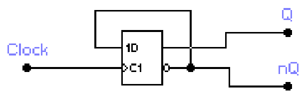
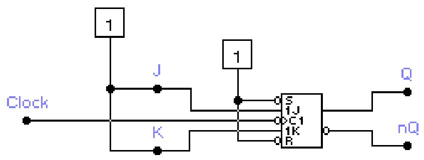
Kapitel 4

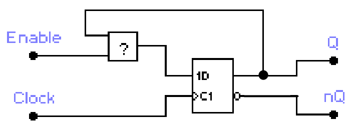
Versuch 415 T-Flipflop

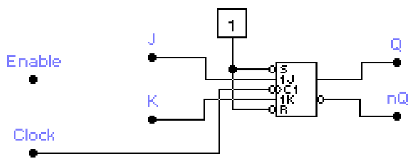
Flipflops, die mit jeder steigenden oder mit jeder fallenden Taktflanke in den entgegengesetzten Zustand kippen, heissen T–Flipflops („Toggle–Flipflops“). T-Flipflops können aus anderen Flipflops aufgebaut werden. In der ersten Zeile sind einfache T-Flipsflops dargestellt (ohne Enable-Eingang). Sie besitzen nur einen Takteingang. Das (linke) D-FF togglet, wenn man den $\overline{Q}$ -Ausgang auf den D-Eingang rückkoppelt. Rechts ist ein kombiniertes RS-JK-Flipflop abgebildet. Um ein „reines“ JK-Flipflop zu erhalten, werden die $\overline{S}$ und $\overline{R}$ Eingänge auf 1 gelegt. Das JK-Flipflop togglet, wenn man J = K = 1 setzt.

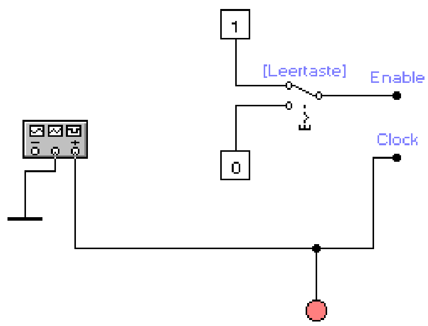
In der zweiten Zeile sind Entwürfe für T-Flipflops mit Enable-Eingang dargestellt. Der Enable-Eingang legt fest, ob das Flipflop togglet (seinen Zustand ändert) oder speichert (seinen Zustand nicht verändert).







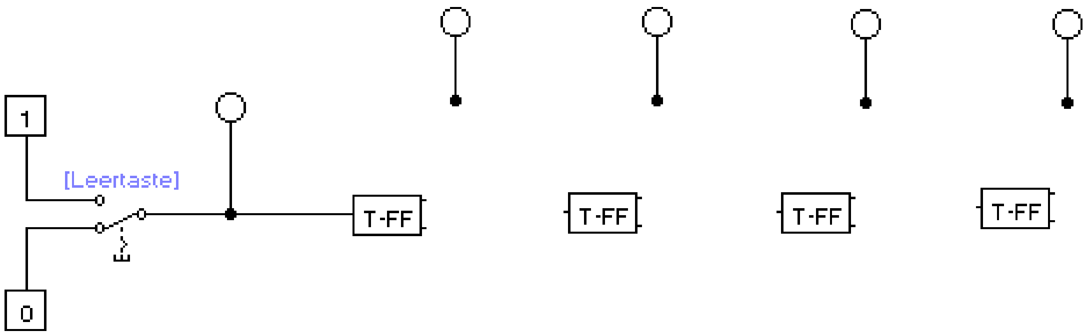




Erzeugen Sie in der Datei v415 aus den Schaltungen der 2. Zeile zwei Makros T-FF D und T-FF JK. Überlegen sie, welches Gatter beim D-Flipflop eingesetzt werden muss bzw. wie J und K mit Enable verbunden werden müssen. Testen Sie die Makros mit der unteren Schaltung. Bei welchem Wert für Enable togglen die Schaltungen, und bei welcher Flanke?

Falls erforderlich, modifiziern Sie beide Schaltungen so, dass sie zum einen bei Enable = 1 togglen, und zum anderen (mit ggfs. kleiner Verögerung) auf der aufsteigenden Flanke schalten.

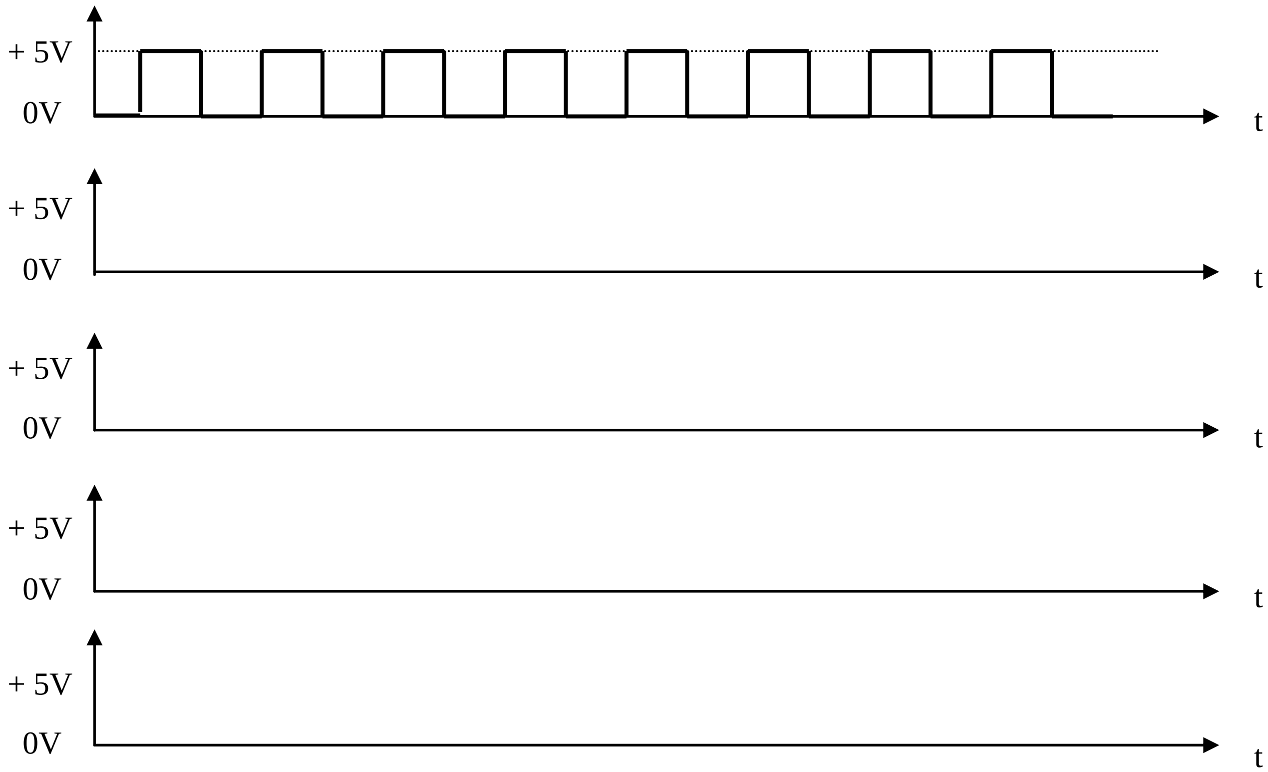
Versuch 420 Einfacher asynchroner Zähler (Ripple Counter)



In dem aufzubauenden Zähler schalten („kippen“) die T-Flipflops nacheinander, das erste (linke) Flipflop bei jedem Takt, das zweite jeden zweiten, das dritte jeden vierten usw. Die Taktflanken wandern wie eine Wellenfront von links nach rechts durch die Schaltung (Ripple: kleine Welle).

Vervollständigen Sie die Schaltung in der Datei v420, um das geforderte Verhalten zu erreichen. Der Zähler soll nach dem Einschalten 0000 anzeigen und dann hochzählen. Beachten Sie, dass das niederwertigste Bit links steht.

Ergänzen Sie ab dem Zählerstand 0000 das nachfolgende Timing. Nehmen Sie an, dass die Verzögerungszeit eines Flipflops etwa einem mm auf der t–Achse entspricht. Jedes Flipflops soll die gleiche Verzögerungszeit haben. Die leeren Diagramme stehen von oben nach unten für die Flipflops von links nach rechts. Das oberste Diagramm stellt den Takt (Leertaste) dar.



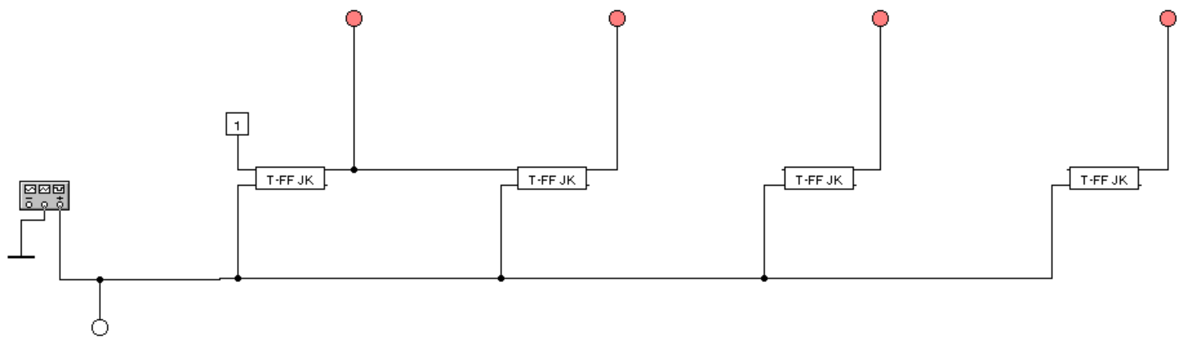
Versuch 425 Synchroner Zähler

Bei einem synchronen Zähler werden alle Flipflops gleichzeitig getaktet. In der Datei v425 finden sie die Grundschaltung eines synchronen 4-Bit Zählers mit T-Flipflops.

Für alle Zähler gilt:
Das FF an der niederwertigsten Stelle soll jeden Takt zählen (togglen).
Das FF an der zweit-niederwertigsten Stelle soll jeden zweiten Takt zählen.
Das FF an der dritt-niederwertigsten Stelle soll jeden vierten Takt zählen.

Allgemein: Das FF an der n-ten Stelle (beginnend mit n=0 für die erste niederwertigste Stelle) soll jeden 2^n-ten Takt zählen.

Simulatorbedingt zeigt der Zähler nach Einschalten der Simulation lauter Einsen an und fällt erst nach der ersten Taktflanke in den Anfangszustand (lauter Nullen) zurück. Wir betrachten erst ab diesem Zeitpunkt das Verhalten des Zählers.
Das erste FF soll jeden Takt zählen. Das erreichen Wir, indem Wir den Enable-Eingang konstant auf 1 legen. Das zweite FF soll zählen, wenn das erste den Zustand 1 hat. Das erreichen wir, indem wir den Enable-Eingang mit dem Ausgang des ersten FFs verbinden. Das dritte FF soll nur dann zählen, wenn die ersten beiden 1 anzeigen. Wie kann man das erreichen?



Ergänzen Sie obige Schaltung zu einem voll funktionsfähigen synchronen Zähler.

Theorie 430 Definition und Realisierung eines endlichen Automaten

Ein abstrakter Automat ist definiert als Tupel der folgenden Form:

$$(E, A, Z, \delta, \lambda)$$

Wiederholen sie anhand ihrer Unterlagen zu der Vorlesung „Rechnerstrukturen“ das Kapitel Automaten. Erklären sie mit eigenen Worten die oben angegebenen fünf Komponenten und ihr Zusammenspiel. Machen sie sich den Unterschied zwischen einem Mealy und einem Moore Automaten klar.

Vollziehen sie insbesondere anhand der Übungen zu RS den Schaltwerkentwurf vom Automatenmodell bis zur Gatterschaltung nach.

Die nachfolgende Schaltung stellt eine Realisiserung eines Automaten auf Gatterebene dar.

Machen sie sich die Schaltung verständlich und beantworten sie dabei die folgenden Fragen:

1. Wie wurden die einzelnen Komponenten des Automaten realisiert?

2. Wie gross sind die Kardinalitäten der Mengen des Automaten maximal?

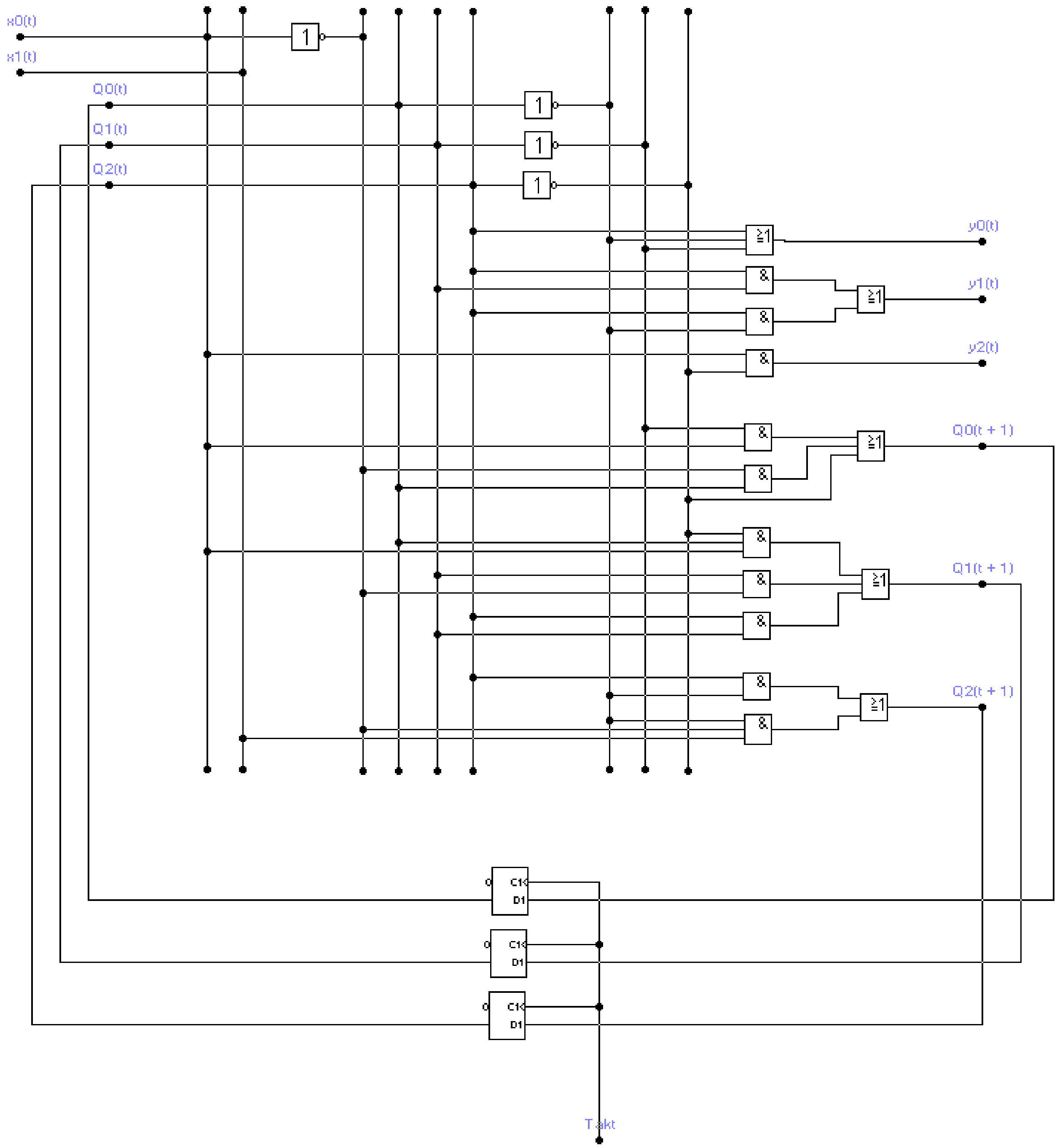
E	
A	
Z	

3. Um welchen Automatentyp handelt es sich? Woran erkennt man das?

Mealy	
Moore	

4. Wie ist das dynamische Verhalten (die Arbeitsweise) eines Mealy-Automaten? Wie wurde dies in der Schaltung umgesetzt? Welche Laufzeiten und Verzögerungen sind zu beachten?

Realisierung eines Automaten auf Gatterebene:



Versuch 435 Entwurf eines synchronen Vorwärts-Rückwärtszählers

In diesem Versuch sollen Sie enen Vorwärts–Rückwärtszähler entwerfen, der nach dem folgendem Prinzip arbeitet.

Spezifikation:

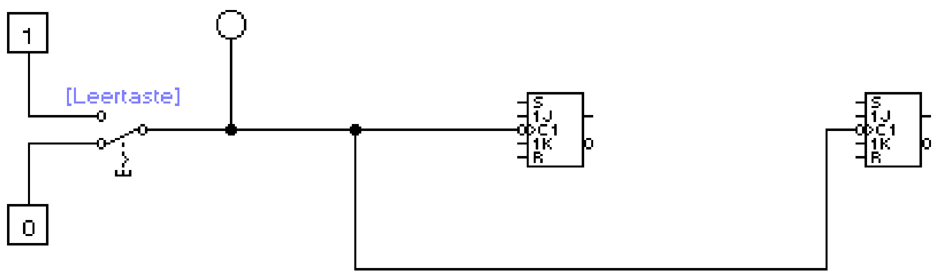
Der Zähler soll drei Eingänge und zwei Ausgänge haben:

- einen Takteingang mit dem Namen „Clock“,
- einen Steuereingang mit dem Namen „Count“,
- einen Steuereingang mit dem Namen „Down“,
- zwei Ausgänge mit den Namen z_0 und z_1.

Die Schaltung soll jeden am Takteingang eingegebenen 0–1–0 Impuls modulo 4 zählen und den Zählerstand als Binärzahl an den Ausgängen z_1 und z_0 ausgeben. Im Grundzustand, d.h. beim Einschalten des Taktes soll die Schaltung sich im Zählerstand z_1z_0 = 00 befinden.

- Wenn am Steuereingang Count = 0 eingegeben wird, soll die Schaltung nicht zählen.
- Wenn am Steuereingang Count = 1 eingegeben wird, soll die Schaltung zählen.
- Wenn am Steuereingang Down = 0 eingegeben wird, soll die Schaltung vorwärts zählen:
  $[z_1, z_0]$ = [0, 0], [0, 1], [1, 0], [1, 1], [0, 0], [0, 1], ...
- Wenn am Steuereingang Down = 1 eingegeben wird, soll die Schaltung rückwärts zählen:
  $[z_1, z_0]$ = [0, 0], [1, 1], [1, 0], [0, 1], [0, 0], [1, 1], ...

Die Schaltung soll mit zwei Exemplaren eines EWB–internen Flipflops realisiert werden. Die miteinander verbundenen Takteingänge der beiden Flipflops bilden den Takteingang der gesamten Schaltung:

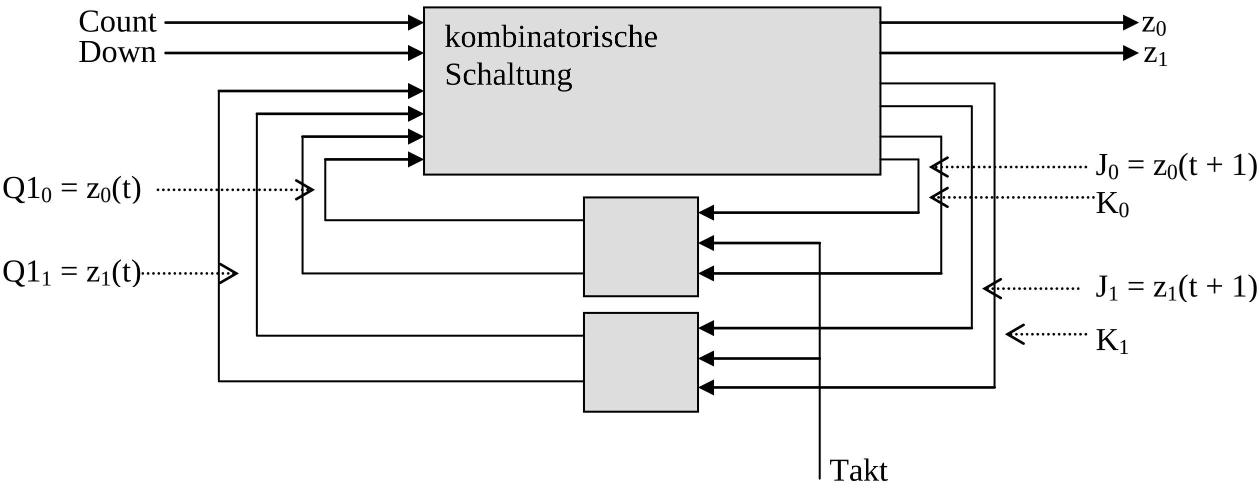


Das verwendete EWB–interne JK–Flipflop hat die folgende Zustandsübergangstabelle:

Q1(t)	**→**	**Q1(t + 1)**	**J**	**K**
0	→	0	0	X
0	→	1	1	X
1	→	0	X	1
1	→	1	X	0

Die Versuchsbeschreibung wird auf den nächsten beiden Seiten fortgesetzt.

Die zu entwerfende Schaltung soll folgende Form haben:



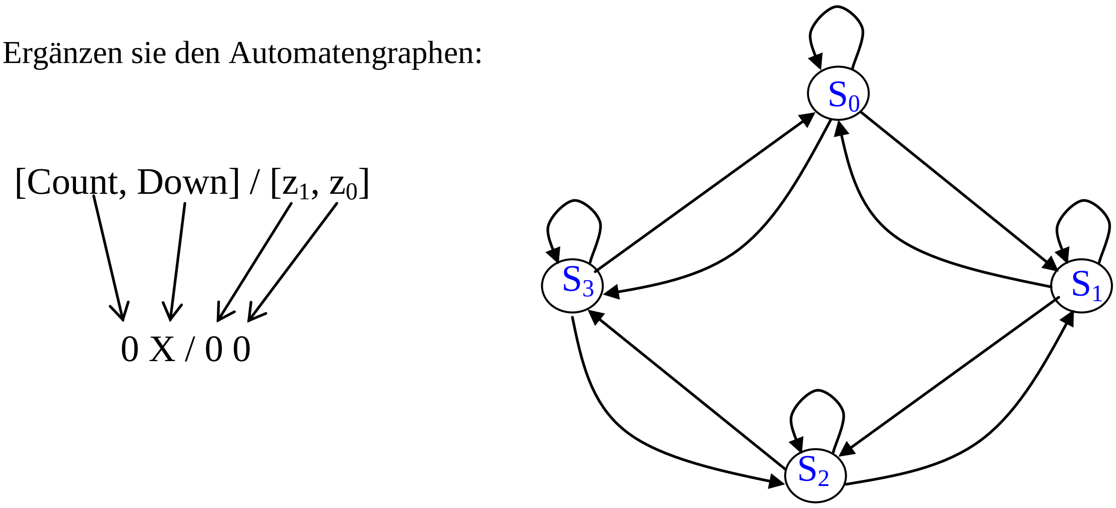
Die Zustände des zu entwerfenden Automaten sind identisch mit den Ausgangszuständen, d.h. die Ausgabefunktion ist die identische Abbildung.

[0,0], [0,1], [1,0], und [1,1] sind die vier möglichen Zählerzustände des Automaten.

Es gilt: $[0,0] = S_0$
$[0,1] = S_1$
$[1,0] = S_2$
$[1,1] = S_3$

Automatentheoretischer Entwurf des Zählers

Ergänzen sie den Automatengraphen:



Man kann in diesem Fall die Zustandsübergangstabelle des gesamten zu entwerfenden Automaten sofort hinschreiben (vervollständigen sie die Tabelle.):

Count Down	**Zustände** **S(t) → S(t+1)**	**Codiert**	**$Q1_1(t+1)$**	**J_1 K_1**	**$Q1_0(t+1)$**	**J_0 K_0**
0 0 0 0 0 0 0 0	$S_0 \rightarrow S_0$ $S_1 \rightarrow S_1$ $S_2 \rightarrow S_2$ $S_3 \rightarrow S_3$	00 → 00 01 → 01 10 → 10 11 → 11				
0 1 0 1 0 1 0 1						
1 0 1 0 1 0 1 0						
1 1 1 1 1 1 1 1						

Entwerfen und minimieren Sie nun die Ansteuerungen der Flipflops mit Hilfe von KV–Diagrammen.

J1

		d	d	
			c	c
$Q1_0(t)$				
$Q1_1(t)$ $Q1_0(t)$				
$Q1_1(t)$				

K1

J0

K0

Bauen Sie dann ihren Schaltungsentwurf in der Datei v435 auf.
Ergänzen Sie ihre Schaltung durch eine „Clear–Schaltung“, die es ermöglicht, den Grudzustand $Q1_0 = 0$; $Q1_1 = 0$ mit einem Tastenschalter einzustellen, und zwar ohne Taktflanke.
Hinweis: Die R- und S-Eingänge des Flipflops haben höhere Priorität gegenüber den J- und K-Eingängen und sind nicht taktgesteuert.

Versuch 445 Asynchroner Zähler modulo n

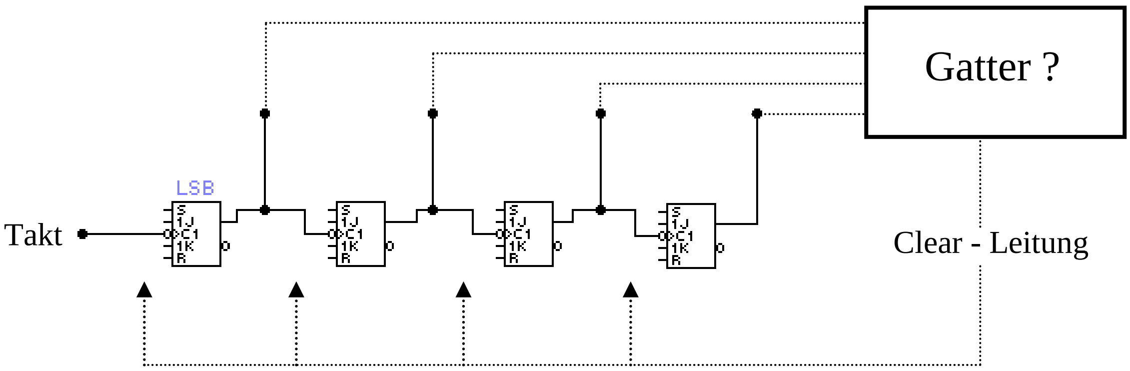
Um das Prinzip des binär zählenden, aus T–Flipflops aufgebauten asynchronen Ripple–Counters zu verstehen, bedarf es nicht der Automatentheorie. Wenn ein nach diesem Prinzip aus m T–Flipflops aufgebauter Zähler mit dem Zählerstand Null (00 ...0) beginnt, springt er nach 2^m Zählschritten wieder auf Null, d.h. er wird nach 2^m Zählschritten „zyklisch“, er zählt modulo 2^m. Der grösste Zählerstand beträgt 2^m-1.

Man kann ohne automatentheoretischen Entwurf auch einen asynchronen Ripple–Counter entwickeln, der modulo n zählt, wobei $n < 2^m$ ist.

Sie finden in der Datei v445 eine Gatterschaltung (d.h. eine kombinatorische Schaltung), die einen aus vier T–Flipflops aufgebauten asynchronen Ripple–Counter modulo 12 zählen lässt.

Die Gatterschaltung soll eine Clear–Schaltung sein, die den Zähler beim Zählerstand 12 auf $Q1_0 = Q1_1 = Q1_2 = Q1_3 = 0$ setzt.

Die Schaltung soll intuitiv, d.h. ohne algorithmisches Entwurfsverfahren gefunden werden.
Gehen Sie davon aus, dass nach dem Einschalten für alle Flipflops Q=0 gilt.



Ergänzen Sie den Zähler um die von Ihnen entworfene Gatterschaltung. Erklären Sie das Verhalten an der „kritsichen“ Stelle, also kurz vor und beim Zurücksetzen.

Def. Eine generische modulo-n Schaltung ist ein Bauplanschema mit einem Parameter n, der angibt, bei welchem Zählerzustand die Schaltung wieder auf 0 gesetzt werden soll.

Wie sieht eine generische modulo-n Schaltung aus mit
$b_{k-1} \ldots b_0$ mit $b_i \in \{0,1\}$sei die Betragszahldarstellung von n ?

Dasselbe Verfahren kann man bei synchronen Zählern modulo n anwenden.