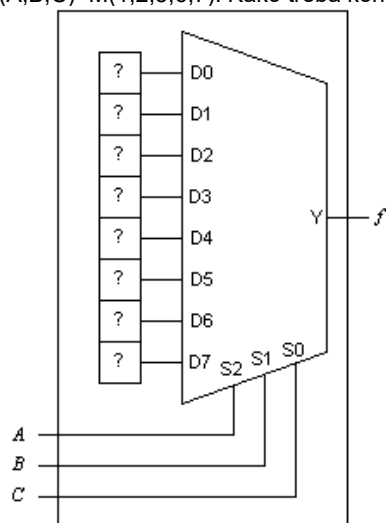


## 5. domaća zadaća (2021Zima)

1. **Točno**

Relativni doprinos: 1.0/1.0

Konfigurabilni logički blok (CLB) sklopa FPGA ostvaren je uporabom pregledne tablice (LUT) s 3 ulaza, kako je prikazano slikom. Uporabom takvog CLB-a potrebno je realizirati funkciju  $f$  zadanu kao produkt maksterma:  $f(A,B,C)=M(1,2,5,6,7)$ . Kako treba konfigurirati LUT?



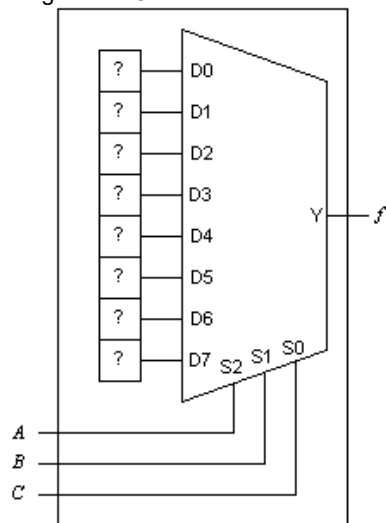
D0	<input type="text" value="1"/>
D1	<input type="text" value="0"/>
D2	<input type="text" value="0"/>
D3	<input type="text" value="1"/>
D4	<input type="text" value="1"/>
D5	<input type="text" value="0"/>
D6	<input type="text" value="0"/>
D7	<input type="text" value="0"/>

Važna napomena: U svako polje za unos treba unijeti samo vrijednost pripadnog ulaza luta za navedenu funkciju. Dozvoljena su dva načina upisa vrijednosti: 1 se tretira jednako kao i true, a 0 je ekvivalentna sa false.

2. **Točno**

Relativni doprinos: 1.0/1.0

Konfigurabilni logički blok (CLB) sklopa FPGA ostvaren je uporabom pregledne tablice (LUT) s 3 ulaza, kako je prikazano slikom. Uporabom takvog CLB-a potrebno je realizirati funkciju:  $f(A,B,C)=((B \text{ AND } B) \text{ AND } C)$ . Kako treba konfigurirati LUT?



D0

D1

D2

D3

D4

D5

D6

D7

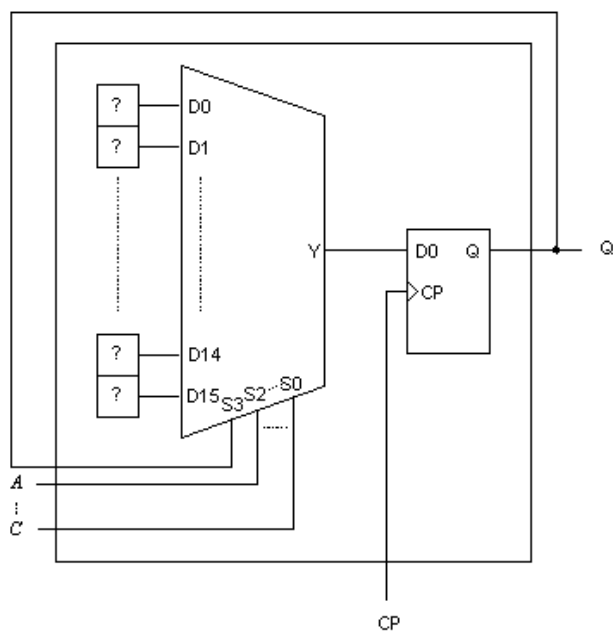
Važna napomena: U svako polje za unos treba unijeti samo vrijednost pripadnog ulaza luta za navedenu funkciju. Dozvoljena su dva načina upisa vrijednosti: 1 se tretira jednako kao i true, a 0 je ekvivalentna sa false.

3. **Točno**

Relativni doprinos: 1.0/1.0

Na raspolaganju je 4-ulazni LUT s D bistabilom, prikazan slikom. Programirati taj bistabil tako da se dobije bistabil čija je funkcija opisana sljedećom tablicom.

A	B	C	Qn+1
0	0	0	1
0	0	1	Qn
0	1	0	not Qn
0	1	1	not Qn
1	0	0	0
1	0	1	Qn
1	1	0	not Qn
1	1	1	0



LUT\_0

LUT\_1

LUT\_2

LUT\_3

LUT\_4

LUT\_5

LUT\_6

LUT\_7

LUT\_8

LUT\_9

LUT\_10

LUT\_11

LUT\_12

LUT\_13

LUT\_14

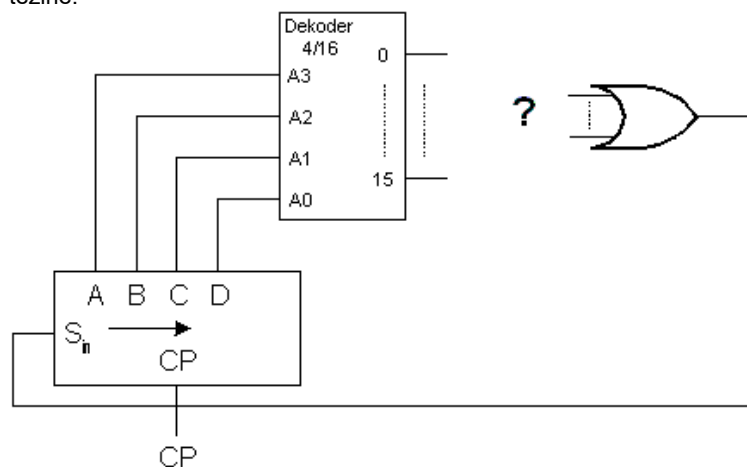
LUT\_15

Važna napomena: U svako polje za unos treba unijeti samo vrijednost pripadnog ulaza luta za navedenu funkciju. Dozvoljena su dva načina upisa vrijednosti: 1 se tretira jednako kao i true, a 0 je ekvivalentna sa false. Bilo koji oblik redundantnog unosa (dupliciranje nula, proširivanje jedinice sa vodećom nulom) povlači netočnost unosa.

#### 4. Djelomično točno **Vaše rješenje** | **Točno rješenje**

Relativni doprinos: 0.72727272727273/1.0

Potrebno je projektirati sklop koji će prolaziti kroz sljedeća stanja: (7, 3, 1, 0, 8, 4, 10, 13, 14, 15, 7). Sklop je potrebno ostvariti uporabom strukture prikazane na slici (posmačni registar + dekodler 4/16). Nespecificirana stanja treba tako riješiti da sklop najbrže stigne u stanje 15. Koje sve izlaze treba spojiti na logički sklop ili ? U polje za unos odgovora je potrebno unijeti indekse izlaza dekodera (vidi sliku), odijeljene zarezima, koje je potrebno povezati na ulaze logičkog sklopa da bi se ostvarilo zadano brojilo. Prilikom očitavanja stanja, izlaz A tretirati kao bit najveće težine.



izlazi  
dekodera

Važna napomena: U polje za unos treba unijeti samo numeričke vrijednosti indeksa izlaza dekodera koje je potrebno povezati na ulaze logičkog sklopa da bi se ostvarilo zadano brojilo. npr: 0,1,8,15,16,22,23,26,27 .

#### 5. **Točno**

Relativni doprinos: 1.0/1.0

Funkcija 4 bitnih registara prikazanih na slici definirana je sljedećom tablicom.

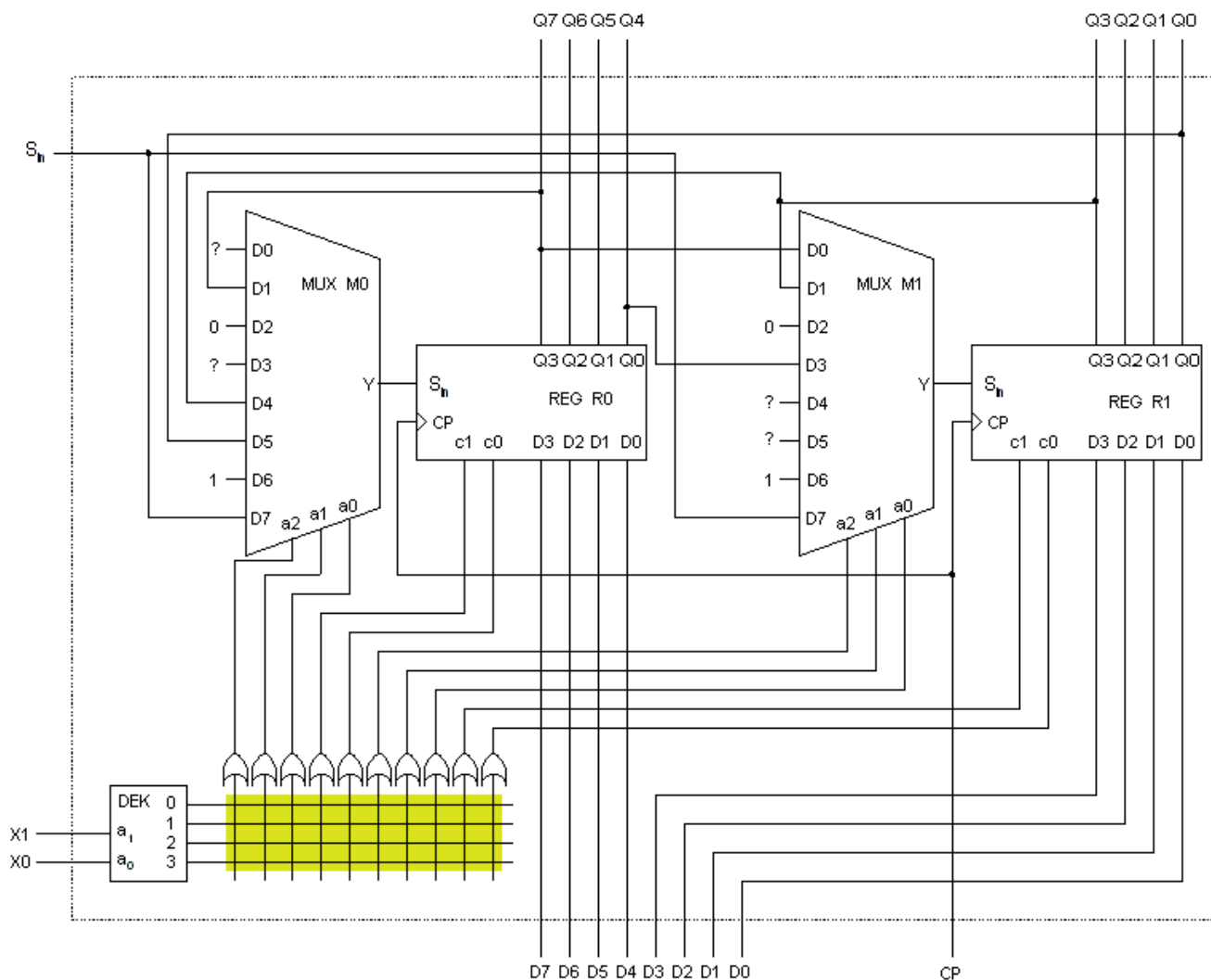
C1	C0	Opis
0	0	NOP - nema nikakve promjene
0	1	Posmak u desno, punjenje sa Sin
1	0	Posmak u lijevo uz čuvanje bita predznaka, punjenje sa Sin
1	1	Posmak u lijevo, punjenje sa Sin

Uporabom multipleksora i dekodera s programirljivom ILI ravninom ostvarena je struktura koja se može programirati tako da se dobije 8-bitni registar s mnoštvom podržanih operacija. Vaš je zadatak programirati ovu strukturu tako da se dobije jedan 8-bitni registar čija je funkcija određena sljedećom tablicom.

X1	X0	Opis
----	----	------

0	0	Posmak u lijevo, punjenje s 1
0	1	Posmak u lijevo, punjenje s 0
1	0	Posmak u lijevo, punjenje sa Sin
1	1	Aritmetički posmak u desno

Ulaze u multiplexsor označene upitnikom zabranjeno je koristiti. U polja za unos rješenja u nastavku potrebno je unijeti zarezima odvojen popis izlaza dekodera koje je potrebno spojiti na odgovarajući ILI sklop, kako bi se ostvarila potrebna funkcija. Ako se za neki ILI sklop ne definira niti jedan izlaz dekodera, taj ILI sklop na izlazu generira logičku nulu.



M0.a2

M0.a1

M0.a0

R0.c1

R0.c0

M1.a2

M1.a1

M1.a0

R1.c1

R1.c0

6. **Točno**

Relativni doprinos: 1.0/1.0

Potrebno je realizirati sinkrono brojiilo koje broji u ciklusu: (4, 7, 1, 3, 0, 5, 2, 6, 4), 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.

S2

R2

S1

R1

S0

R0

Važna napomena: U polja za unos rješenja treba unijeti logičke izraze funkcija dovedenih na ulaze bistabila, uz koje će se ostvariti zadano brojilo.

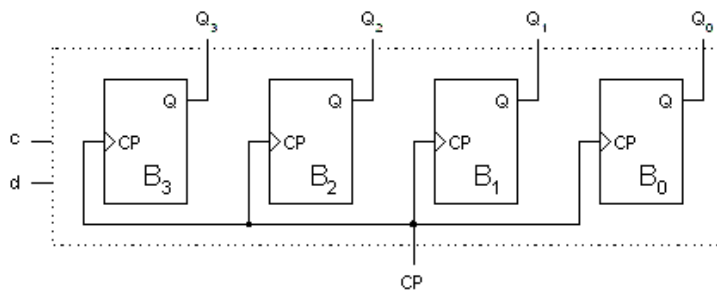
## 7. Točno

Relativni doprinos: 1.0/1.0

Uporabom D bistabila realizirati 4-bitno brojilo koji broji ovisno o signalu d: ako je  $d=1$ , tad je sljedeće stanje = trenutno stanje-3, inače sljedeće stanje = trenutno stanje+2 (pod pojmom stanje misli se na binarno kodirani broj zapisan kroz bistabile, pri čemu je izlaz Q0 izlaz najmanje težine). Sklop treba imati i sinkroni ulaz za brisanje c (kojeg bistabili nemaju). Koristiti minimalni broj osnovnih logičkih 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:

$c \text{ and } Q2 \text{ and not } Q1 \text{ or not } c \text{ and } d \text{ and not } Q0$ .



B3.D

B2.D

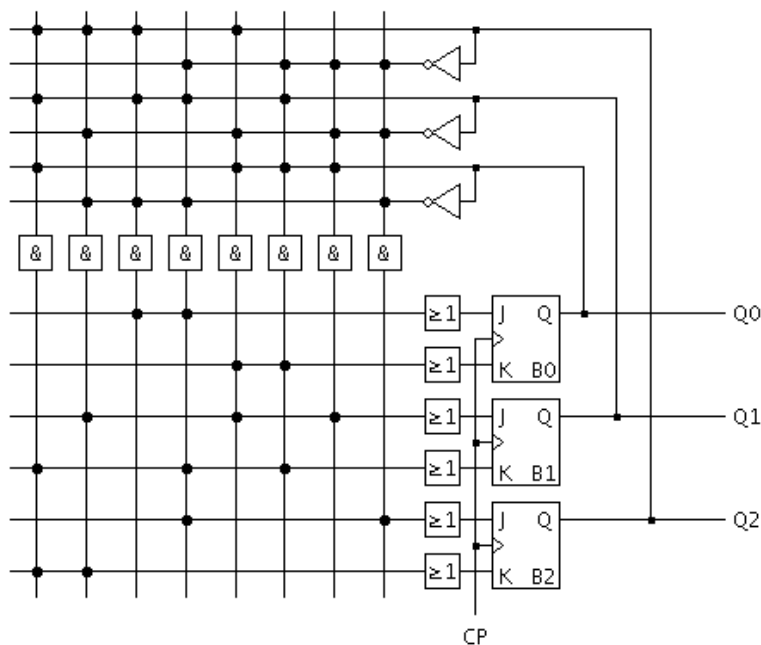
B1.D

B0.D

## 8. Točno

Relativni doprinos: 1.0/1.0

Neko sinkrono brojilo prikazano je sljedećom slikom.



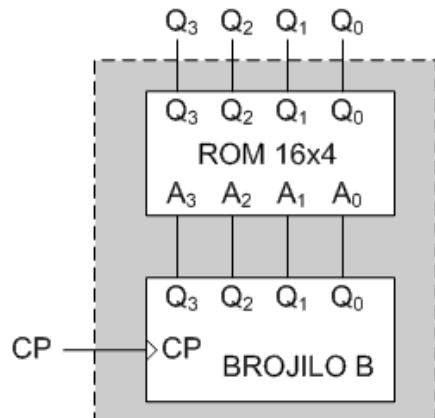
U kojem ciklusu broji to brojilo? Prilikom očitavanja stanja izlaz Q0 tretirati kao bit najmanje težine.

- ☒ (6, 7, 1, 3, 0, 4, 2, 5, 6)
- ☐ (6, 1, 5, 7, 2, 4, 0, 3, 6)
- ☐ (2, 1, 5, 7, 4, 3, 6, 0, 2)
- ☐ (5, 7, 6, 1, 3, 0, 4, 2, 5)

### 9. Točno

Relativni doprinos: 1.0/1.0

Na raspolaganju je sklop prikazan na slici.



Brojilo B koje je iskorišteno za brojanje broji u ciklusu 6 -> 8 -> 13 -> 4 -> 5 -> 3 -> 14 -> 10 -> 2 -> 7 -> 12 -> 15 -> 9 -> 0 -> 1 -> 11. Programirajte memoriju tako da se čitav sklop ponaša kao standardno binarno brojilo unaprijed.

Uključenjem na napajanje brojilo B ulazi u stanje 6, a izlaz čitavog 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.

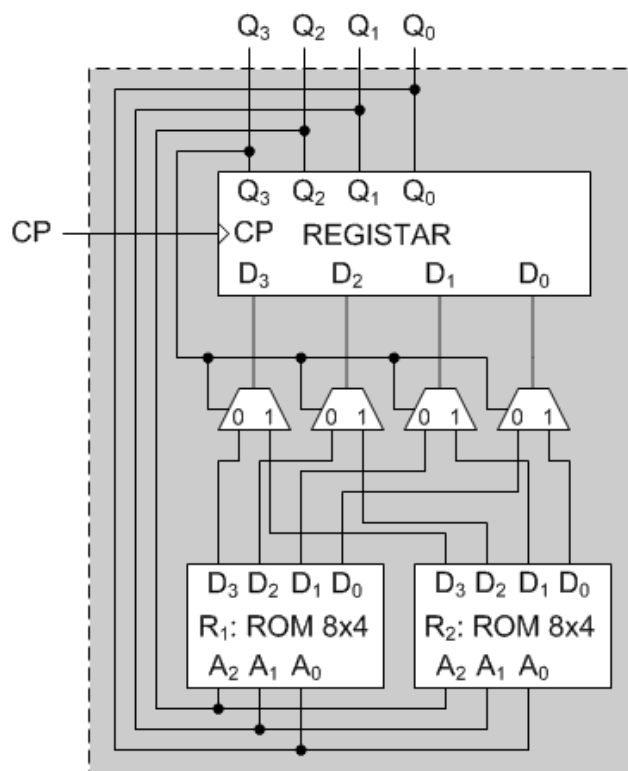
0.
1.
2.
3.
4.
5.
6.
7.

8.
9.
10.
11.
12.
13.
14.
15.

10. **Točno**

Relativni doprinos: 1.0/1.0

Na raspolaganju je sklop prikazan na slici.



Memorije R1 i R2 potrebno je programirati tako da se dobije brojiilo koje broji u ciklusu 12 -> 15 -> 7 -> 11 -> 13 -> 2 -> 10 -> 3 -> 8 -> 9 -> 5 -> 14 -> 0 -> 4 -> 1 -> 6 (prilikom očitavanja stanja Q3 tretirati kao bit najveće težine). Napomena: kao rješenje upišite sadržaj memorija po lokacijama, i to svaku lokaciju kao jednu heksadekadsku znamenku.

- R1:
0.
- R1:
1.
- R1:
2.
- R1:
3.
- R1:
4.
- R1:
5.
- R1:
6.
- R1:
7.

R2: 0.

R2: 1.

R2: 2.

R2: 3.

R2: 4.

R2: 5.

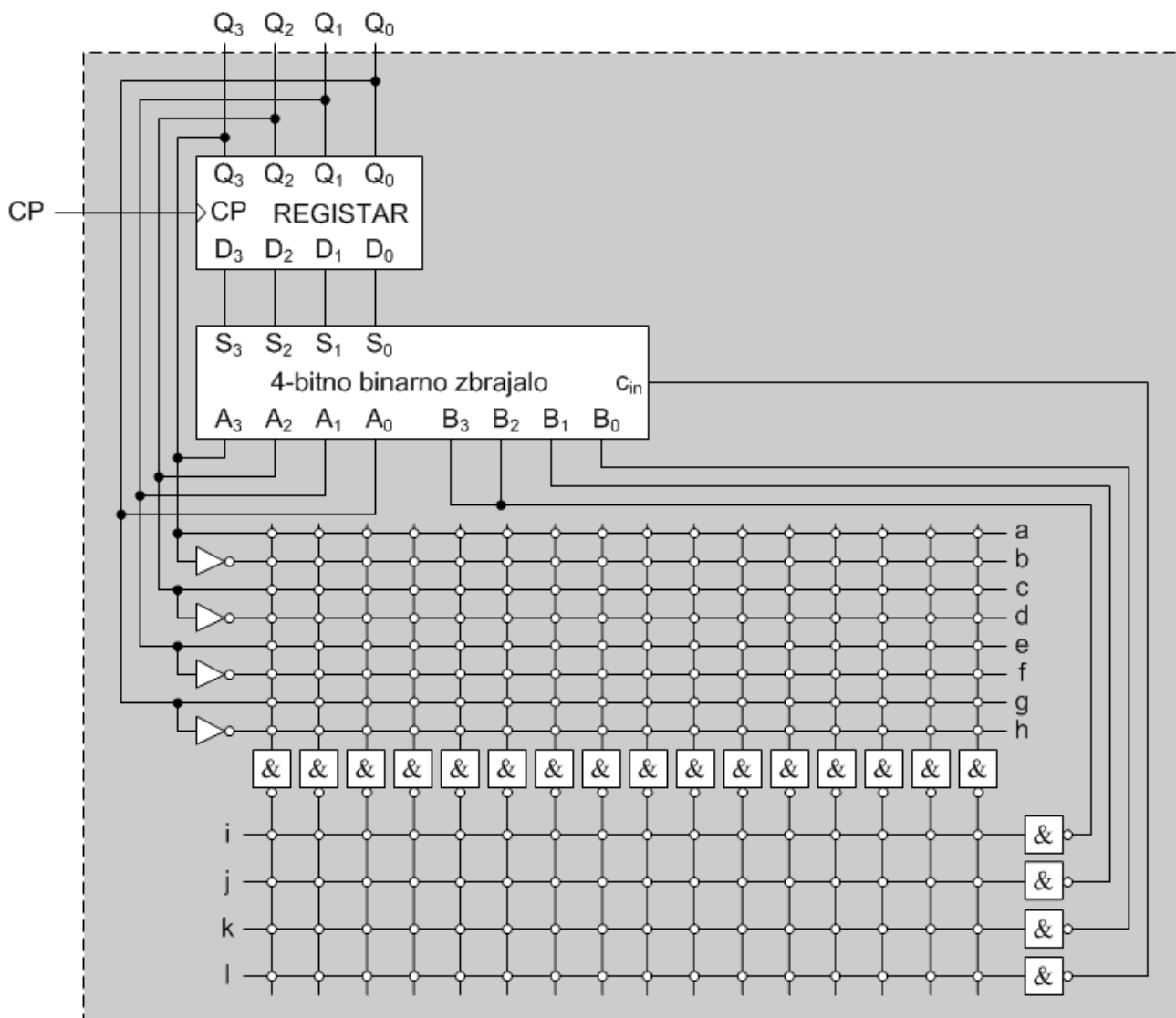
R2: 6.

R2: 7.

# 11. Nije riješen **Vaše rješenje** | **Točno rješenje**

Relativni doprinos: 0.0/1.0

Uporabom registra, binarnog zbrajala i sklopa PLA želi se ostvariti brojilo koje broji u ciklusu 0 -> 15 -> 14 -> 13 -> 12 -> 9 -> 11 -> 10 -> 8 -> 5 -> 7 -> 6 -> 4 -> 3 -> 2 -> 1. 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 čine polje programiranih sklopki. Kao rješenje se za svaki redak unosi 1 na mjestu uključene sklopke, odnosno 0 na mjestu isključene sklopke. Primjerice, ukoliko prvi i drugi NI sklop koriste Q<sub>3</sub>, tada će redak 'a' biti 1100000000000000. Uočite kako konfiguracija svakog retka ima točno 16 znamenki.

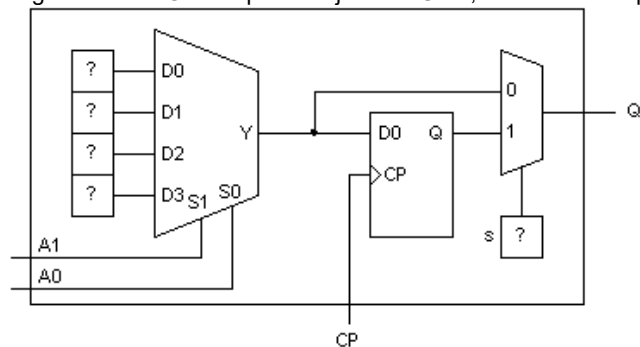


- a 0000000011111111  
b 1111111100000000  
c 0000111100001111  
d 1111000011110000  
e 0011001100110011  
f 1100110011001100  
g 0101010101010101  
h 1010101010101010  
i 1111101110111111  
j 1111110101010111  
k 0000001000100000  
l 1111101110111111

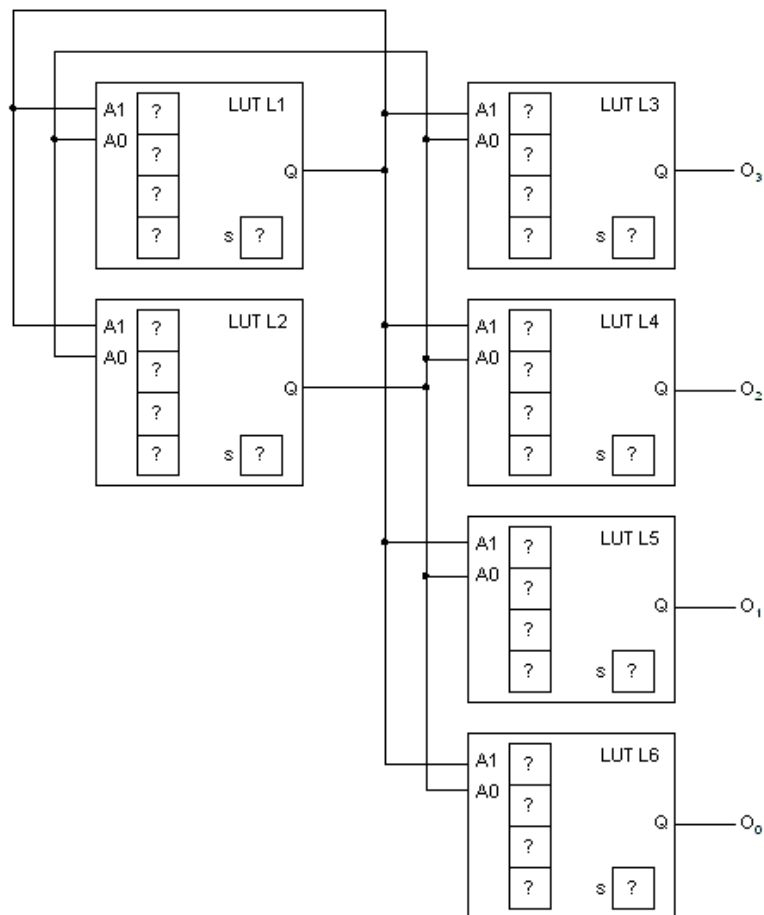
12. **Djelomično točno** Vaše rješenje | **Točno rješenje**

Relativni doprinos: 0.5/1.0

Logički 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 logičke blokove tako da se dobije sklop koji na izlazu generira sekvencu: 3, 3, 3, 15.

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

L1.lut

L1.s

L2.lut

L2.s

L3.lut

L3.s

L4.lut

L4.s

L5.lut

L5.s

L6.lut

L6.s

Važna napomena: U polja za unos u koje treba unijeti vrijednost pregledne tablice (Li.lut) unosi se popis zarezima odvojenih vrijednosti. Pri tome se najprije unosi vrijednost koja odgovara nultoj lokaciji pregledne tablice. U polja za unos u koje treba unijeti vrijednost adresnog ulaza multipleksora (Li.s) unosi se jedan broj.

13. **Točno**

Relativni doprinos: 1.0/1.0

Za neki 3-bitni pretvornik sa sukcesivnom aproksimacijom poznato je da se ulazni napon iznosa 11.4V pretvara 77 ms. Koliko iznosi vrijeme pretvorbe za ulazni napon iznosa 5.8V? Podrazumijevana mjerna jedinica je ms.

[Povratak](#)