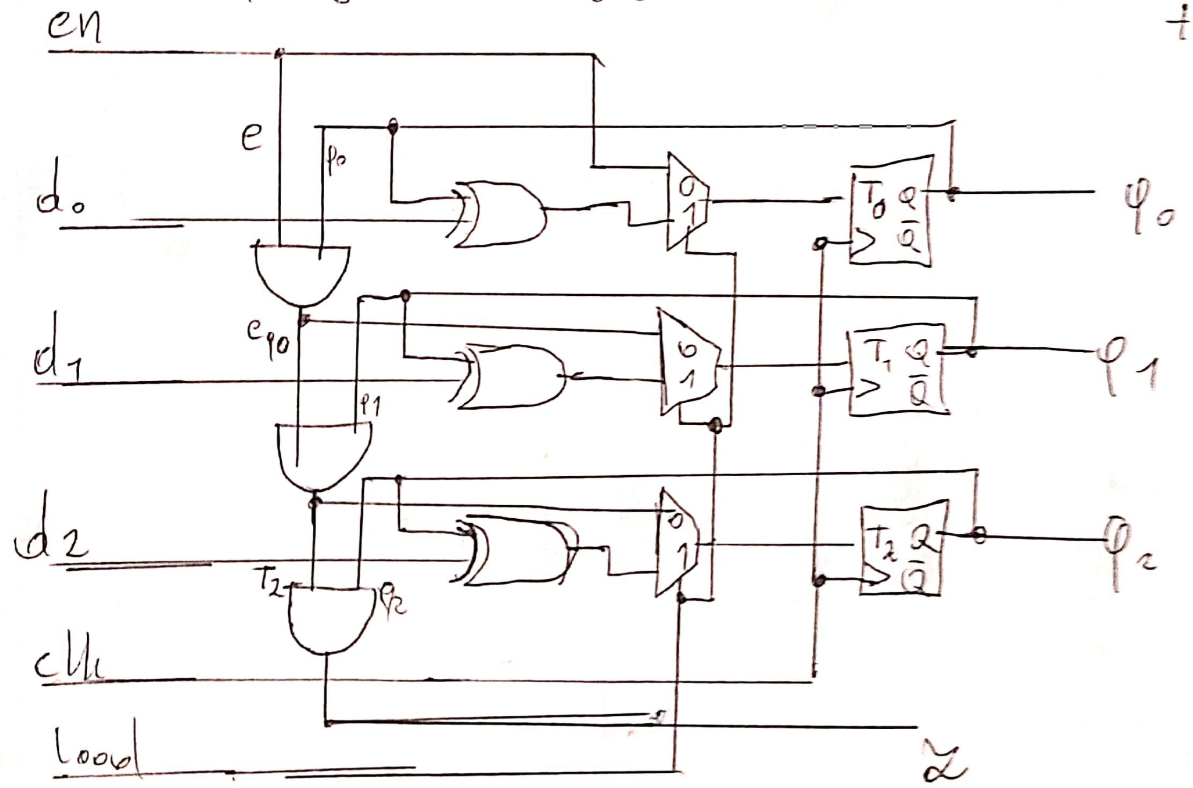
1) Mamy (z wykł.) $T_0 = e$, $T_1 = e q_0 = T_0 q_0$, $T_2 = e q_0 q_1 = T_1 q_1$ | $T_0 = d_0 \wedge q_0$

—||—, gdy load $= 0$, gdy load $= 1$ to $T = \bar{Q}'$ czyli dla ~~$t = Q' = \ldots \Rightarrow T = 0$~~

$t = Q' = 0$ $t = Q' = 1 \Rightarrow T = 0$

(XOR)



② dla $\overline{up}/down = 0$ tak samo jak na wykładzie

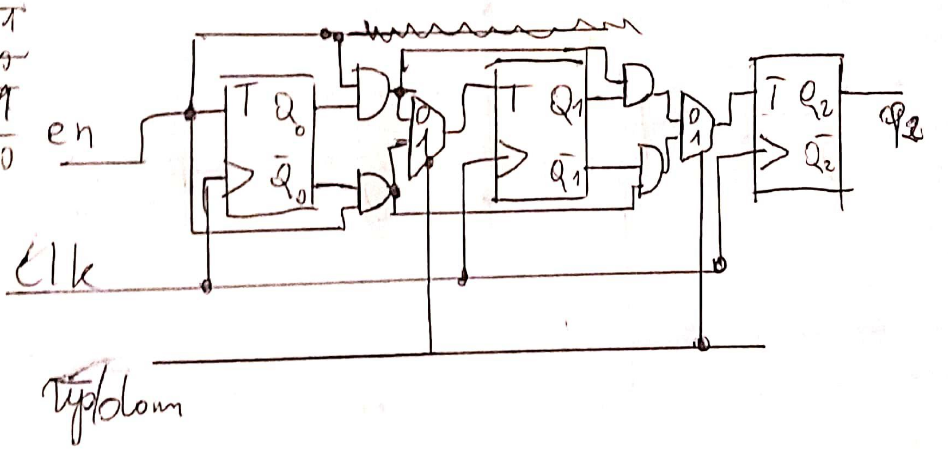
$\overline{up}/down = 1$

q_0 alternuje

q_1 zmienia stan, gdy $q_0' = 0 \Rightarrow \overline{q_0}' = 1$

q_2 zmienia, gdy $q_1' = 0 \wedge q_0' = 0 \Rightarrow \overline{q_1}'\overline{q_2}' = 1$

n	$q_2\, q_1\, q_0$
8	0 0 0
7	1 1 1
6	1 1 0
5	1 0 1
4	1 0 0
3	0 1 1
2	0 1 0
1	0 0 1
0	0 0 0



(3) Ponówmy T analogicznie do p. Wtedy gdy na początku mamy $p_0 = p_1 = p_2 = 0$ zawsze $T_0 = 1$, $T_1 = p_0$, $T_2 = p_1$

p_0	p_1	p_2	T_0	T_1	T_2
0	0	0	1	0	0
1	0	0	1	1	0
0	1	0	1	0	1
1	1	1	1	1	1
0	0	0	0	0	0 ⇐ cykl

Czyli $\{p_2, p_1, p_0\}$ jest w sekwencji:

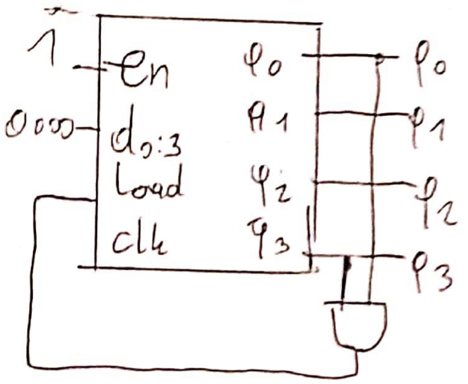
$000 \to 001 \to 010 \to 111 \to 000 \to \dots$

5) Załóżmy, że licznik z wykładu używa przerzutników T (czyli taki jak z zad 1 ~~)~~ tylko dla 4 bitów).

$T_0 = e$, $T_1 = e q_0 = T_0 q_0$, $T_2 = e q_0 q_1 = T_1 q_1$, $T_3 = e q_0 q_1 q_2 = T_2 q_2$

Nieużywane stany to 1010, 1011, 1100, 1101, 1110, 1111

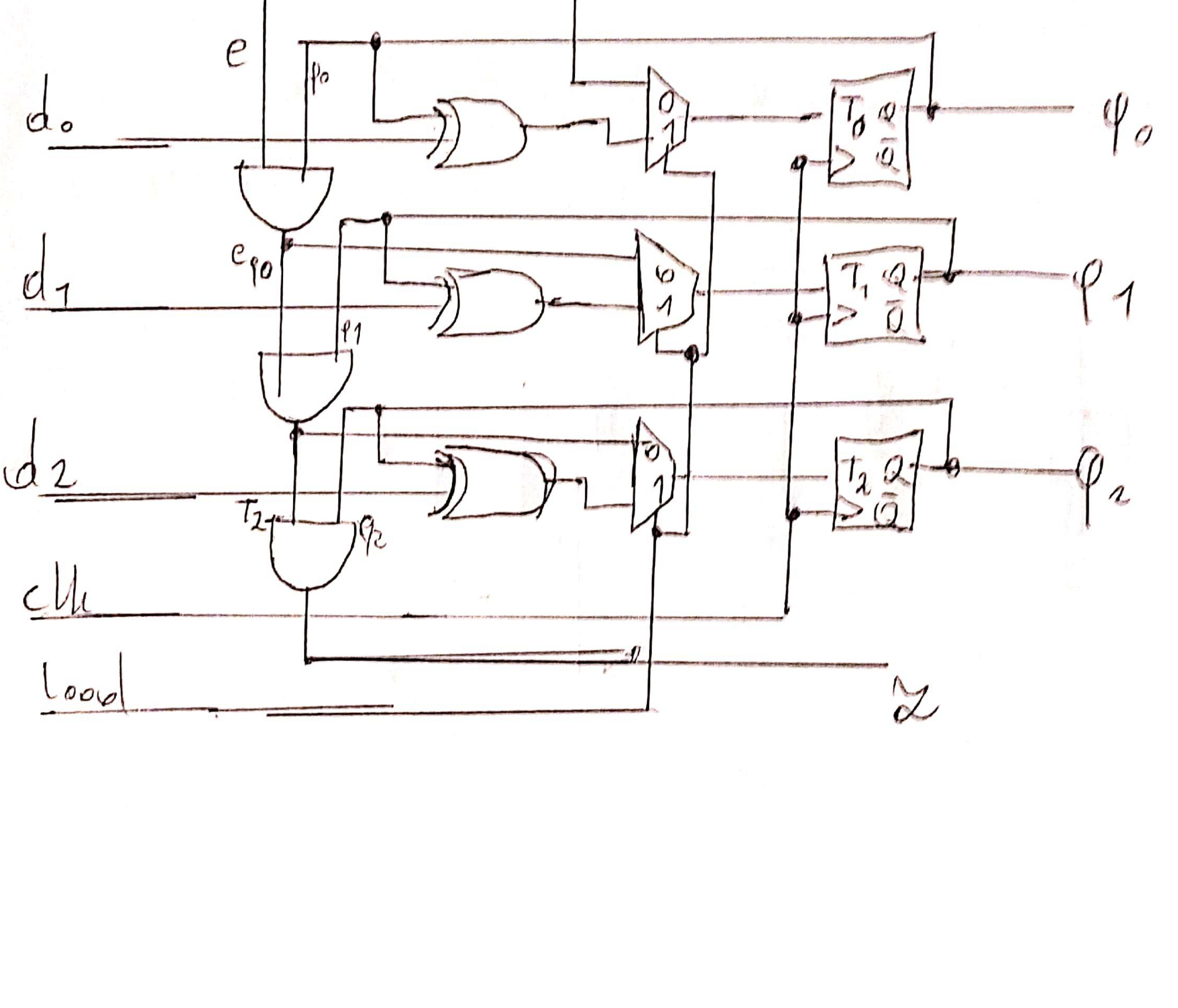
10, 11, 12, 13, 14, 15



$q_3 \ldots q_0$: 1011 ↓ 0000	1101 ↓ 0000	1111 ↓ 0000	bo $q_0 q_3 = 1$
1010 $\downarrow T_0$ 1011 ↓ 0000	1100 $\downarrow T_0$ 1101 ↓ 0000	1110 $\downarrow T_0$ 1111 ↓ 0000	

Czyli w każdym z przypadków w co najwyżej 2 cyklach układ ~~wróci~~ się zresetuje (poprawnie ustawi)

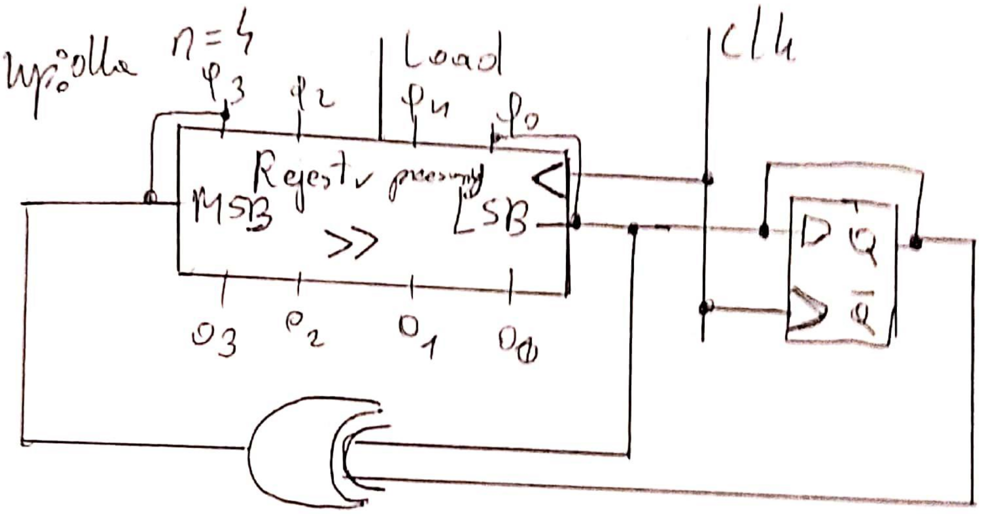
d) Mamy (z wykł.) $T_0 = e$, $T_1 = e\varphi_0 = T_0\varphi_0$, $T_2 = e\varphi_0\varphi_1 = T_1\varphi_1$ | $T_0 = d_0 \wedge \varphi_0$
— || —, gdy load $= 0$, gdy load $= 1$ to $T = Q'$ więc dla $1 = Q' = 0$ ~~wtedy $Q' = 1 = 1 = 0$~~ (XOR)
$1 = Q' = 0$ $t = Q' = 1 \Rightarrow 1 = 0$
(XOR)



d_0	φ_0	T
0	0	0
0	1	1
1	0	1
1	1	0

XOR

4



Mamy n-bitowy rejestr przesuwny, przesuwający w kierunku najmniej znaczącego bitu oraz przerzutnik D zapamiętujący pierwsze wystąpienie jedynki.

Żeby znaleźć uzupełnienie do 2 wystarczy, że przepiszemy wszystkie zera, aż do pierwszego wystąpienia jedynki (włącznie), a kolejne bity zanegujemy.

np.: $\underbrace{1\,0\,0}_{\sim}\,\underline{1}\,0\,0 \rightarrow \underbrace{0\,1\,1}_{\sim}\,\underline{1}\,0\,0$ | $000 \rightarrow 000$ | $\underbrace{0\,0}_{\sim}\underline{1} \rightarrow 11\underline{1}$

pierwsze wystąpienie jedynki

~~LSB~~	~~Q(spotkano 1)~~	~~XOR~~
~~0~~	~~0~~	~~0 (=0)~~
~~1~~	~~0~~	~~0 (=1)~~
~~0~~	~~1~~	
~~1~~	~~1~~	

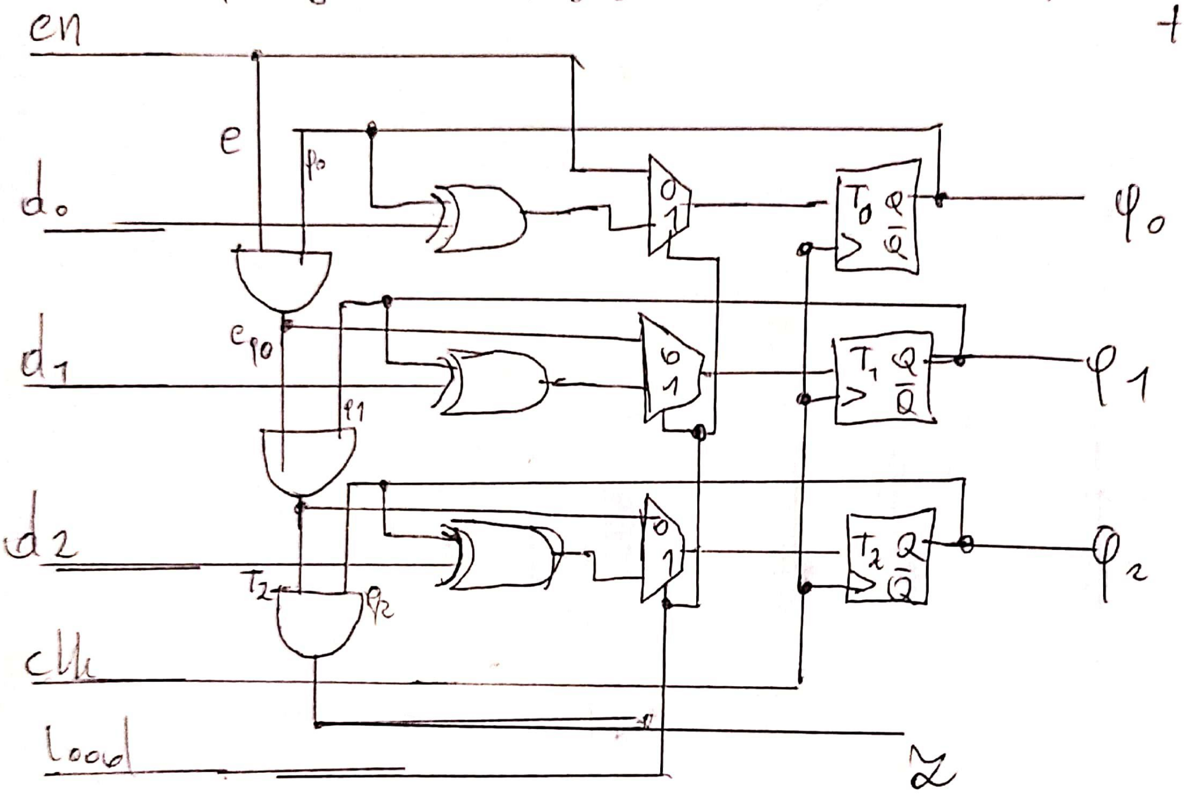
LSB	Q(spotkano 1)	XOR	
0	0	0	przepisywanie
1	0	1	
0	1	1	neguje
1	1	0	

1) Mamy (z wykł.) $T_0 = e$, $T_1 = e\varphi_0 = T_0 \varphi_0$, $T_2 = e\varphi_0\varphi_1 = T_1\varphi_1$ | $T_0 = d_0 \wedge \varphi_0$

-||-, gdy load $= 0$, gdy load $= 1$ to $T = \overline{Q}'$ czyli dla ~~$t = Q' = 0$ ... $t = Q' = 1 \Rightarrow T = 0$ (XOR)~~

$t = Q' = 0$ $t = Q' = 1 \Rightarrow T = 0$

(XOR)



d_0	φ_0	T
0	0	0
0	1	1
1	0	1
1	1	0

XOR

③ Ponazwijmy T analogicznie do φ. Wtedy gdy na początku mamy $\varphi_0, \varphi_1, \varphi_2 = 0$ zawsze $T_0 = 1$, $T_1 = \varphi_0$, $T_2 = \varphi_1$

φ_0	φ_1	φ_2	T_0	T_1	T_2	
0	0	0	1	0	0	
1	0	0	1	1	0	
0	1	0	1	0	1	
1	1	1	1	1	1	
0	0	0	0	0	0	$\Leftarrow$ cykl

Czyli $\{\varphi_2, \varphi_1, \varphi_0\}$ jest w sekwencji:

$000 \to 001 \to 010 \to 111 \to 000 \to \ldots$