Universität Hamburg, Fachbereich Informatik

Arbeitsbereich Technische Aspekte Multimodaler Systeme

Praktikum der Technischen Informatik

T1 – 3

Getaktete Schaltwerke

Name: ....

Bogen erfolgreich bearbeitet: ....

Getaktete Schaltwerke

Zur Realisierung getakteter Schaltwerke benötigt man Zeitglieder, mit denen man zu definierten Zeitpunkten von einem Zustand in den Folgezustand umschalten kann. Dieses erreicht man praktisch nur mit Einschränkungen:

a. Die Taktzeitpunkte dürfen nicht beliebig schnell aufeinander folgen, da die Voreinstell- und Haltezeiten t_S, t_H der Zeitglieder eingehalten werden müssen.

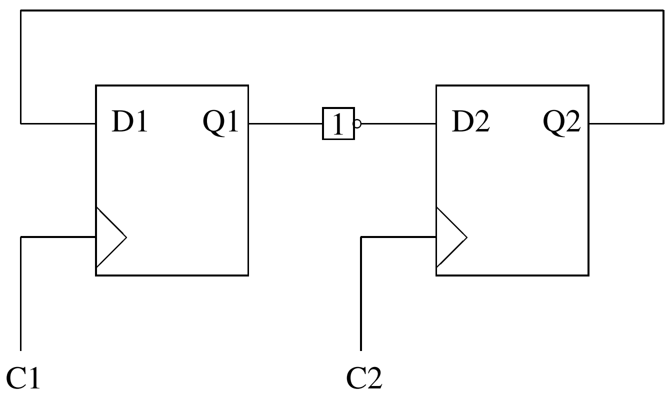
b. Bei abgeflachtern Flanken des Taktimpulses schalten verschiedene Zeitglieder desselben inneren Aufbaus aufgrund von Exemplarstreuungen zu leicht differierenden Zeiten.

c. Bei räumlich getrennten Zeitgliedern können erhebliche Laufzeitdifferenzen des Taktes hinzukommen.

Prinzipiell sind einflanken-gesteuerte Flipflops als Zeitglieder in getakteten Schaltwerken geeignet. Jedoch können die unter a.-c. genannten Effekte zu Problemen führen.

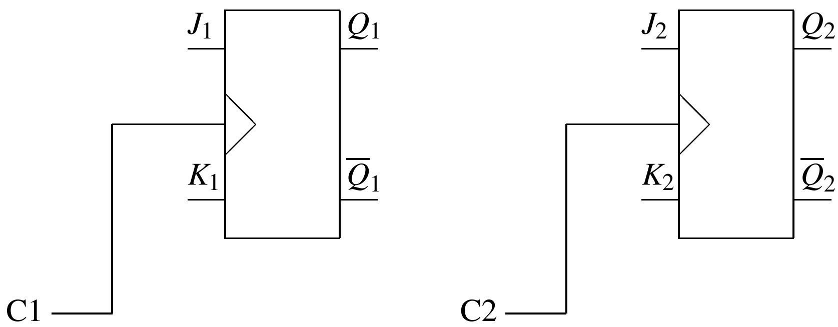
<u>Versuch 3.1:</u> Einfaches Schaltwerk mit rückflankengesteuerten D-Flipflops

Wir untersuchen folgendes einfache Schaltwerk:



Stellen Sie das Zustandsdiagramm auf. Beachten Sie dabei, dass Zustandsänderungen nur durch Rückflanken verursacht werden. Eine **0** an einem Pfeil bedeutet somit z.B. “keine Rückflanke”, eine **1** “Rückflanke am Takteingang”.

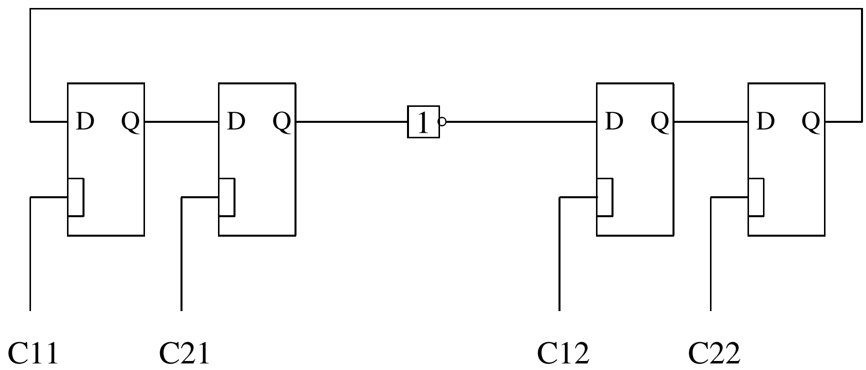
Vervollständigen Sie folgendes Schaltbild und simulieren Sie dann die Schaltung mit HADES, indem Sie Taktimpulse generieren. Beachten Sie dabei, dass auch der Eingang *NR* (von nicht-Reset), der dem $\overline{CLR}$ aus Versuch 2.6 entspricht, beschaltet werden muss. Beobachten Sie, ob die gewünschte Sequenz von Zuständen erreicht wird.



Aufgabe gelöst: Gruppe:	TeilnehmerIn: .... vom Betreuer auszufüllen	

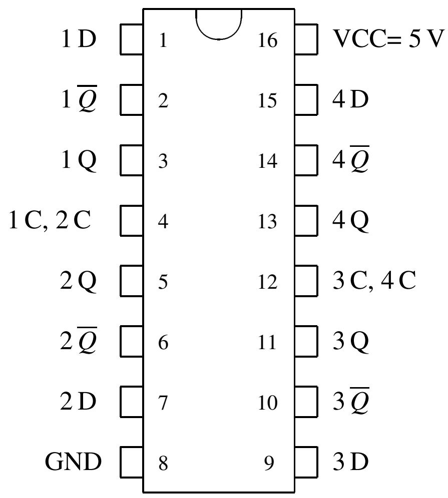
Versuch 3.2 Zweitaktschaltwerk mit pegelgesteuerten Flipflops

Jedes Schaltwerk aus **flankengesteuerten** D-Flipflops (also auch das aus Versuch 3.1) lässt sich auch mit **pegelgesteuerten** Flipflops realisieren, indem man einfach jedes flankengesteuerte Flipflop durch zwei pegelgesteuerte Flipflops ersetzt und diese mit zeitlich versetzten Takten versorgt. Die Schaltung aus Versuch 3.1 sieht dann also wie folgt aus:

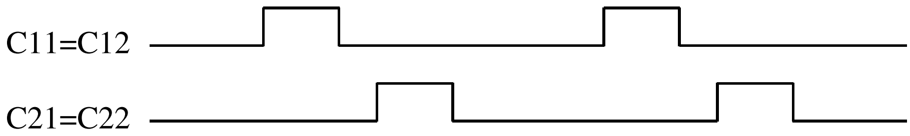


Bauen Sie das Schaltwerk mit dem pegelgesteuerten Flipflop SN 74LS375, das folgende Pinbelegung hat, auf oder simulieren Sie sie mit HADES (in diesem Fall nehmen Sie bitte für die vier Flipflops Bauteile von Typ **LatchR** aus dem Menue **create/flipflops**. Dies sind pegelgesteuerte Flipflops mit einem zusätzlichen low-aktiven Reset-Eingang, der dann natürlich auch belegt werden muss).

Eingänge		Ausgänge	
D	C	Q	$\overline{Q}$
L	H	L	H
H	H	H	L
$*$	L	Q_0	$\overline{Q}_0$



Prüfen Sie die Funktion des Schaltwerkes mit

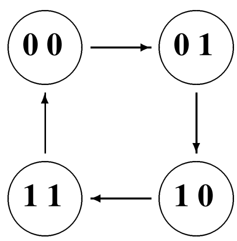


Aufgabe gelöst: Gruppe: .... TeilnehmerIn: ....

vom Betreuer auszufüllen

<u>Versuch 3.3:</u> Entwurf eines Modulo-4-Zählers (1)

Wir wollen nun selbst ein einfaches synchrones Schaltwerk mit den J-K-Flipflops aus Versuch 2.6 bzw. 3.1 entwerfen. Synchron bedeutet dabei, dass es einen zentralen Takt gibt, mit dem alle Flipflops gleichzeitig angesteuert werden. Als Beispiel wählen wir einen Modulo-4-Zähler, da sich dieser noch mit erträglichem Verkabelungsaufwand realisieren lässt. Zu entwerfen ist also ein Schaltwerk, das folgende Zustandsfolge durchläuft:



Offenbar benötigen wir für unseren Zähler zwei Flipflops, deren Eingänge J und K geeignet beschaltet werden müssen. Zunächst wollen wir uns klar machen, durch welche Belegung von J und K man bei einem J-K-Flipflop von einem Zustand Q^n zum Zustand Q^{n+1} kommt. Tragen Sie J und K in die folgende Tabelle ein. Beachten Sie dabei auch die Möglichkeit von Don't-Cares:

Q^n	Q^{n+1}	J	K
0	0		
0	1		
1	0		
1	1		

Damit können wir jetzt die Belegung für unseren Zähler notieren:

Q_1^n	Q_2^n	Q_1^{n+1}	Q_2^{n+1}	J_1	K_1	J_2	K_2
0	0						
0	1						
1	0						
1	1						

Formulieren Sie J_1, K_1, J_2, K_2 als logische Funktion von Q_1^n und Q_2^n:

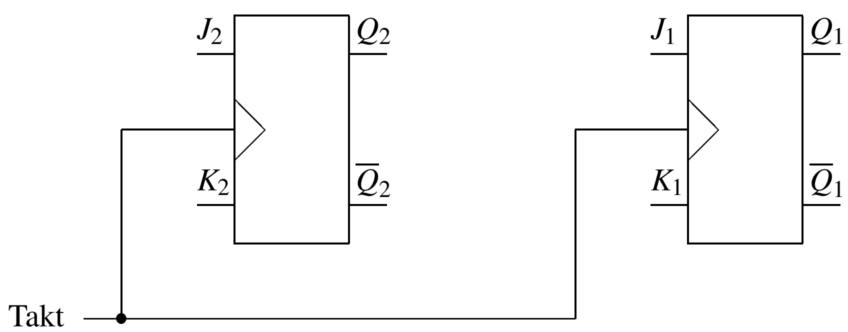
$J_1=$

$K_1=$

$J_2=$

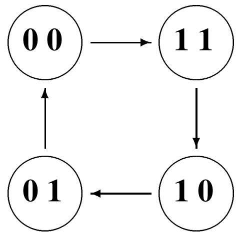
$K_2=$

Vervollständigen Sie den Schaltplan:



Bauen Sie nun die so entworfene Schaltung auf und testen Sie sie.

Wir haben bis jetzt einen Zähler entworfen, der in der üblichen binären Codierung "hochzählt". Genauso lässt sich ein Zähler entwerfen, der "herunterzählt", d.h. folgende Zustandsfolge durchläuft:



Entwerfen Sie wie oben einen solchen Zähler:

Q_1^n	Q_2^n	Q_1^{n+1}	Q_2^{n+1}	J_1	K_1	J_2	K_2
0	0						
0	1						
1	0						
1	1						

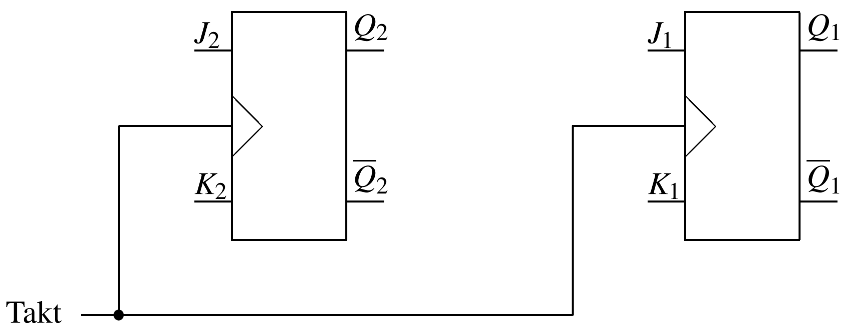
$J_1=$

$K_1=$

$J_2=$

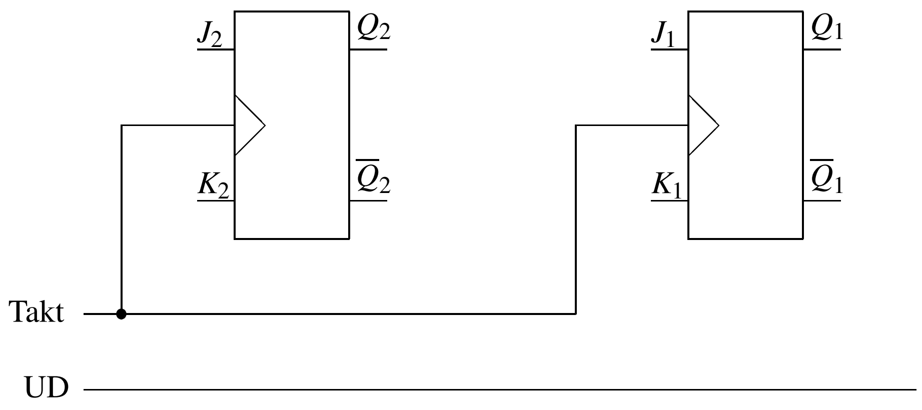
$K_2=$

Vervollständigen Sie den Schaltplan:



Zum Schluss wollen wir nun eine Schaltung entwerfen, die abhängig von einem Kontrollsignal UD "hoch" oder "herunter" zählen kann. Wir wollen dabei fordern: UD= **0** hochzählen, UD= **1** herunterzählen.

Entwerfen Sie die Schaltung und testen Sie sie aus. Es dürfen dabei alle bisher bekannten logischen Funktionen verwendet werden.



Aufgabe gelöst: Gruppe: .... TeilnehmerIn: ....
vom Betreuer auszufüllen

Versuch 3.4: Entwurf eines Modulo-4-Zählers (2)

In den Versuchen 3.1 und 3.2 hatten wir gesehen, dass sich ein Schaltwerk aus flankengesteuerten Flipflops auch mit pegelgesteuerten Flipflops aufbauen lässt. Der Entwurf mit pegelgesteuerten D-Flipflops unterscheidet sich dabei nicht wesentlich von dem oben durchgeführten. Überlegen Sie sich aber noch einmal, warum wir in Versuch 3.2 vier Flipflops und zwei Takte benötigten.

Hochzählen

Q_1^n	Q_2^n	Q_1^{n+1}	Q_2^{n+1}	D_1	D_2
0	0				
0	1				
1	0				
1	1				

Formulieren Sie D_1 und D_2 als logische Funktion von Q_1^n und Q_2^n:

D_1=

D_2=

Herunterzählen

Q_1^n	Q_2^n	Q_1^{n+1}	Q_2^{n+1}	D_1	D_2
0	0				
0	1				
1	0				
1	1				

D_1=

D_2=

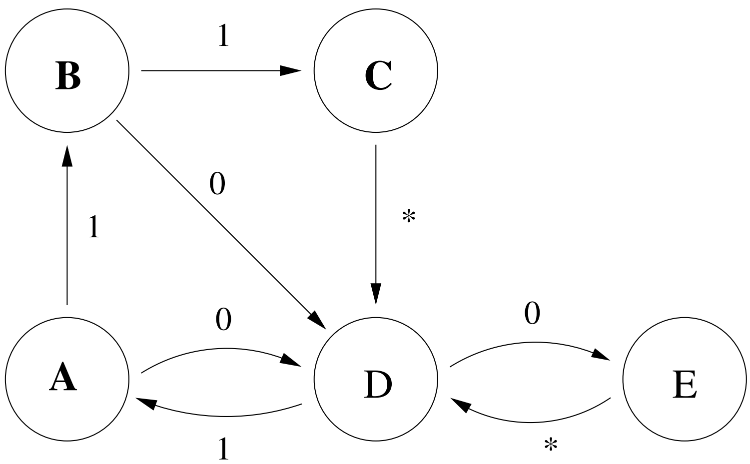
Entwerfen Sie nun wie oben einen Zähler, der in Abhängigkeit von einem Signal **UD** hoch- bzw. herunterzählen kann, mit den **pegelgesteuerten** Flipflops aus Versuch 3.2 und simulieren Sie ihn mit HADES, wobei das dort gesagte gilt.

Aufgabe gelöst:	Gruppe:	TeilnehmerIn: .... vom Betreuer auszufüllen

<u>Versuch 3.5:</u> Entwurf einer Ampelschaltung

Entwerfen Sie mit flankengesteuerten D-Flipflops einen Automaten, der in primitiver Weise die Funktion einer Verkehrsampel nachbildet. Der Automat soll dabei einen Eingang x und drei Ausgänge $y_1\,y_2\,y_3$ haben, die den Farben rot, gelb, grün entsprechen. Weiter habe der Automat fünf Zustände **A**= 'rot', **B**= 'rot/gelb', **C**= 'grün', **D**= 'gelb' und **E**, in dem keine der Lampen leuchtet.

Die Zustandsübergänge sollen wie folgt erfolgen:



Überlegen Sie sich eine geeignete Zustandscodierung $(q_3q_2q_1)$ (offenbar genügen für fünf Zustände drei Bit). Günstig ist dabei insbesondere eine, bei der jeder der drei Lampen eines der Bits direkt zugeordnet ist.

A=

B=

C=

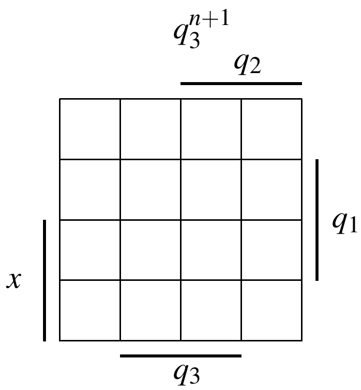
D=

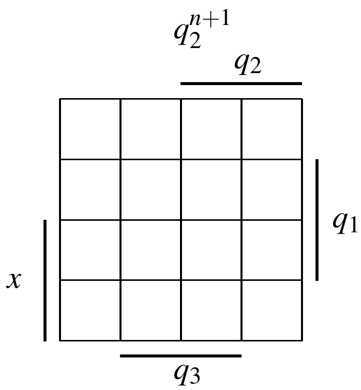
E=

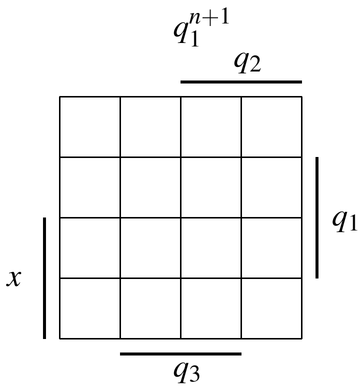
Füllen Sie jetzt die folgende Tabelle für die Zustandsübergänge aus (beachten Sie dabei die Möglichkeit von Don't Cares!):

	Alter Zustand			Neuer Zustand		
x	q_3^n	q_2^n	q_1^n	q_3^{n+1}	q_2^{n+1}	q_1^{n+1}
0	0	0	0			
0	0	0	1			
0	0	1	0			
0	0	1	1			
0	1	0	0			
0	1	0	1			
0	1	1	0			
0	1	1	1			
1	0	0	0			
1	0	0	1			
1	0	1	0			
1	0	1	1			
1	1	0	0			
1	1	0	1			
1	1	1	0			
1	1	1	1			

und ermitteln Sie die Schaltfunktionen für die Zustandsübergänge:







Tragen Sie bitte hier Ihre Schaltfunktionen ein (auch die, die eventuell noch zusätzlich zur Ansteuerung der drei Lampen notwendig sind) und skizzieren Sie Ihre

Schaltung.

Simulieren Sie die Schaltung mit HADES. Wählen Sie dabei bitte Flipflops vom Typ DFFR, die einen Reset-Eingang haben, mit dem sich die Schaltung in einen definierten Anfangszustand bringen lässt, bei dem alle Ausgänge Q den Wert **0** haben. Verwenden Sie bitte für die drei Lampen LEDs und **keine** Opins, weil sich diesen keine Farbe zuweisen lässt.

Aufgabe gelöst:	Gruppe:	TeilnehmerIn: .... vom Betreuer auszufüllen