

RAZVOJ DIGITALNIH SISTEMOV

Izpit 29. 01. 2016

1. Realizirajte funkcijo f s čim manj izbiralniki 4/1.

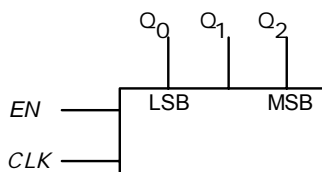
$$f(a,b,c,d) = (a \cdot \bar{b} + b \cdot c \cdot \bar{d} + b \cdot c) \cdot ((a \cdot c \cdot d) \cdot (\bar{c} + d))$$

2. Uporabite PAL3L3 (namišljen čip) za realizacijo naslednjih funkcij:

- $f_1 = x_1 \oplus x_2$
- $f_2 = x_1 \cdot x_2 \cdot x_3$
- f_3 = funkcijo treh spremenljivk, ki vrne '1' pri vsaj dveh enicah na vseh vseh.

Vezje ima 3 vhode in 3 izhode. Vsaka disjunkcija (OR) ima 4 konjunkcije (AND). Oznaka L pomeni, da je izhod negiran. Programirane povezave označite s piko. Vezje PAL je narisano na hrbtni strani izpita.

3. Prikažite sintezo 3-bitnega sinhronnega števca navzgor po Grayevi kodi s T flip-flopi in logičnimi vrati. Števec ima 3-bitni števeni izhod (Q_2, Q_1, Q_0), vhod za signal ure (CLK) in signal za omogočanje štetja (EN). Števec stoji, ko je EN='0' in šteje, ko je EN='1'. Uporabite poimenovanje signalov, kot je narisano na spodnji sliki.



4. Načrtajte diagram stanj Moore-ovega avtomata končnih stanj, ki krmili delovanje garažnih vrat: Garažna vrata imajo vhod VRATA ter vhod ZAŠČITA, ki postane '1' vedno, ko preko motorja steče dovolj velik tok. Z meritvijo toka na motorju obenem izdelamo funkcijo detekcije obeh končnih položajev, kot tudi zaščito proti oviram na poti vrat. Vezje ima 2-bitni izhod za enosmerni motor:

Koda operacije		Funkcija izhoda
OP ₁	OP ₀	
0	0	motor stoji
0	1	motor pomika vrata navzgor
1	0	motor pomika vrata navzdol

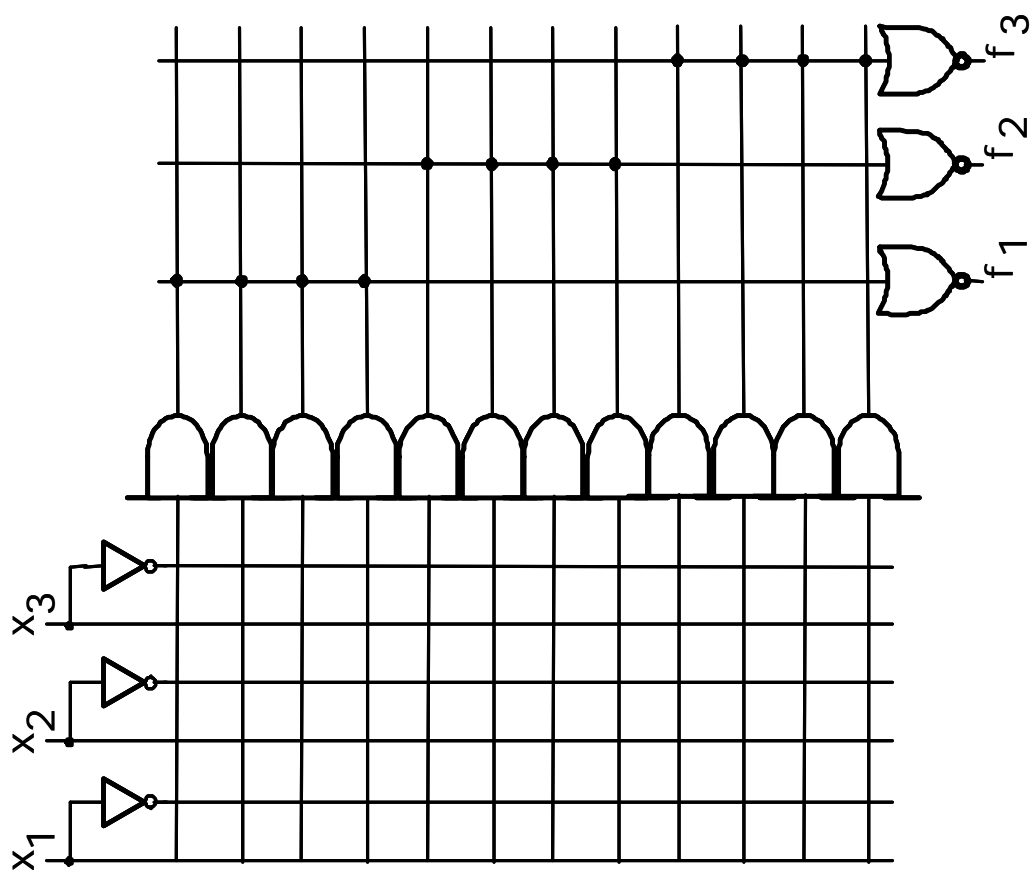
Če pritisnemo gumb VRATA, se vrata začno pomikati navzgor. Če na poti naletijo na oviro ali pridejo do zgornje končne lege, se motor ustavi. Če pritisnemo gumb VRATA ponovno, se začnejo gibati v smeri navzdol.

Podobno je v obratni smeri: Če na poti naletijo na oviro ali pridejo do spodnje končne lege, se motor ustavi. Če pritisnemo gumb VRATA ponovno, se začnejo pomikati v smeri navzgor.

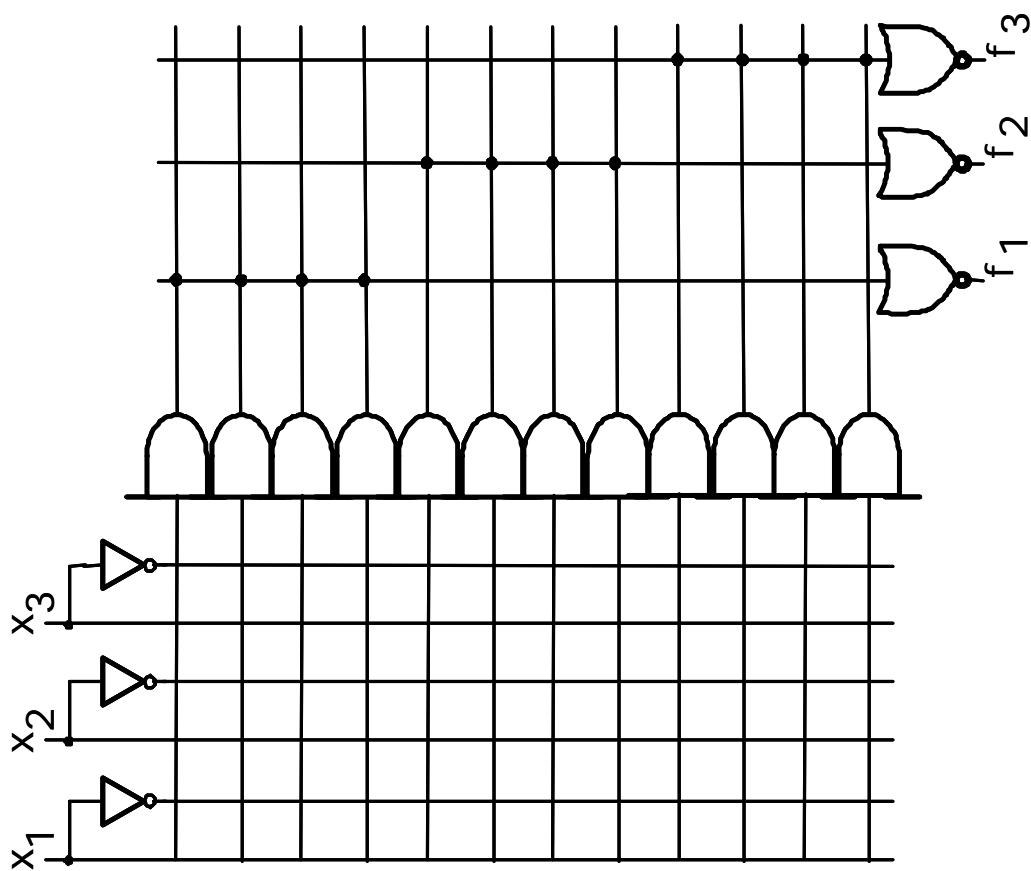
Čas pisanja je 60 minut. Vsaka naloga je vredna 10 točk.

Na list z rešitvami se podpišite in napišite še vpisno številko ter kateri predmet pišete.

Rezultati bodo objavljeni v sistemu STUDIS.



Če se zmotite, prečrtajte napačno shemo in uporabite drugo shemo!



Čas pisanja je 60 minut. Vsaka naloga je vredna 10 točk.

Na list z rešitvami se podpišite in napišite še vpisno številko ter kateri predmet pišete.

Rezultati bodo objavljeni v sistemu STUDIS.

Rešitev 1. naloge:

Funkcija f je podana v večnivojski (nenormalni) obliki:

$$f(a,b,c,d) = (a \cdot \bar{b} + b \cdot c \cdot \bar{d} + b \cdot c) \cdot ((a \cdot c \cdot d) \cdot (\bar{c} + d))$$

zato jo najprej poenostavimo z uporabo pravil Boole-ove logike. Izpišemo desni člen funkcije in uporabimo lastnost Boole-ove logike $x \cdot \bar{x} = 0$, lastnost $x \cdot x = x$ in lastnost $0 + x = x$.

$$f(a,b,c,d) = (a \cdot \bar{b} + b \cdot c \cdot \bar{d} + b \cdot c) \cdot (a \cdot c \cdot d \cdot \bar{c} + a \cdot c \cdot d \cdot d)$$

Nad rezultatom ponovno uporabimo lastnost Boole-ove logike $x \cdot \bar{x} = 0$ in lastnost $x \cdot x = x$.

$$f(a,b,c,d) = (a \cdot \bar{b} + b \cdot c \cdot \bar{d} + b \cdot c) \cdot (a \cdot c \cdot d)$$

Rezultat vnesemo v levi del funkcije in znova uporabimo omenjene lastnosti Boole-ove logike:

$$f(a,b,c,d) = (a \cdot \bar{b} \cdot a \cdot c \cdot d + b \cdot c \cdot \bar{d} \cdot a \cdot c \cdot d + b \cdot c \cdot a \cdot c \cdot d)$$

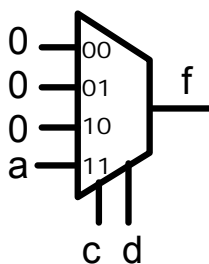
Dobimo dva člena in ju zapišemo v obliki PDNO, ki jo nato minimiziramo s pomočjo Veitch-evega diagrama ali z uporabo lastnosti združevanja Boole-ove algebre $x + \bar{x} = 1$:

$$f(a,b,c,d) = a \cdot \bar{b} \cdot c \cdot d + b \cdot a \cdot c \cdot d$$

$$f_{PDNO}(a,b,c,d) = V(11,15)$$

$$f_{MDNO}(a,b,c,d) = a \cdot c \cdot d \cdot (\bar{b} + b) = a \cdot c \cdot d$$

in jo realiziramo z enim izbiralnikom 4/1, tako da naredimo Shannon-ov razvoj funkcije. Glede na kombinacijo naslovnih vhodov izbiralnika dobimo 6 možnih rešitev (ac, ca, ad, da, cd, dc).



Čas pisanja je 60 minut. Vsaka naloga je vredna 10 točk.

Na list z rešitvami se podpišite in napišite še vpisno številko ter kateri predmet pišete.

Rezultati bodo objavljeni v sistemu STUDIS.

Rešitev 2. naloge:

Za funkcije zapišemo najprej pravilnostno tabelo, nato narišemo Veitch–eve diagrame.

x_1	x_2	x_3	f_1	f_2	f_3
0	0	0	0	0	0
0	0	1	0	0	0
0	1	0	1	0	0
0	1	1	1	0	1
1	0	0	1	0	0
1	0	1	1	0	1
1	1	0	0	0	1
1	1	1	0	1	1

Vezje PAL ima negirane izhode, zato bomo pri realizaciji funkcij z Veitch–evimi diagrami realizirali \overline{f} in ne f .

$\overline{f_1}$:

	x_1		
x_2	0	0	1
	1	1	0
		x_3	

Prvo funkcijo zapišemo enostavno, saj je negacija XOR funkcije dveh spremenljivk kar funkcija ekvivalence:

$$\overline{f_1} = x_1 \cdot x_2 + \overline{x_1} \cdot \overline{x_2}$$

Podobno lahko naredimo za drugo funkcijo, kjer za negacijo konjunkcije treh spremenljivk uporabimo De Morgan–ovo enakost.

$\overline{f_2}$:

	x_1		
x_2	0	1	0
	0	0	0
		x_3	

$$\overline{f_2} = \overline{x_1 \cdot x_2 \cdot x_3}$$

$$\overline{f_2} = \overline{x_1} + \overline{x_2} + \overline{x_3}$$

Zadnjo funkcijo minimiziramo z uporabo Veitch–evega diagrama, tako da zbiramo ničle.

$\overline{f_3}$:

	x_1		
x_2	1	1	1
	0	1	0
		x_3	

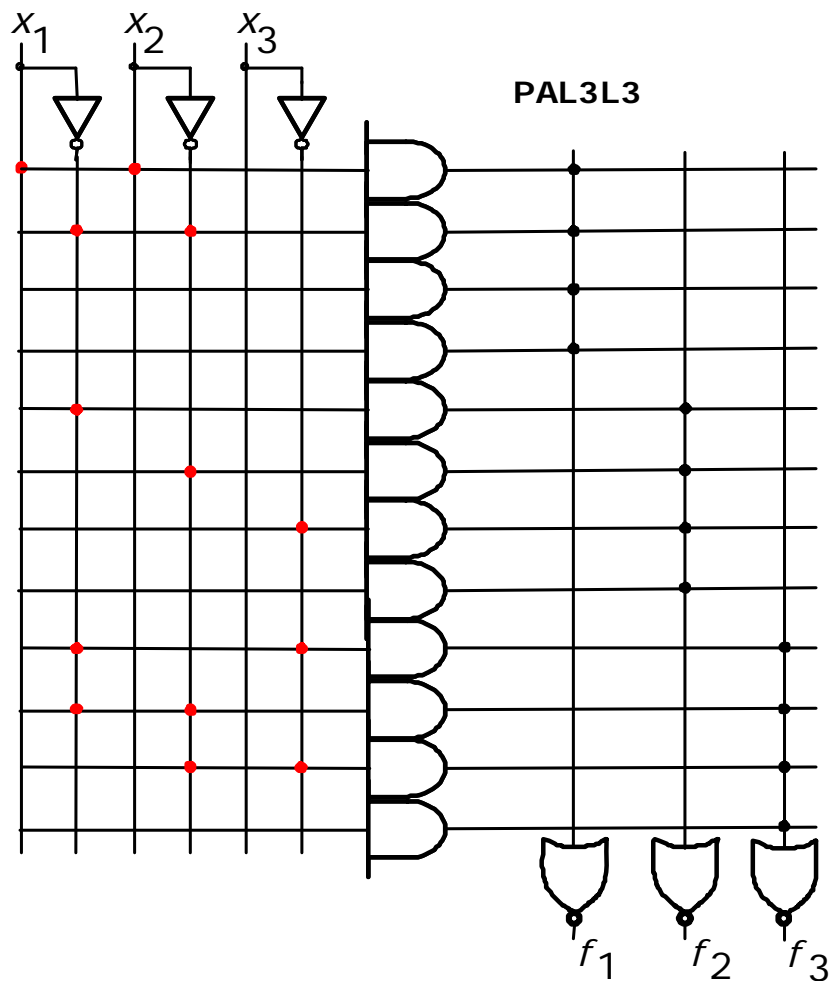
$$\overline{f_3} = \overline{x_1 \cdot x_3} + \overline{x_1 \cdot x_2} + \overline{x_2 \cdot x_3}$$

Pri realizaciji PAL vezja upoštevamo poenostavljeno strukturo, pri kateri ne vezemo vsake povezave na konjunkcije, saj so AND vrata na narisani strukturi 6–vhodna. Vezje PAL3L3 je AND–NOR arhitekture in vsebuje 4 konjunkcije na en NOR člen. Pri PAL vezju je programabilen samo AND del vezja.

Čas pisanja je 60 minut. Vsaka naloga je vredna 10 točk.

Na list z rešitvami se podpišite in napišite še vpisno številko ter kateri predmet pišete.

Rezultati bodo objavljeni v sistemu STUDIS.



Predstavljeno PAL3L3 vezje je sicer izmišljeno, vendar demonstrira strukturo in uporabo večjih (realnih) PAL vezij kot so npr. PAL14L4, GAL16V8 in GAL22V10. GAL vezja so nadgradnja osnovne PAL strukture. Slednji so izključno kombinacijski, GAL vezja pa imajo v OLMC (ang. Output Logic MacroCell) strukturi še D-FF, s katerim lahko realiziramo tudi sekvenčna vezja.

Čas pisanja je 60 minut. Vsaka naloga je vredna 10 točk.

Na list z rešitvami se podpišite in napišite še vpisno številko ter kateri predmet pišete (VSŠ, UNI).

Rezultati bodo objavljeni na: <https://estudent.fri.uni-lj.si>

Rešitev 3. naloge:

Postopek sinteze zahteva, da zapišemo tabelo prehajanja stanj števca navzgor po Gray-evi kodi. Desetiška števna sekvenca po 3-bitni Grayevi kodi se glasi:
 ... 0, 1, 3, 2, 6, 7, 5, 4, 0...

Števno sekvenco zapišemo v tabelo:

trenutno stanje			naslednje stanje			enačbe T-FF		
Q ₂	Q ₁	Q ₀	Q ₂	Q ₁	Q ₀	T ₂	T ₁	T ₀
0	0	0	0	0	1	0	0	1
0	0	1	0	1	1	0	1	0
0	1	0	1	1	0	1	0	0
0	1	1	0	1	0	0	0	1
1	0	0	0	0	0	1	0	0
1	0	1	1	0	0	0	0	1
1	1	0	1	1	1	0	0	1
1	1	1	1	0	1	0	1	0

Iz tabele prehajanja stanj števca določimo enačbe T-FF:

Za T₀ narišemo Veitchev diagram. Funkcija je funkcija linearna, zato jo bomo izrazili z XOR operacijami.

		Q ₂	
T ₀	Q ₁	1	0
	Q ₀	0	1

$$T_0 = \overline{Q_2} \cdot \overline{Q_1} \cdot \overline{Q_0} + \overline{Q_2} \cdot Q_1 \cdot Q_0 + Q_2 \cdot \overline{Q_1} \cdot Q_0 + Q_2 \cdot Q_1 \cdot \overline{Q_0}$$

$$T_0 = \overline{Q_2} \cdot (\overline{Q_1} \cdot \overline{Q_0} + Q_1 \cdot Q_0) + Q_2 \cdot (\overline{Q_1} \cdot Q_0 + Q_1 \cdot \overline{Q_0})$$

$$T_0 = \overline{Q_2} \cdot (\overline{Q_1} \oplus Q_0) + Q_2 \cdot (Q_1 \oplus Q_0)$$

Uvedemo novo spremenljivko x:

$$x = Q_1 \oplus Q_0$$

in jo vstavimo v izraz za T₀:

$$T_0 = \overline{Q_2} \cdot \overline{x} + Q_2 \cdot x = \overline{Q_2} \oplus x$$

$$T_0 = 1 \oplus Q_2 \oplus Q_1 \oplus Q_0$$

Podobno za T₁ narišemo Veitchev diagram:

		Q ₂	
T ₁	Q ₁	0	1
	Q ₀	0	1

Iz diagrama za T₁ sledi:

$$T_1 = Q_2 \cdot Q_1 \cdot Q_0 + \overline{Q_2} \cdot \overline{Q_1} \cdot Q_0$$

$$T_1 = (Q_2 \cdot Q_1 + \overline{Q_2} \cdot \overline{Q_1}) \cdot Q_0$$

Operacija v oklepajih je ekvivalenca, zato enačbo lahko poenostavimo v:

$$T_1 = (\overline{Q_2} \oplus Q_1) \cdot Q_0$$

Podobno storimo še za T₂:

		Q ₂	
T ₂	Q ₁	0	0
	Q ₀	1	0

Iz diagrama za T₂ sledi:

$$T_2 = \overline{Q_2} \cdot Q_1 \cdot \overline{Q_0} + Q_2 \cdot \overline{Q_1} \cdot \overline{Q_0}$$

$$T_2 = (\overline{Q_2} \cdot Q_1 + Q_2 \cdot \overline{Q_1}) \cdot \overline{Q_0}$$

$$T_2 = (Q_2 \oplus Q_1) \cdot \overline{Q_0}$$

Operacija v oklepajih je negacija XOR, zato enačbo lahko poenostavimo v:

$$T_2 = (Q_2 \oplus Q_1) \cdot \overline{Q_0}$$

Manj potratno možnost realizacije predstavlja dvojiški 3-bitni sinhroni števec navzgor. Tak števec realiziramo s tremi T-FF in enim AND vrati.

Nastalemu sinhronemu števcu na izhodu dodamo pretvornik kode iz dvojiškega v Gray-evo kodo z XOR vrati po enačbah:

$$G_{MSB} = B_{MSB}$$

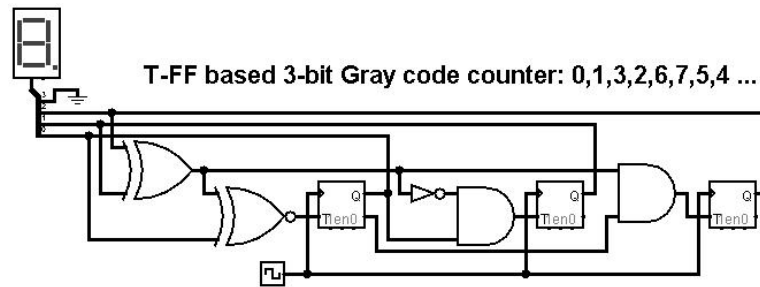
$$G_i = B_{i+1} \oplus B_i$$

Čas pisanja je 60 minut. Vsaka naloga je vredna 10 točk.

Na list z rešitvami se podpišite in napišite še vpisno številko ter kateri predmet pišete (VŠŠ, UNI).

Rezultati bodo objavljeni na: <https://estudent.fri.uni-lj.si>

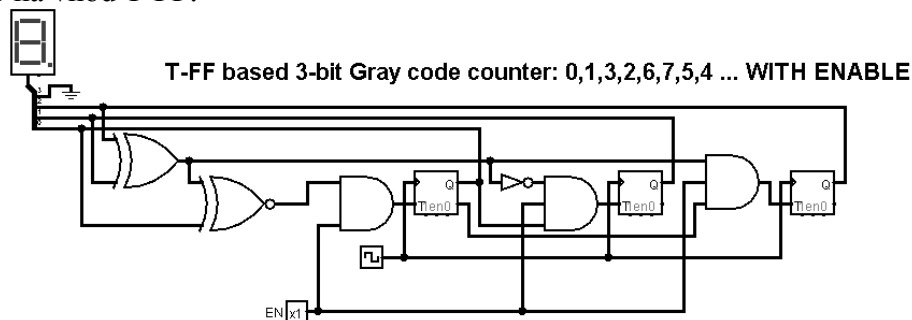
Vezje se nahaja v Logisim predlogah rešenih nalog na domači strani predmeta:
Logisim\counter\3-bit Gray code counter_revB.circ



Funkcijo omogočanja štetja bi lahko realizirali s pisanjem tabele prehajanje stanj števca za štiri spremenljivke (ENABLE, Q_0 , Q_1 , Q_2). Z malo razmišljanja se izognemo dolgotrajnemu pisanju: Števec bo obstal (ENABLE='0'), če so vsi vhodi T-FF enaki '0'. Števec bo štel (ENABLE='1'), ko bodo vhodi T-FF lahko spreminjali stanje, torej se bo izhod prejšnje stopnje nespremenjen pojavil na vhodu naslednje stopnje T_{i+1} . Povedano strnemo v tabelo za vhod vsakega T-FF:

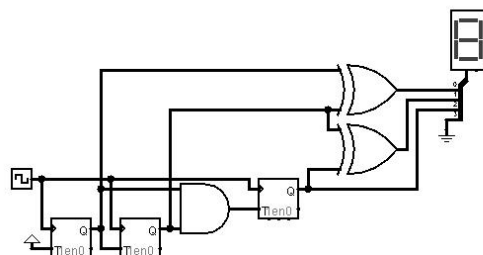
ENABLE	Izhod prejšnje stopnje	T_{i+1}
0	0	0
0	1	0
1	0	0
1	1	1

Funkcijo ENABLE torej realiziramo z dvovhodnimi AND vrati pred vhodom posameznega T-FF. Na vhoda AND vrat vodimo ENABLE in izhod iz prejšnje stopnje štetja, kot kaže desni del slike. Na LSB mestu štetja AND vrat ne potrebujemo, ampak ENABLE priključimo neposredno na vhod T-FF.



Enostavnejšo izvedbo štetja dosežemo z uporabo sinhronnega 3 bitnega dvojiškega števca in pretvornika kode iz dvojiške v Gray-evo kodo.

T-FF based 3-bit Gray code counter: 0,1,3,2,6,7,5,4 ...



Vezje se nahaja v Logisim predlogah rešenih nalog na domači strani predmeta:
Logisim\counter\3-bit Gray code counter_revC.circ

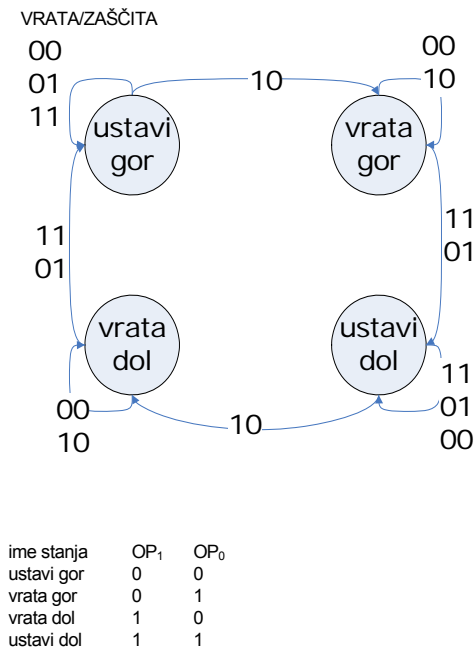
Čas pisanja je 60 minut. Vsaka naloga je vredna 10 točk.

Na list z rešitvami se podpišite in napišite še vpisno številko ter kateri predmet pišete (VSŠ, UNI).

Rezultati bodo objavljeni na: <https://estudent.fri.uni-lj.si>

Rešitev 4. naloge:

Narišemo Moore-ov diagram stanj:



Iz opisa naloge je razvidno, da stanje "ustavi" ni samo eno, ker si moramo zapomniti v katero smer so se gibala vrata, da bi lahko šli v nasprotni smeri. Glede na to imamo stanja "ustavi gor", ki določa, da se bodo vrata ob naslednjem pritisku na gumb gibala gor in stanje "ustavi dol", ki določa, da se bodo vrata ob naslednjem pritisku na gumb gibala dol. Če stanja ločimo tako, potem v stanju "ustavi gor" ostajamo toliko časa, dokler ne pritisnemo VRATA in jasno na motorju ni napake, se pravi kombinacija "10". Vrata se nato pomikajo gor (preidemo v stanje "vrata gor"). V tem stanju lahko tipko spustimo in vrata se pomikajo navzgor. To se dogaja toliko časa, dokler ne naletimo na pogoj ZAŠČITA='1' (se pravi kombinaciji "11" in "01". Ko postane pogoj ZAŠČITA='1' se postavimo v stanje "ustavi dol" in v tem stanju ostajamo dokler vztraja pogoj ZAŠČITA='1' oz. dokler ne pritisnemo tipke

VRATA='1' (kombinacija "10"). Takrat na podoben način preidemo v stanje "vrata dol", kjer ostanemo dokler ne naletimo na oviro (tla prostora recimo), ko preidemo v stanje "ustavi gor".