q_1 oder es ist 'am Rand', wie Q_3 und Q_0.



Exemplarisch werden nun die Schaltungen vor Q_2 und vor Q_3 erläutert.

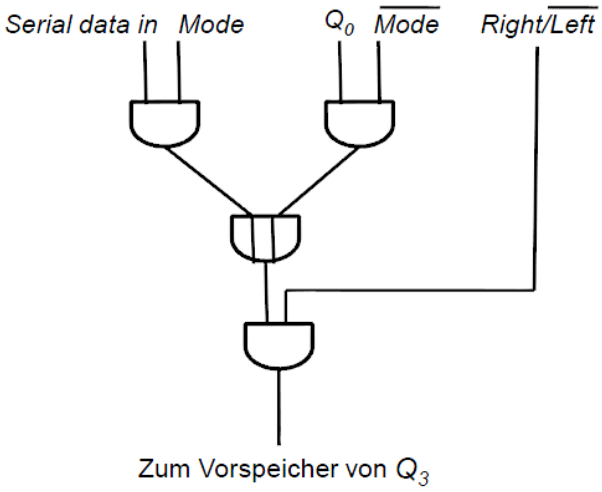
Bei Q_2 wird im Falle eines Linksshift der Wert von Q_1 übernommen. Dazu muss also $Right/\overline{Left} = 0$ sein. Diese Bedingung ist mit dem UND-Gatter G_6 realisiert. Im Falle eines Rechtsshift wird der Wert von Q_3 nach Q_2 übernommen. Dieser Fall ist mit dem Gatter G_2 abgedeckt.

Bei Q_3 ist der Linksshift mit dem Gatter G_5 realisiert. Der Rechtsshift ist mit dem Gatter G_1 realisiert, wobei hier anzumerken ist, dass der $SerialIn$ in Q_3 eingelesen wird.

b) Mithilfe des $SerialIn$ kann das Register auf jeden beliebigen Wert initialisiert werden. Dazu sind allerdings genau 4 Takte notwendig. Zunächst wird die Shiftrichtung festgelegt und für den restlichen Initialisierungsvorgang beibehalten. Beide Shiftrichtungen sind möglich. Wir legen uns exemplarisch auf den Rechtsshift fest. Soll $Q_3Q_2Q_1Q_0$ z.B. auf 1011 initialisiert werden so muss zunächst das am weitesten rechts stehende Bit, also eine 1, an den $SerialIn$ angelegt werden. Dann wird ein Taktsignal ausgeführt und diese 1 wandert nach Q_3. Dann werden der Reihe nach die nächsten Bits, also 1,0,1 an $SerialIn$ gelegt und dazwischen je ein Taktsignal gegeben. Am Ende steht die gewünschte Zahl im Register.

c) Um das 'im Kreis shiften' zu ermöglichen, muss das Schaltwerk so angepasst werden, dass je nach dem Wert von $Mode$ die äußeren Registerbits entweder den $SerialIn$ oder aber das gegenüberliegende Registerbit berücksichtigen.

Wir erklären hier die nötigen Modifikationen für Q_3, also das am weitesten links stehende Registerbit. Das Vorgehen für Q_0 ist analog. Für Q_3 muss nur das Verhalten im Falle eines Rechtsshift modifiziert werden, denn beim Linksshift wird wie bisher der Wert von Q_2 übernommen. Der Rechtsshift war, wie oben erklärt, mit dem Gatter G_1 realisiert. Dieses Gatter muss nun ersetzt werden durch das folgende Netz:



Weiterhin gilt, dass $Right/\overline{Left} = 1$ die oberste Bedingung dafür ist, dass eine 1 ausgegeben wird. Dann hängt es von $Mode$ ab, ob $SerialIn$ oder Q_0 durchgeschaltet wird.

Aufgabe 5: ($\star$) **Addierwerke**

a) Demonstrieren Sie die Arbeitsweise eines Parallel-Addierwerks, eines Serien-Addierwerks und eines von-Neumann-Addierwerks (jeweils 4-Bit-Addierwerke) für die nacheinander ausgeführten Berechnungen $4+10$, $13+5$ und $10+10$, indem Sie die Inhalte vorhandener 1-Bit Register schrittweise in einer Tabelle protokollieren.[2] Nehmen Sie hierbei an, dass das Übertragungs-Bit U jedes Addierwerks mit 0 initialisiert ist.

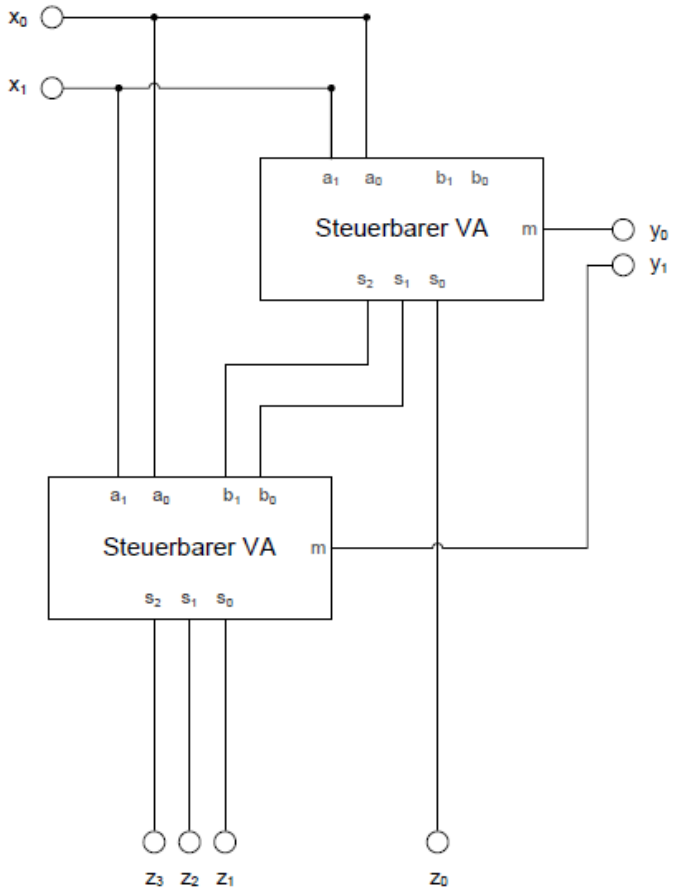
S	U	$P_3P_2P_1P_0$	dezimal	$A_3A_2A_1A_0$	dezimal(inkl. U)
0	0	0000	0	0000	0
1	0	0100	4	1010	10
...	...	....	...	...	...

b) Nennen Sie für jedes der drei Addierwerke möglichst viele Vor- und Nachteile.

Aufgabe 6: Digitale Schaltungen

Betrachten Sie die folgende Schaltung:

[2]Das Status-Bit S entfällt bei Parallel- und Serien-Addierwerk.



Die beiden steuerbaren Volladdierer besitzen zwei Eingänge für zwei zweistellige binäre Zahlen (a_1, a_0) und (b_1, b_0), die entsprechend der Vorlesung addiert werden, falls der Steuereingang $m = 1$ ist. Eingang (a_1, a_0) wird abgeschaltet (d.h. $a_1 = a_0 = 0$), sobald der Steuereingang $m = 0$ ist. Das Ergebnis wird jeweils in (s_2, s_1, s_0) geschrieben.

a) Stellen Sie die Wertetabelle für die Schaltfunktion $F((x_1, x_0), (y_1, y_0)) = (z_3, z_2, z_1, z_0)$ auf.

b) Welche Funktion wird durch F realisiert?

c) Formulieren Sie mit eigenen Worten die Funktionsweise der Schaltung.

Lösungsvorschlag

a) Die Wertetabelle für die Funktion lautet:

x_1	x_0	y_1	y_0	z_3	z_2	z_1	z_0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	0	1	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	0	0	0	0	0
0	1	0	1	0	0	0	1
0	1	1	0	0	0	1	0
0	1	1	1	0	0	1	1
1	0	0	0	0	0	0	0
1	0	0	1	0	0	1	0
1	0	1	0	0	1	0	0
1	0	1	1	0	1	1	0
1	1	0	0	0	0	0	0
1	1	0	1	0	0	1	1
1	1	1	0	0	1	1	0
1	1	1	1	1	0	0	1

b) Die Funktion F realisiert ein Multiplizierwerk.

c) Multiplikationen können leicht auf Additionen zurückgeführt werden. Somit lässt sich eine Multiplikation X $\cdot$ Y leicht mit Hilfe eines Volladdierers realisieren, indem X Mal Y aufsummiert wird. Für aufwändige Multiplikationen ist dieses Verfahren jedoch zu langsam. Die hier vorgestellte Schaltung bedient sich dem aus der Schule bekannten Schiebekonzept. Analog zum Dezimalsystem werden im Binärsystem die einzelnen Ziffern betrachtet und je nach Wertigkeit geschoben. Will man z.B. $13_{(10)} \cdot 5_{(10)}$ berechnen, würde man erst $3_{(10)} \cdot 5_{(10)} = 15_{(10)}$ und $1_{(10)} \cdot 5_{(10)} = 5_{(10)}$ berechnen, den Zehner um 1 nach links schieben und das Ergebnis anschließend zu $15_{(10)} + 50_{(10)} = 65_{(10)}$ addieren. In der Binärdarstellung kann pro Bit nur mit dem Faktor 0 oder 1 multipliziert werden, was einer Addition oder keiner Addition entspricht. Somit werden die Multiplikationen auch vollständig in Additionen überführt. Mit diesem Schema kann parallel multipliziert und dementsprechend bessere Performanz erzielt werden.