

## Peta Domaca Zadaca iz Digitalne Logike-Tocna Rješenja

1.

Zadan je sklop koji je ostvaren kao sinkrono brojilo s 4 D bistabila. Poznato je da stanje 7 pripada glavnom ciklusu brojila (prilikom očitavanja stanja izlaz Q0 bistabila B0 promatrati kao bit najmanje težine).

Sljedece stanje svakog bistabila je opisano ovim algebarskim zapisima:

$$D3(Q3, Q2, Q1, Q0) = (Q3' * Q1' * Q0') + (Q3 * Q1) + (Q2' * Q1' * Q0') + (Q2 * Q1),$$

$$D2(Q3, Q2, Q1, Q0) = (Q3' * Q2' * Q1') + (Q3 * Q1 * Q0') + (Q2' * Q1' * Q0) + (Q2 * Q1 * Q0'),$$

$$D1(Q3, Q2, Q1, Q0) = (Q3' * Q0) + (Q3 * Q2' * Q1') + (Q3 * Q2' * Q0') + (Q3 * Q1' * Q0'),$$

$$D0(Q3, Q2, Q1, Q0) = (Q3' * Q2 * Q0') + (Q3' * Q1' * Q0) + (Q3 * Q2 * Q1 * Q0) + (Q3 * Q1' * Q0') + (Q2' * Q1 * Q0').$$

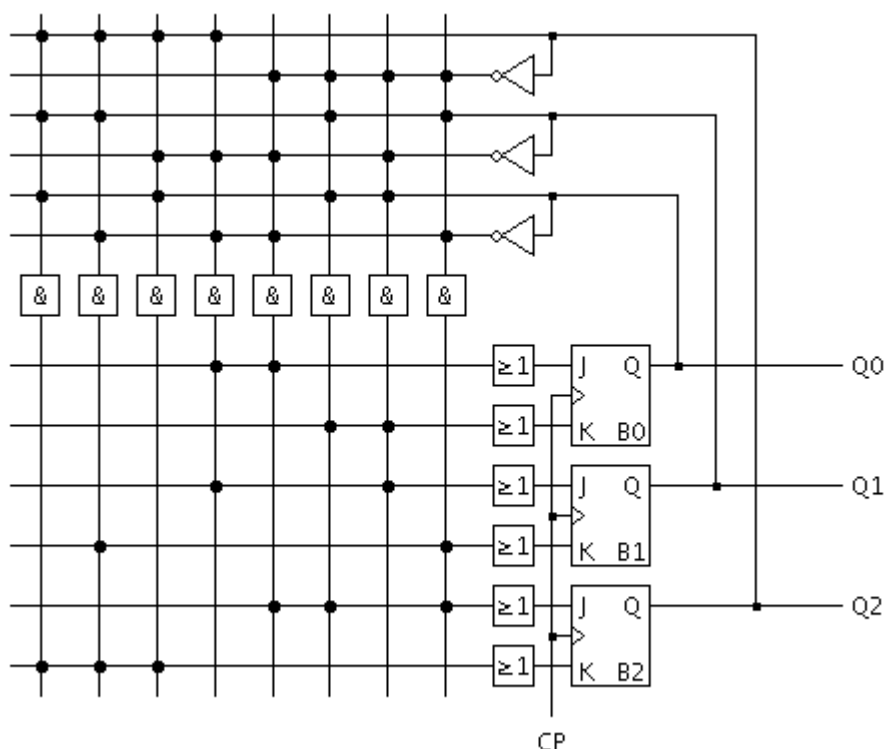
Ima li zadani sklop siguran start?

Objašnjenje korištenih oznaka: oznaka " ' " iza nekog ulaza Qx zamjenjuje operator 'NOT' (dakle Qx' znaci NOT Qx), znak " \* " zamjenjuje operator 'AND' a znak " + " zamjenjuje operator 'OR'.

**RJ. ne, nema siguran start**

2.

Neko sinkrono brojilo prikazano je sljedećom slikom.

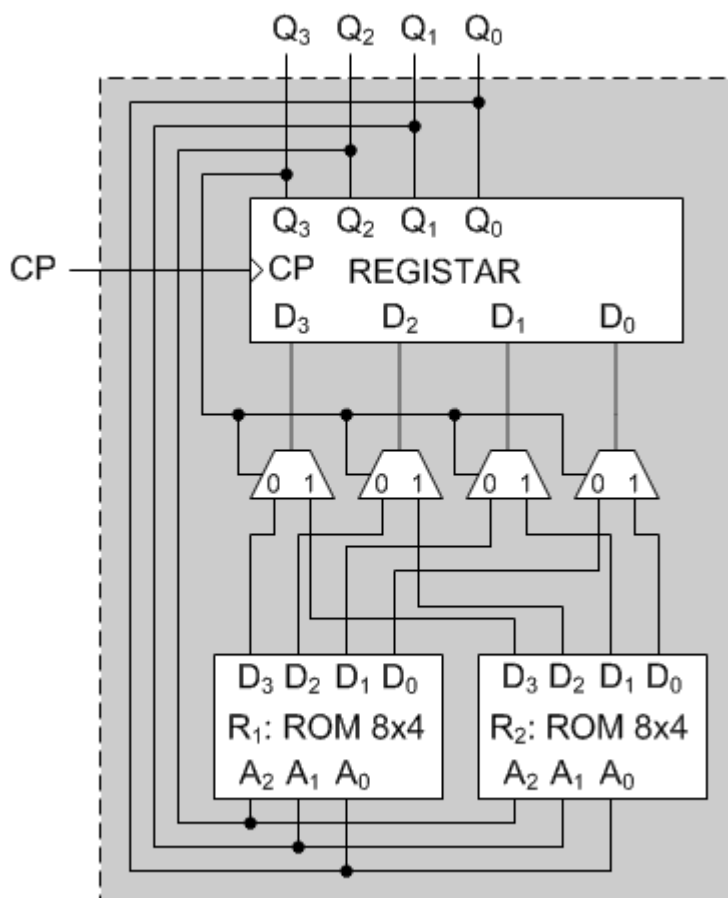


U kojem ciklusu broji to brojilo? Prilikom očitavanja stanja izlaz Q0 tretirati kao bit najmanje težine.

**RJ. 2, 4, 7, 3, 6, 0, 5, 1, 2**

3.

Na raspolaganju je sklop prikazan na slici.



Memorije  $R_1$  i  $R_2$  potrebno je programirati tako da se dobije brojilo koje broji u ciklusu 3 -> 9 -> 8 -> 1 -> 12 -> 14 -> 15 -> 13 -> 5 -> 7 -> 10 -> 6 -> 4 -> 2 -> 0 -> 11 (prilikom očitavanja stanja  $Q_3$  tretirati kao bit najveće težine). Napomena: kao rješenje upišite sadržaj memorija po lokacijama, i to svaku lokaciju kao jednu heksadekadsku znamenku.

RJ.

R1.0: **B**

R1.1: **C**

R1.2: **0**

R1.3: **9**

R1.4: **2**

R1.5: **7**

R1.6: **4**

R1.7: **A**

R2.0: **1**

R2.1: **8**

R2.2: **6**

R2.3: **3**

R2.4: **E**

R2.5: **5**

R2.6: **F**

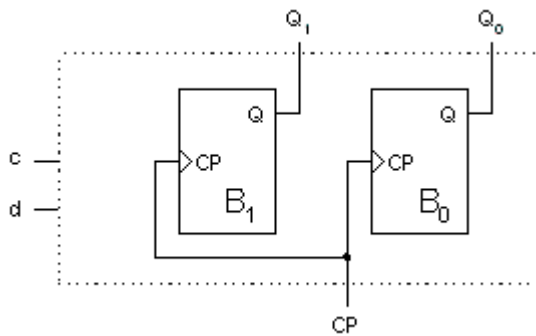
R2.7: **D**

4.

Uporabom SR bistabila realizirati 2-bitno brojilo koji broji ovisno o signalu d: ako je  $d=1$ , tad je  $\text{sljedece\_stanje} = \text{trenutno\_stanje}+1$ , inace  $\text{sljedece\_stanje} = \text{trenutno\_stanje}+0$  (pod pojmom stanje misli se na binarno kodirani broj zapisan kroz bistabile, pri cemu je izlaz Q0 izlaz najmanje težine). Sklop treba imati i sinkroni ulaz za brisanje c (kojeg bistabili nemaju). Koristiti minimalni broj osnovnih logickih sklopova. U svako polje za unos potrebno je unijeti algebarski zapis funkcije tog bistabila. Prilikom očitavanja stanja izlaz Q0 bistabila B0 tretirati kao izlazni bit najmanje težine.

Prilikom unosa algebarskog oblika za stanja bistabila koristiti oznake Qj (gdje je j broj bistabila; npr. Q2). Primjer jednog takvog rješenja:

c and Q2 and not Q1 or not c and d and not Q0.



RJ.

B1.S: NOT C AND D AND NOT Q1 AND Q0

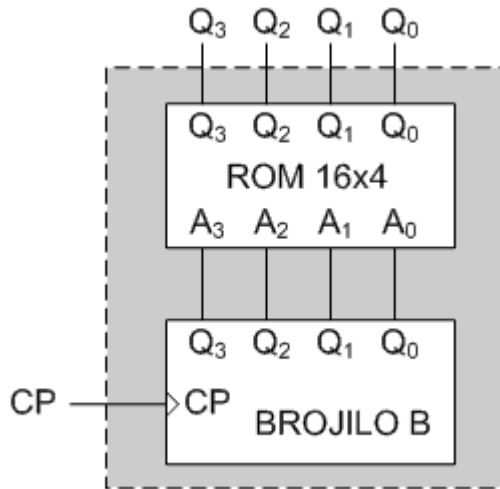
B1.R: (c) OR (d AND Q1 AND Q0)

B0.S: NOT C AND D AND NOT Q0

B0.R: (c) OR (d AND Q0)

5.

Na raspolaganju je sklop prikazan na slici.



Brojilo B koje je iskorišteno za brojanje broji u ciklusu 10 -> 3 -> 12 -> 6 -> 13 -> 14 -> 9 -> 4 -> 0 -> 11 -> 5 -> 1 -> 7 -> 8 -> 15 -> 2. Programirajte memoriju tako da se citav sklop ponaša kao standardno binarno brojilo unaprijed. Uključenjem na napajanje brojilo B ulazi u stanje 10, a izlaz citavog sklopa treba poprimiti stanje 0 (prilikom očitavanja stanja Q3 tretirati kao bit najveće težine). Napomena: kao rješenje upišite sadržaj memorije po lokacijama, i to svaku lokaciju kao jednu heksadekadsku znamenku.

RJ.

0. – 15.: **8, B, F, 1, 7, A, 3, C, D, 6, 0, 9, 2, 4, 5, E**

6.

Potrebno je realizirati sinkrono brojilo koje broji u ciklusu: (6, 5, 3, 0, 1, 4, 7, 2, 6), ako je na raspolaganju 3 SR bistabila. Prilikom očitavanja stanja izlaz Q0 bistabila B0 promatrati kao bit najmanje težine. U polja za unos rješenja unijeti minimizirane algebarske oblike funkcija.

RJ.

S2: **NOT Q2 AND NOT Q1 AND Q0 OR Q1 AND NOT Q0**

R2: **Q2 AND Q0**

S1: **Q2 AND NOT Q1**

R1: **NOT Q2 AND Q0 OR Q2 AND Q1 AND NOT Q0**

S0: **NOT Q1 AND NOT Q0 OR Q2 AND NOT Q0**

R0: **NOT Q2 AND Q0 OR Q1 AND Q0**

7.

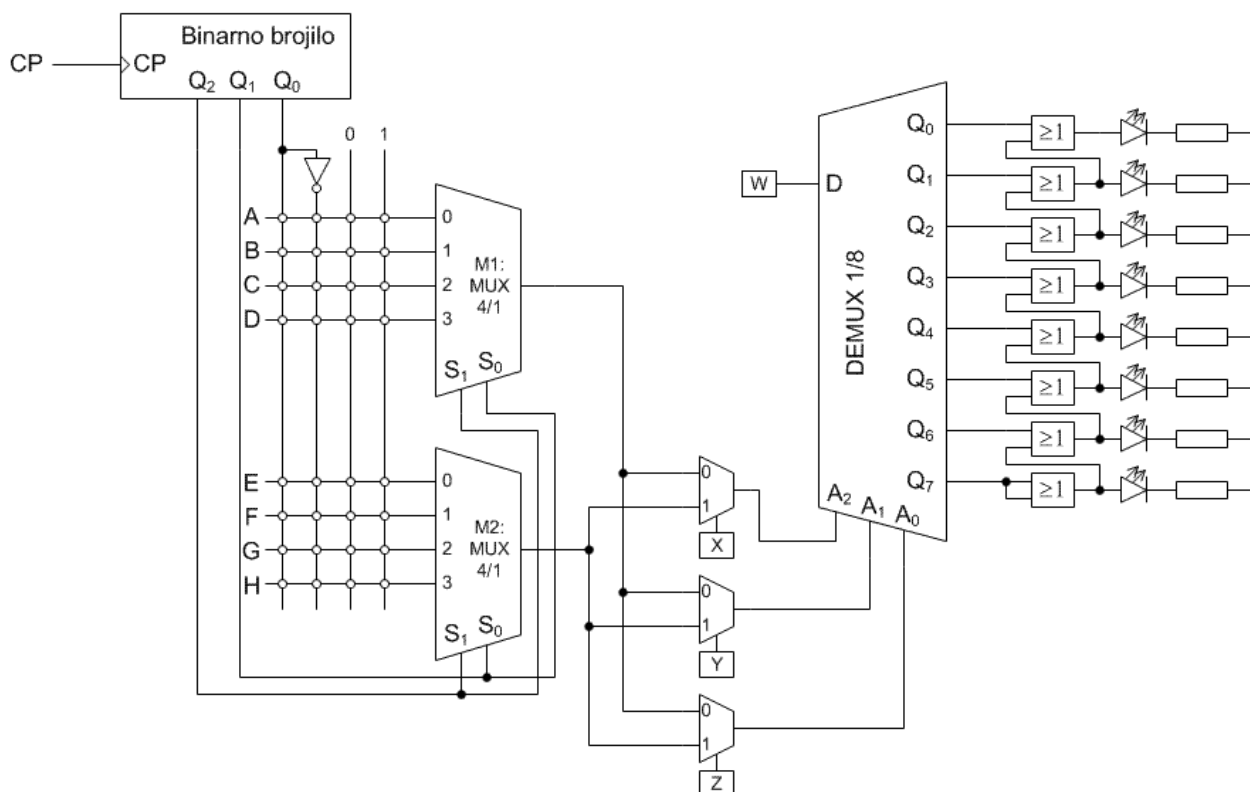
Za neki 7-bitni pretvornik sa sukcesivnom aproksimacijom poznato je da se ulazni napon iznosa 5.4V pretvara 31 ms. Koliko iznosi vrijeme pretvorbe za ulazni napon iznosa 10.0V? Podrazumijevana mjerna jedinica je ms.

RJ.

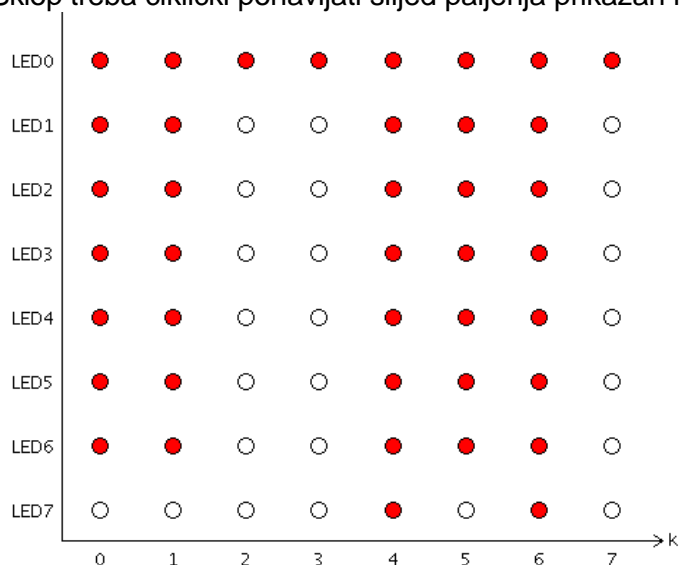
**31ms**

8.

Uporabom binarnog brojila želi se ostvariti sklop koji upravlja nizom svjetlećih dioda (LED), po principu termometarske skale. Sklop je prikazan u nastavku.



Sklop treba ciklicki ponavljati slijed paljenja prikazan na sljedećem dijagramu.



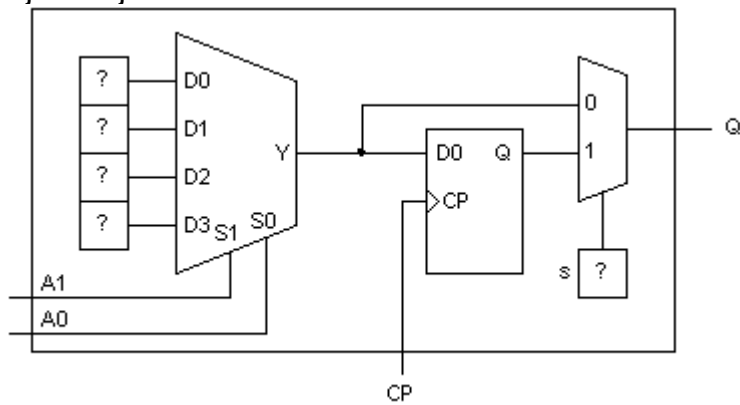
Po uključenju, binarno brojilo nalazi se u stanju 0, te niz dioda treba svijetliti kao na dijagramu u trenutku  $t=0$ . Napomena: A, B, C, D, E, F, G i H cine polje programirljivih sklopki. Kao rješenje se za svaki redak unosi 1 na mjestu ukljućene sklopke, odnosno 0 na mjestu iskljućene sklopke. Primjerice, ukoliko se na ulaz 0 multipleksora M1 (redak A) želi dovesti komplement od  $Q_0$ , kao rješenje ce se upisati 0100 (1., 3. i 4. sklopka su iskljućene, 2. je ukljućena). X, Y, Z i W su zastavice koje mogu poprimiti vrijednost 0 ili 1.

RJ. (8):

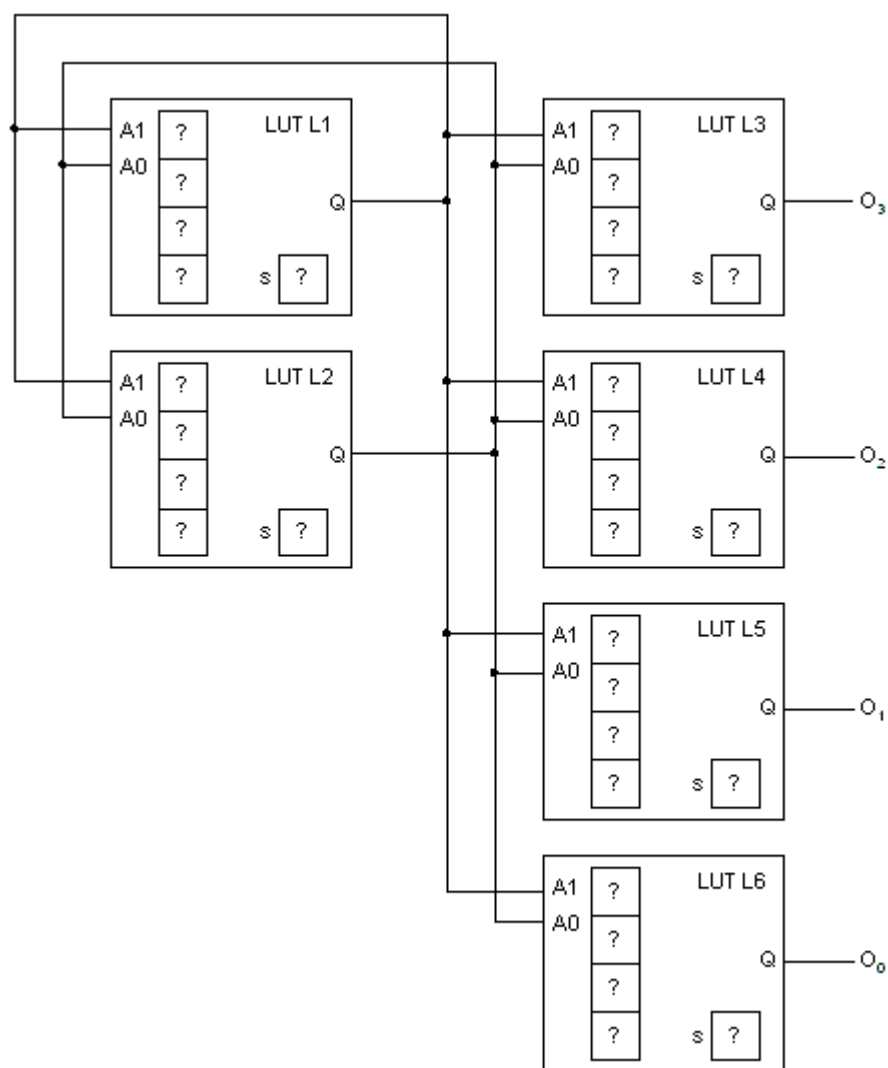
A: 0001  
B: 0010  
C: 0001  
D: 0100  
E: 0010  
F: 0010  
G: 0100  
H: 0100  
W: 1  
X: 0  
Y: 0  
Z: 1

9.

Logicki blok FPGA sklopa temeljen na LUT-u, bistabilu i multipleksoru prikazan je na sljedećoj slici.



Više takvih logičkih blokova povezano je u sklop prikazan u nastavku.



Programirajte sve logicke blokove tako da se dobije sklop koji na izlazu generira sekvencu:

7, 7, 0, 11.

Pri tome izlaz O3 tretirati kao izlaz najveće, a izlaz O0 kao izlaz najmanje težine.

RJ.

L1.lut: **0,1,1,0**

L1.s: **1**

L2.lut: **1,0,1,0**

L2.s: **1**

L3.lut: **0,0,0,1**

L3.s: **0**

L4.lut: **1,1,0,0**

L4.s: **0**

L5.lut: **1,1,0,0**

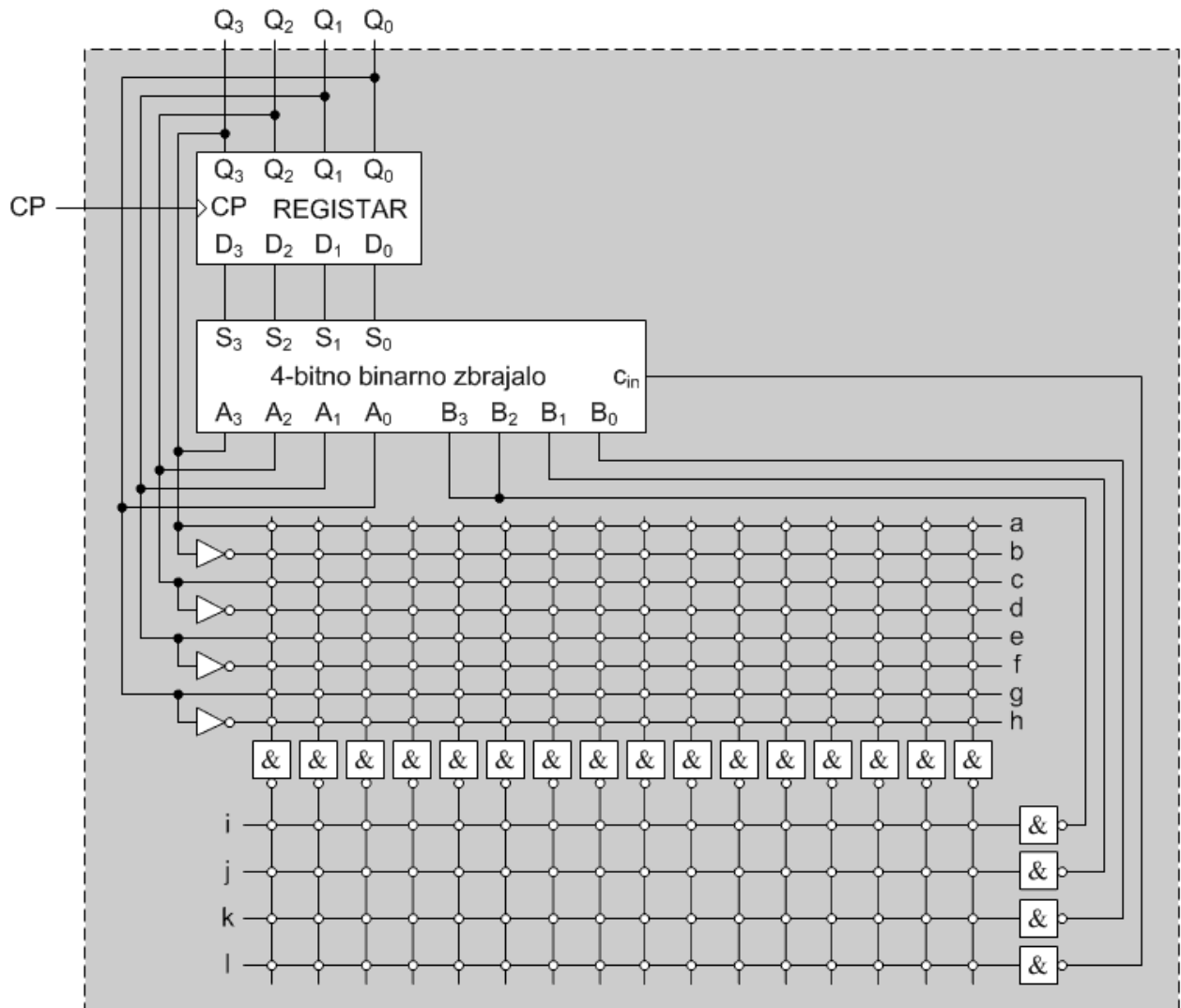
L5.s: **0**

L6.lut: **1,1,0,0**

L6.s: **0**

10.

Uporabom registra, binarnog zbrajala i sklopa PLA želi se ostvariti brojilo koje broji u ciklusu 0 -> 14 -> 13 -> 15 -> 12 -> 9 -> 11 -> 10 -> 8 -> 6 -> 7 -> 5 -> 4 -> 1 -> 3 -> 2. Sklop je prikazan u nastavku.



Programirajte sklop PLA tako da ostvarite zadano brojilo. Napomena: a, b, c, d, e, f, g, h, i, j, k, l cine polje programirljivih sklopki. Kao rješenje se za svaki redak unosi 1 na mjestu ukljucene sklopke, odnosno 0 na mjestu iskljucene sklopke. Primjerice, ukoliko prvi i drugi NI sklop koriste  $Q_3$ , tada ce redak 'a' biti 1100000000000000. Uocite kako konfiguracija svakog retka ima točno 16 znamenki.



RJ.

a	0000000011111111
b	1111111100000000
c	0000111100001111
d	1111000011110000
e	0011001100110011
f	1100110011001100
g	0101010101010101
h	1010101010101010
i	1011110110111011
j	0101010001010110
k	1010001110100000
l	1011110110111011

**\*\*\*RIJEŠENJA SU 100% TOCNA;  
ALI NE SADRŽE I POSTUPKE\*\*\***