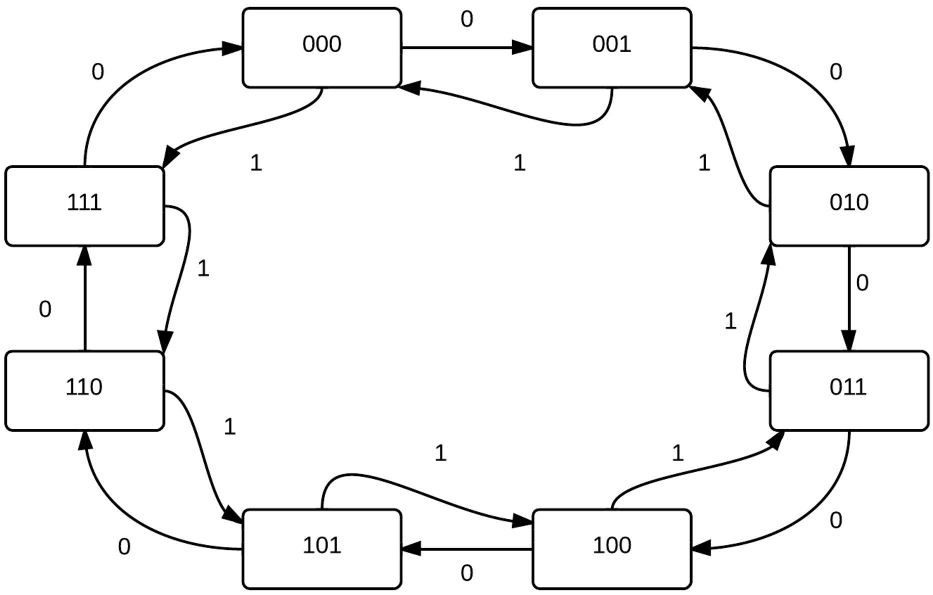
Technische Informatik: Abgabe 8

Michael Mardaus Andrey Tyukin

19. Dezember 2013

Exercise 8.1 (JK Flipflop Ringcounter)

a) We want to model a ring-counter that counts forwards if the input w is 0, and backwards if w is 1. It can be modeled by a Moore-automaton (the output depends only on the state). The state diagram looks as follows:

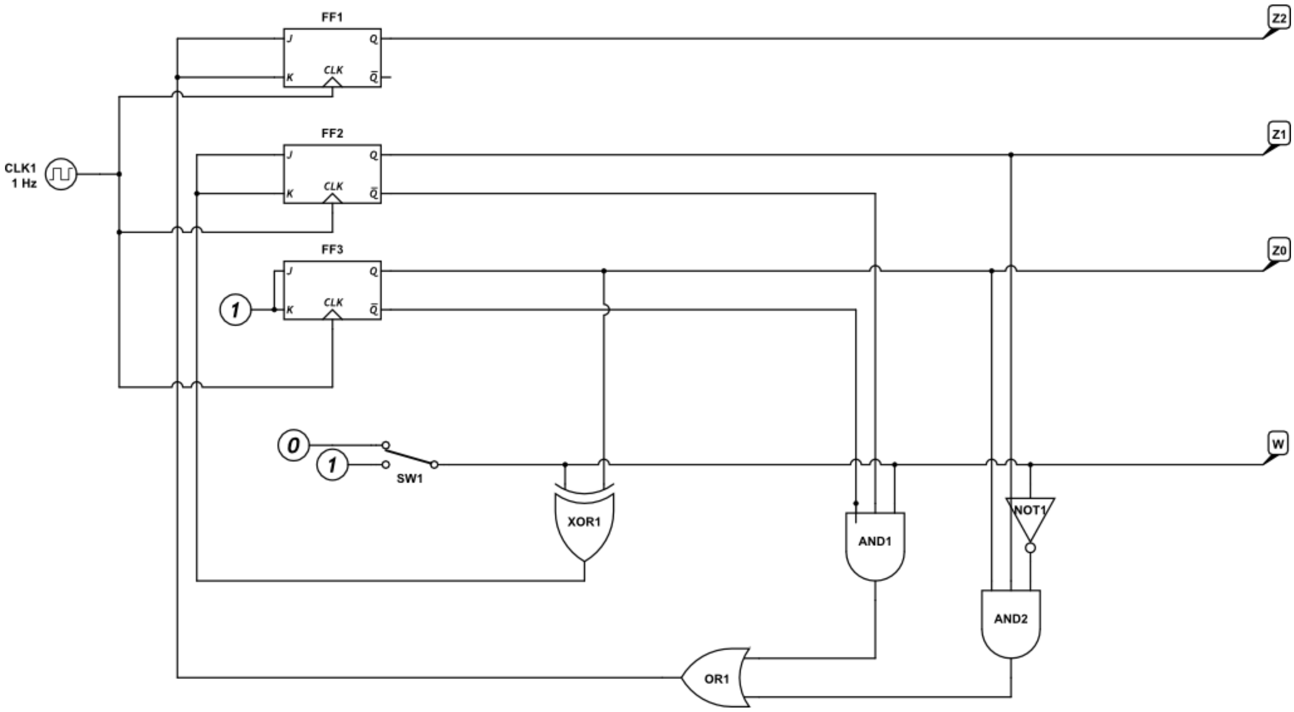


State diagram for the ring-counter. Output is omitted, it's just equal to the node index.

Same in table-form:

State	w = 0	w = 1	Output
000	001	111	000
001	010	000	001
010	011	001	010
011	100	010	011
100	101	011	100
101	110	100	101
110	111	101	110
111	000	110	111

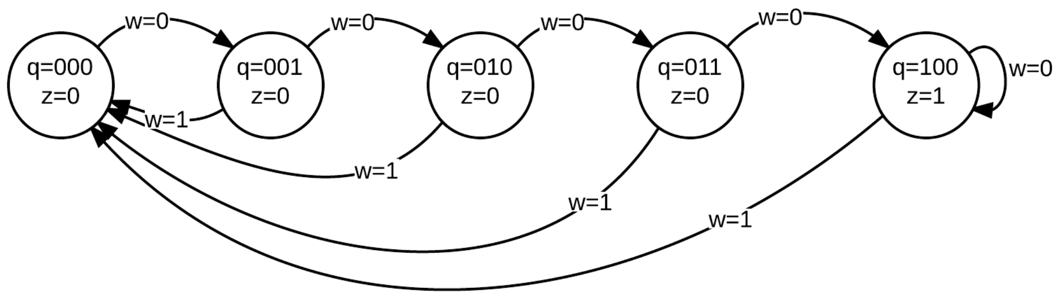
b) The corresponding circuit looks as follows:



Exercise 8.2 (JK Flipflip for 4 equal inputs)

First we make one input signal w out of the w_1 and w_2 to make the automaton simpler. Therefore we XOR the two signals to one. If both signals are equal XOR makes $w = 0$. If they are different XOR will be $w = 1$. The state diagram for the automaton:

State	Next state								Output
	$w = 0$				$w = 1$				
$y_2y_1y_0$	$Y_2Y_1Y_0$	J_2K_2	J_1K_1	J_0K_0	$Y_2Y_1Y_0$	J_2K_2	J_1K_1	J_0K_0	z
000	001	0d	0d	1d	000	0d	0d	0d	0
001	010	0d	1d	d1	000	0d	0d	d1	0
010	011	0d	d0	1d	000	0d	d1	0d	0
011	100	1d	d1	d1	000	0d	d1	d1	0
100	100	d0	0d	0d	000	d1	0d	0d	1
101	ddd	dd	dd	dd	000	dd	dd	dd	d
110	ddd	dd	dd	dd	000	dd	dd	dd	d
111	ddd	dd	dd	dd	000	dd	dd	dd	d



This leads to these K-maps:

J_0	y_1y_0			
wy_2	00	01	11	01
00	1	d	d	1
01		d	d	d
11		d	d	d
10		d	d	

J_1	y_1y_0			
wy_2	00	01	11	01
00		1	d	d
01		d	d	d
11		d	d	d
10			d	d

J_2	y_1y_0			
wy_2	00	01	11	01
00			1	
01	d	d	d	d
11	d	d	d	d
10				

K_0	y_1y_0			
wy_2	00	01	11	01
00	d	1	1	d
01	d	d	d	d
11	d	d	d	d
10	d	1	1	d

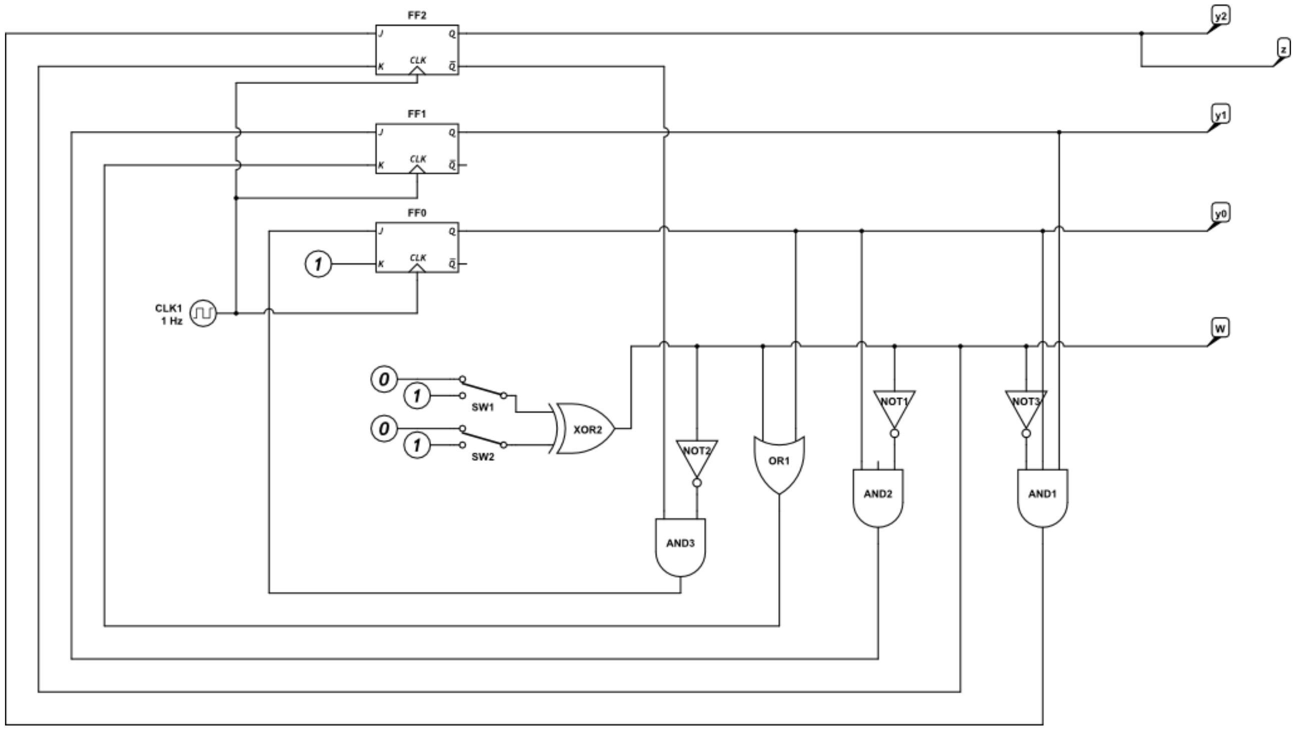
K_1	y_1y_0			
wy_2	00	01	11	01
00	d	d	1	
01	d	d	d	d
11	d	d	d	d
10	d	d	d	1

K_2	y_1y_0			
wy_2	00	01	11	01
00	d	d	d	d
01		d	d	d
11	1	d	d	d
10	d	d	d	d

These K-maps lead us to:

$$
\begin{aligned}
J_0 &= \bar{w}\bar{y}_2 \\
J_1 &= \bar{w}y_0 \\
J_2 &= \bar{w}y_1y_0 \\
K_0 &= 1 \\
K_1 &= w + y_0 \\
K_2 &= w
\end{aligned}
$$

Which brings us to this circuit:

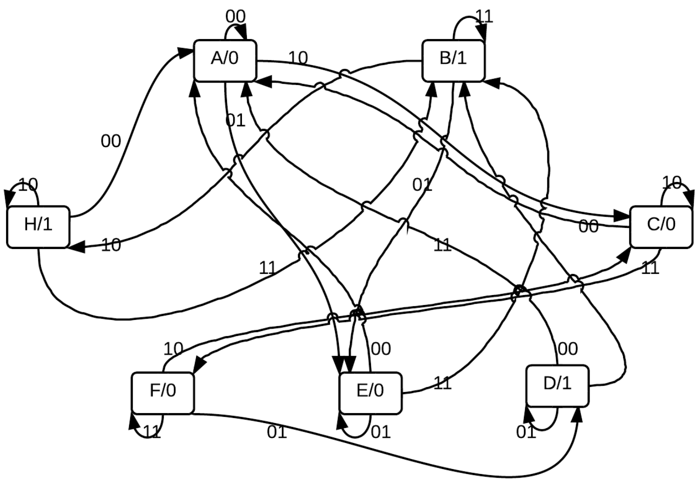


Exercise 8.3

a) We keep partitioning the states until nothing changes:

Initial partition	(ABCDEFGH)
Different outputs	(ACEFG)(BDH)
Second column	(AEG)(C)(F)(B)(D)(H)
Thirrd column	(AG)(E)(C)(F)(B)(D)(H)
No changes.	

That is, only the states A and G seem to be equivalent. The corresponding state diagram looks as follows:



b) We now number all the states by indices from 000 to 110:

State	In = 00	In = 01	In = 10	In = 11	Output
A = 000	000	100	010	ddd	0
B = 001	ddd	100	110	001	1
C = 010	000	ddd	010	101	0
D = 011	000	011	ddd	001	1
E = 100	000	100	ddd	001	0
F = 101	ddd	011	010	101	0
H = 110	000	ddd	110	001	1

Now one could create three K-Maps for 5 variables (state transitions), one K-Map for 3 variables (mapping from 3 registers to the output), select the prime-implicants, and implement the corresponding circuit with D-Flip-Flops. (Todo, incomplete)