

# RAZVOJ DIGITALNIH SISTEMOV

Izpit 30. 01. 2013

1. Realizirajte funkcijo  $f$  v obliki PDNO z redundancami s čim manj izbiralniki 4/1.

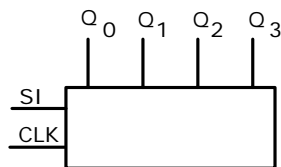
$$f(x_1, x_2, x_3, x_4) = V(1, 2, 9, 13, 15) \text{ in } V_x(0, 5, 11, 12)$$

2. Uporabite ROM vezje za realizacijo naslednjih funkcij:

$$g_1 = x_1 + \overline{x_2} \cdot \overline{x_3} \quad g_2 = \overline{x_1} \cdot x_3 + x_1 \cdot x_2$$

ROM vezje ima 3 naslovne spremenljivke in 2 bitno vsebino. Narišite shemo elementa in v shemi označite programirane povezave oz. 'varovalke' s piko (●).

3. Narišite vezje 4-bitnega pomikalnega registra s T-celicami in logičnimi vrati. Register ima zaporedni vhod  $SI$  (ang. serial input) in vzporedni izhod ( $Q_0, Q_1, Q_2, Q_3$ ).



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 (npr. tla prostora). Vezje ima 2-bitni izhod za enosmerni motor:

Koda izhoda		Funkcija izhoda
OP <sub>1</sub>	OP <sub>0</sub>	
0	0	motor stoji
0	1	motor pomika vrata navzgor
1	0	motor pomika vrata navzdol

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

Podobno je v obratni smeri: Če vrata na poti naletijo na oviro ali pridejo do spodnje končne lege (tla prostora), se motor ustavi. Če uporabnik pritisne gumb VRATA, se ta 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 (VSŠ, UNI).

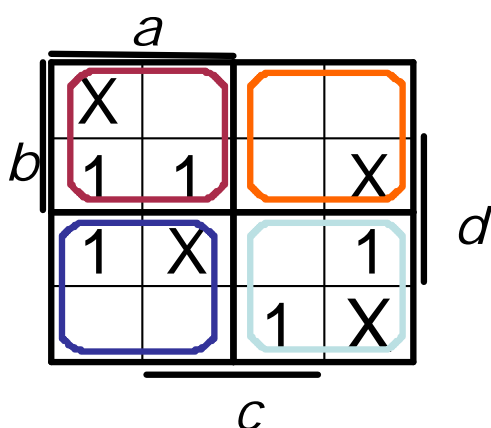
Rezultati bodo objavljeni na: <https://estudent.fri.uni-lj.si>

Rešitev 1. naloge:

Funkcija  $f$  je podana v obliki PDNO z redundancami.

$$f = V(1,2,9,13,15) \text{ in } V_x(0,5,11,12)$$

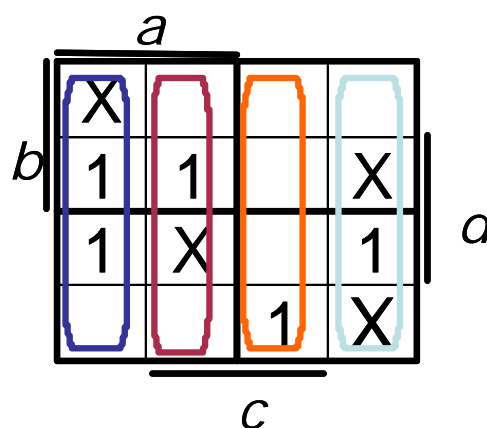
Dobljeno funkcijo vrišemo v Veitch-ev diagram. Ker iščemo najcenejšo realizacijo z izbiralnikom 4/1, bomo naredili razvoj po vseh kombinacijah naslovnih spremenljivk v Veitchev-em diagramu. Če izberemo kot naslovni spremenljivki a b, potem dobimo:



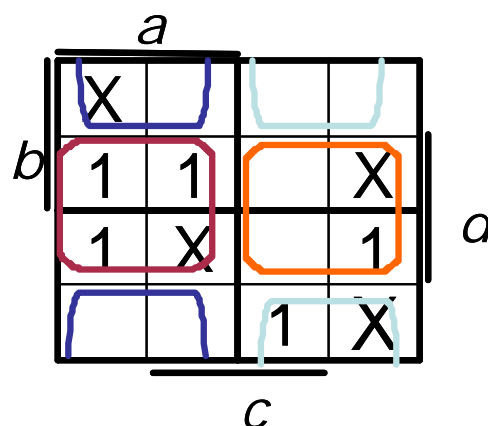
V zgornjem Veitch-evem diagramu so označena vsa štiri polja štirih mintermov, če izberemo vhodni spremenljivki a in b. Zgornji levi kvadrat (rdeč) pomeni, da bo to polje izbrano ko bosta  $ab="11"$ , oranžni kvadrat ko bo  $ab="01"$ , temno modri ko bo  $ab="10"$  in svetlo modri ko bo  $ab="00"$ . Vsakega od teh kvadratov poskušamo opisati s čimbolj enostavno funkcijo: Vrednost zgornjega levega kvadrata opišemo s spremenljivko d, če postavimo redundanco na '0'. Vrednost spodnjega desnega kvadrata je bolj komplicirana, saj moramo vsako '1' opisati posebej: Za zgornjo '1' v tem kvadratu velja  $c \cdot d$ , za spodnjo '1' pa  $c \cdot d'$ . Funkcija bo torej  $c \cdot d + c \cdot d'$ , kar je enačba funkcije XOR. Najbolj enostavna realizacija je zgornji desni kvadrat, ki je kar '0', če postavimo redundanco na '0'. Zato, da bi pregledali še ostale možnosti, moramo

narisati še preostalih pet kombinacij dveh naslovnih vhodov.

Če izberemo kot naslovni spremenljivki a in c, dobimo levi Veitchev diagram, če a in d, pa desnega. Podobno kot v prejšnjem primeru poiščemo realizacije ustreznih kvadratov in iščemo najcenejšo realizacijo: Izogibamo se veliko različnim funkcijam in iščemo inačice kvadratov, ki vsebujejo same '1' ali same '0'. Pri razvoju po a in c imamo pri  $ac="01"$  najneugodnejšo funkcijo, saj vsebuje eno samo '1'.



Pri razvoju po a in d nikjer ne nastopa ena sama '1' ali tri '1' ali diagonala (XOR) dveh '1'.

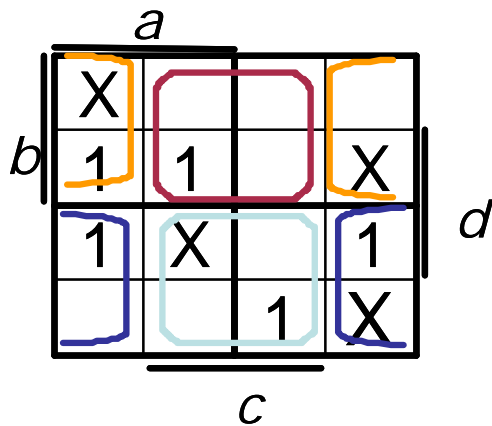


Čas pisanja je 60 minut. Vsaka naloga je vredna 10 točk.

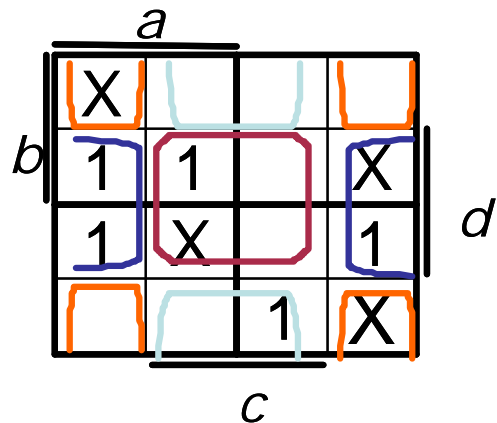
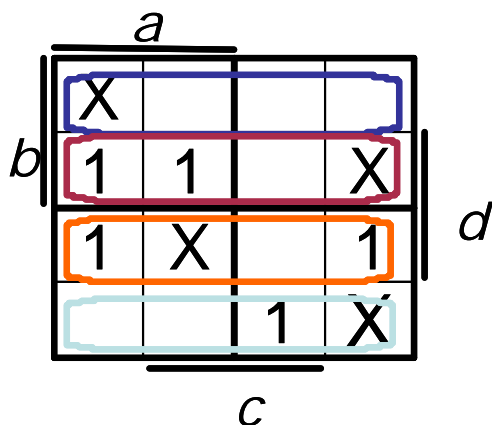
Na list z rešitvami se podpišite in napišite še vpisno številko ter kateri predmet pišete (VSŠ, UNI).

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

Nato izberemo naslovni spremenljivki b in c, (levi Veitchev diagram) in b in d (desni diagram). Pri razvoju po b in c imamo pri bc="11" najneugodnejšo funkcijo (rdeč), saj vsebuje eno samo '1'.

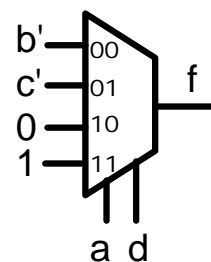


Pri razvoju po b in d pri bd="11" (rdeč) najneugodnejšo funkcijo, saj vsebuje tri '1'.



Zadnja kombinacija naslovnih vhodov je cd. Pri razvoju po c in d imamo pri cd="10" najneugodnejšo funkcijo (svetlo moder), saj vsebuje eno samo '1'. Najbolj ugodna kombinacija za realizacijo je torej razvoj po spremenljivkah a in d.

Končna realizacija funkcije:



## Rešitev 2. naloge:

Če se funkcije ne nahajajo v popolni disjunktivni normalni obliki (PDNO), jih prevedemo v to obliko z uporabo pravil Boole-ove algebre. Funkcijo lahko tudi izpišemo v Veitch-ev diagram in izpišemo številke mintermov, kjer je funkcija enaka '1'.

$$g_1(x_1, x_2, x_3) = x_1 + \overline{x_2} \cdot \overline{x_3} = x_1 \cdot (\overline{x_2} \cdot \overline{x_3} + x_2 \cdot \overline{x_3} + \overline{x_2} \cdot x_3 + x_2 \cdot x_3) + (\overline{x_1} + x_1) \cdot \overline{x_2} \cdot \overline{x_3}$$

$$g_1(x_1, x_2, x_3) = x_1 \cdot \overline{x_2} \cdot \overline{x_3} + x_1 \cdot x_2 \cdot \overline{x_3} + x_1 \cdot \overline{x_2} \cdot x_3 + x_1 \cdot x_2 \cdot x_3 + \overline{x_1} \cdot \overline{x_2} \cdot \overline{x_3} + \overline{x_1} \cdot x_2 \cdot \overline{x_3}$$

$$g_1(x_1, x_2, x_3) = x_1 \cdot \overline{x_2} \cdot \overline{x_3} + x_1 \cdot x_2 \cdot \overline{x_3} + x_1 \cdot \overline{x_2} \cdot x_3 + x_1 \cdot x_2 \cdot x_3 + \overline{x_1} \cdot \overline{x_2} \cdot \overline{x_3}$$

$$g_1(x_1, x_2, x_3) = V(4, 6, 5, 7, 0)$$

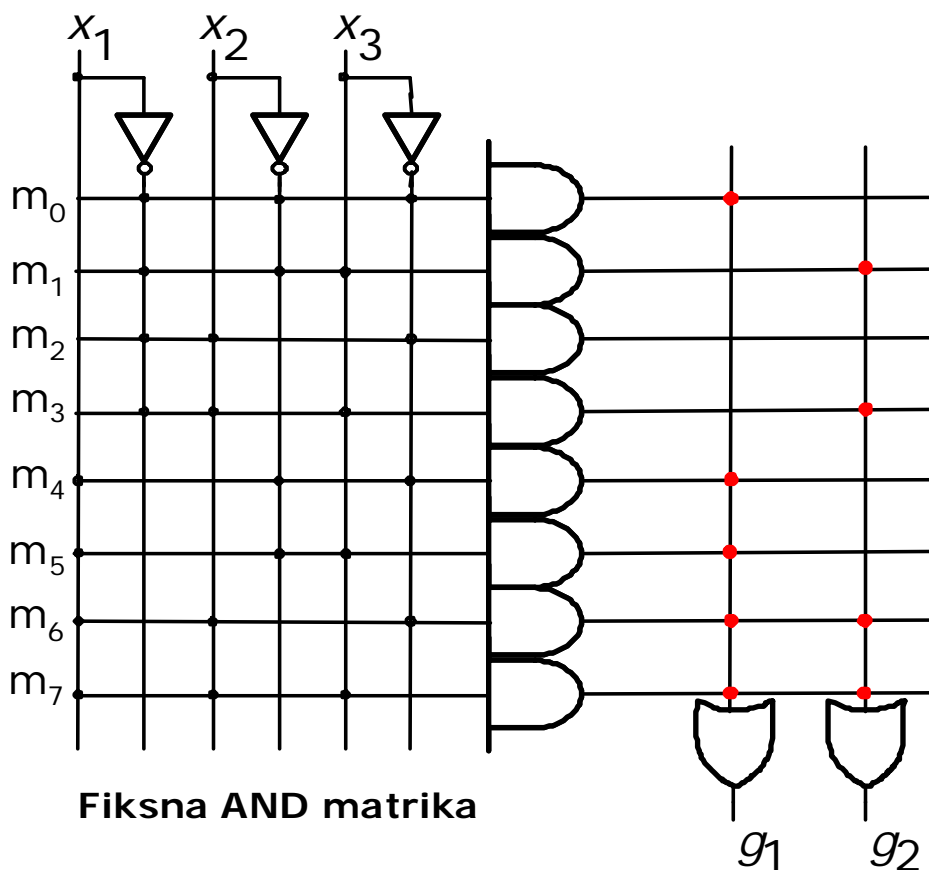
Podobno storimo še za funkcijo  $g_2$ :

$$g_2 = \overline{x_1} \cdot x_3 + x_1 \cdot x_2 = \overline{x_1} \cdot (\overline{x_2} + x_2) \cdot x_3 + x_1 \cdot x_2 \cdot (\overline{x_3} + x_3)$$

$$g_2(x_1, x_2, x_3) = \overline{x_1} \cdot \overline{x_2} \cdot x_3 + \overline{x_1} \cdot x_2 \cdot x_3 + x_1 \cdot x_2 \cdot \overline{x_3} + x_1 \cdot x_2 \cdot x_3$$

$$g_2(x_1, x_2, x_3) = V(1, 3, 6, 7)$$

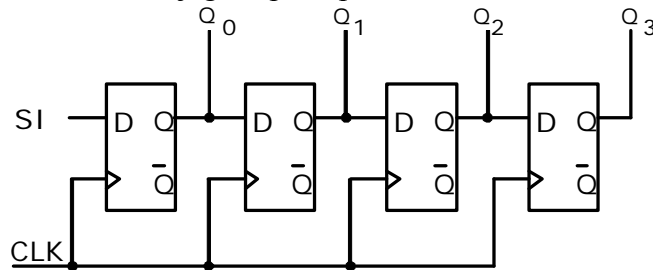
PDNO je najprimernejša oblika za realizacijo z ROM, ker je matrika AND fiksna. Programirane vrednosti AND matrike predstavljajo vse minterme funkcije treh spremenljivk ( $x_1 x_2 x_3$ ) od  $m_0$  do  $m_7$ . Številka minterma določa naslov lokacije ROM pomnilnika.



Narišemo celotno vezje ROM strukture in vstavimo pike (•) v OR matriki tam, kjer želimo programirati določeno spremenljivko v členu PDNO.

### Rešitev 3. naloge:

Zaporedno–vzporedni (SIPO) pomikalni register, realiziran s pomočjo D–FF, je veriga kaskadno vezanih D–FF, v kateri je izhod prejšnjega flip–flopa  $Q_{i-1}$  vezan na vhod naslednjega flip–flopa  $D_i$ .



Če želimo pomikalni register sestaviti iz T–FF in logičnih vrat, moramo pravzaprav realizirati celico D–FF s pomočjo T–FF in logičnih vrat.

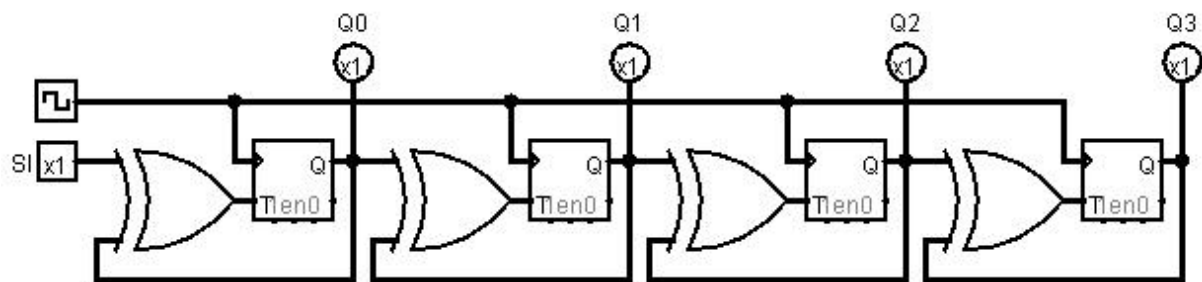
V ta namen zapišemo tabelo D–FF, pri kateri dodamo izhodni stolpec T vhoda.

$D$	$Q(t)$	$Q(t+1)$	$T$
0	0	0	0
0	1	0	1
1	0	1	1
1	1	1	0

Iz tabele sledi, da je  $T$  vhod XOR operacija  $Q(t)$  in vhoda D–FF, ki ga realiziramo.

$$Q(t+1) = Q(t) \oplus D$$

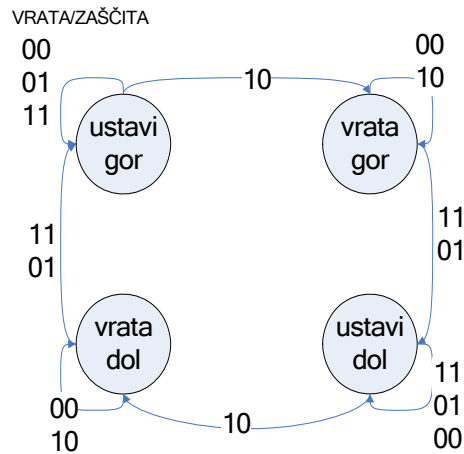
Če nastali D–FF sestavimo skupaj v 4–bitni pomikalni register dobimo spodnjo realizacijo.



Vezje se nahaja v Logisim predlogah rešenih nalog na domači strani predmeta:  
Logisim\shift\_reg\shift\_reg\_4bit\_using\_tff\_xor.circ

#### Rešitev 4. naloge:

Moore-ov diagram stanj avtomata krmilja za garažna vrata:



ime stanja	izhod stanja	
	OP <sub>1</sub>	OP <sub>0</sub>
ustavi gor	0	0
vrata gor	0	1
vrata dol	1	0
ustavi dol	0	0

Iz opisa naloge je razvidno, da stanje "ustavi" ni samo eno, ker si mora avtomat zapomniti v katero smer se bodo gibala vrata ob naslednjem pritisku na tipko VRATA.

Avtomat ima torej:

- stanje "ustavi gor", ki določa, da se bodo vrata ob naslednjem pritisku na gumb gibala navzgor,
- stanje "ustavi dol", ki določa, da se bodo vrata ob naslednjem pritisku na gumb gibala navzdol.

Avtomat v stanju "ustavi gor" ostaja toliko časa, dokler uporabnik ne pritisne tipke (VRATA='1') in dokler na motorju ni napake (ZAŠČITA='0'), kar je kombinacija "10".

Vrata se pomikajo gor (avtomat preide v stanje "vrata gor"). V tem stanju lahko uporabnik tipko spusti (VRATA='0') in vrata se še vedno pomikajo navzgor. To se dogaja, dokler avtomat ne naleti na pogoj ZAŠČITA='1' (kombinaciji "11" in "01").

Ko postane ZAŠČITA='1', se avtomat postavi v stanje "ustavi dol" in v tem stanju ostaja dokler ostaja ZAŠČITA='1'. V tem stanju se motor ustavi in ZAŠČITA postane '0', ker čez motor ne teče tok. Če v tem stanju uporabnik pritisne tipko (VRATA='1') in je (ZAŠČITA='0') bo avtomat prešel v stanje "vrata dol". V tem stanju se vrata gibljejo navzdol, dokler ne naletijo na oviro (npr. tla prostora), ko preide v stanje "ustavi gor" ob pogoju (ZAŠČITA='1') ne glede na stanje tipke uporabnika.