

# RAZVOJ DIGITALNIH SISTEMOV

Izpit

31. 01. 2018

1. Realizirajte podano funkcijo  $f$  z redundancami s čim manj 4-bitnimi aritmetičnimi–logičnimi enotami (ALU). Negacije vhodnih spremenljivk izvedite z ALU.

$$f(x_1, x_2, x_3, x_4) = V(0, 5, 6, 9, 10, 12) \text{ in } V_x(3, 15)$$

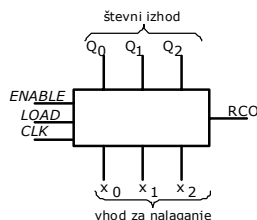
2. Uporabite PAL3L3 (namišljen čip) za realizacijo naslednjih funkcij:

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

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.

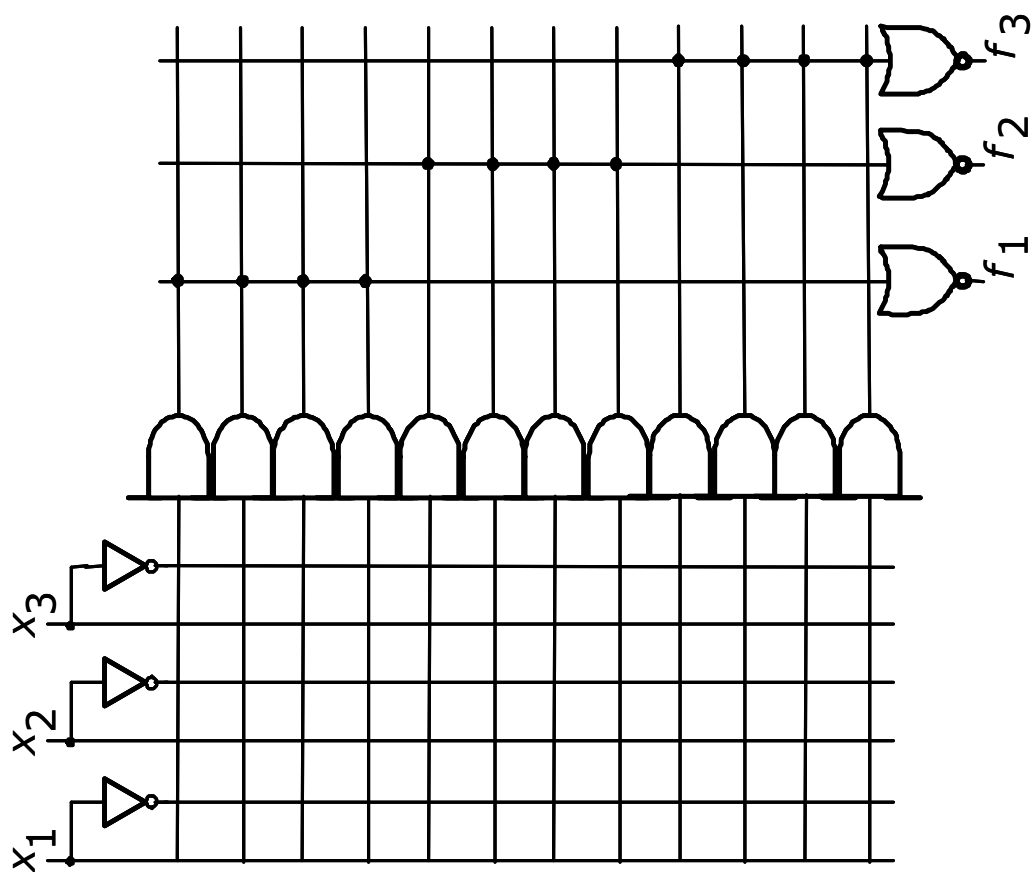
3. Prikažite sintezo 3-bitnega sinhronega števca navzgor z omogočanjem štetja (ENABLE) in vzporednim nalaganjem (LOAD) z D flip-flopi, izbiralniki 2/1 in logičnimi vrati. Podatek se vzporedno nalaga z vhodov ( $x_2, x_1, x_0$ ).

Števec naj ima poleg števnega izhoda ( $Q_2, Q_1, Q_0$ ) tudi izhodni prenos za krmiljenje naslednjih stopenj (RCO – ripple carry out). Logika vseh krmilnih signalov je pozitivna. Uporabite poimenovanje signalov, kot je narisano na spodnji sliki.

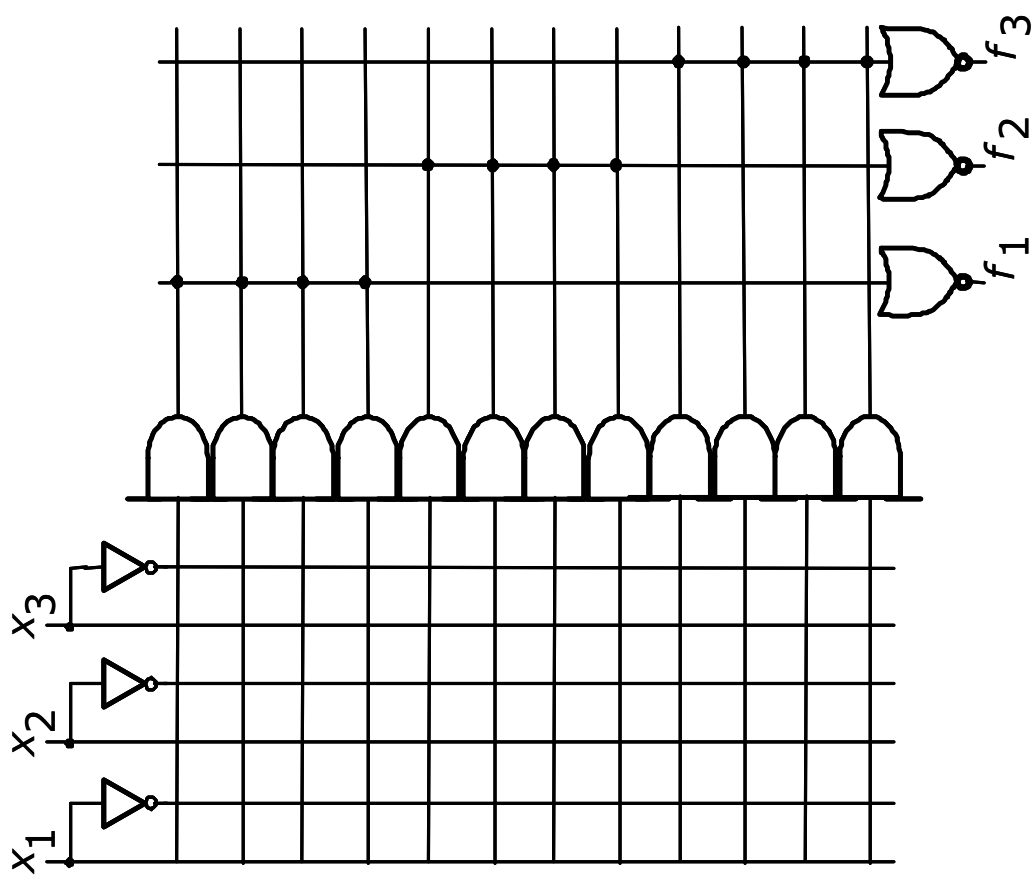


4. Minimizirajte podani avtomat končnih stanj z uporabo metode z razdelki ter zapišite tabelo prehajanja stanj nastalega minimalnega avtomata.

Trenutno stanje	Naslednje stanje		Izhod
	$w=0$	$w=1$	
A	B	C	1
B	D	F	1
C	F	E	0
D	B	G	1
E	F	C	0
F	E	D	0
G	F	G	0

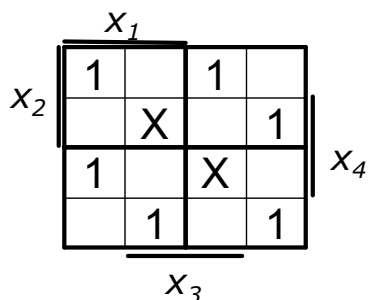


Če se zmotite, prečrtajte napačno shemo in uporabite drugo shemo!



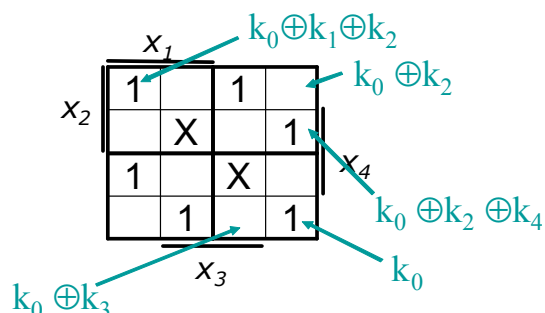
Rešitev 1. naloge:

Funkcijo najprej izrišemo v Veitch–ev diagram:



Funkcija vsebuje same diagonalne člene, zato realizacija v obliki KNO oz. DNO ne nudi minimalne oblike. Če se izkaže, da je funkcija linearna, jo lahko realiziramo s pomočjo XOR funkcij. Linearnost funkcije ugotavljamo tako, da prepogibamo kvadrate diagrama: Začnemo v desnem spodnjem kotu (kjer je minterm 0) in prepognemo kvadrat navzgor, da se spremeni samo ena spremenljivka naenkrat ( $x_4$  postane 0 v prvi iteraciji).

Opazujemo, ali se prepogne na novi kvadrat čisto enako ali pa popolnoma negirano. Če postavimo obe redundanci na '1', lahko s prepogibanjem ugotovimo, da je funkcija linearna.



Podana funkcija je funkcija 4 spremenljivk, zato lahko njeno splošno izražavo kot linearno funkcijo pišemo kot:

$$f(x_1, x_2, x_3, x_4) = k_0 \oplus k_1 x_1 \oplus k_2 x_2 \oplus k_3 x_3 \oplus k_4 x_4$$

S pomočjo Veitch–evega diagrama izračunamo koeficiente.

Iz enačb sledi:  $k_0=1$  in  $k_0 \oplus k_3=0$ , kar pomeni  $1 \oplus k_3=0 \rightarrow k_3=1$ .

In če napišemo še enačbo za  $k_0 \oplus k_2=0$ , kar pomeni  $1 \oplus k_2=0$  sledi da je  $k_2=1$ .

Iz enačbe  $k_0 \oplus k_2 \oplus k_4=1$ , kar pomeni  $1 \oplus 1 \oplus k_4=1 \rightarrow k_4=1$ .

Analiziramo naprej in dobimo  $k_0 \oplus k_1 \oplus k_2=1$ , kar pomeni  $1 \oplus k_1 \oplus 1=0 \rightarrow k_1=1$ .

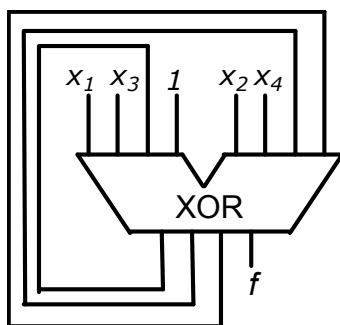
Vstavimo dobljene koeficiente v enačbo za splošno izražavo in dobimo:

$$f(x_1, x_2, x_3, x_4) = 1 \oplus x_1 \oplus x_2 \oplus x_3 \oplus x_4$$

Aritmetično–logično enota lahko poleg aritmetičnih naenkrat realizira štiri dvovhodne logične operacije *istega tipa* (OR, AND, NOT, NOR, NAND, XOR, XNOR), zato nas zanima realizacija zgornje funkcije z dvovhodnimi operatorji enega tipa. Pri realizaciji uporabimo lastnost združevanja, ki velja za XOR funkcijo.

$$f(x_1, x_2, x_3, x_4) = 1 \oplus ((x_1 \oplus x_2) \oplus (x_3 \oplus x_4))$$

Rešitev:



## 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.

### Rešitev 3. naloge:

Postopek sinteze zahteva, da zapišemo tabelo prehajanja stanj števca:

trenutno stanje			naslednje stanje			D–FF		
Q <sub>2</sub>	Q <sub>1</sub>	Q <sub>0</sub>	Q <sub>2</sub>	Q <sub>1</sub>	Q <sub>0</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
0	0	0	0	0	1	0	0	1
0	0	1	0	1	0	0	1	0
0	1	0	0	1	1	0	1	1
0	1	1	1	0	0	1	0	0
1	0	0	1	0	1	1	0	1
1	0	1	1	1	0	1	1	0
1	1	0	1	1	1	1	1	1
1	1	1	0	0	0	0	0	0

Iz tabele prehajanja stanj števca določimo enačbe D–FF:

Za D<sub>0</sub> se iz tabele vidi  $D_0 = Q_0'$ .

Za D<sub>1</sub> narišemo Veitchev diagram

$D_1$ :

	Q <sub>2</sub>			
Q <sub>1</sub>	1	0	0	1
	0	1	1	0
	Q <sub>0</sub>			

$$D_1 = Q_0 \oplus Q_1$$

Podobno za D<sub>2</sub> narišemo Veitchev diagram

$D_2$ :

	Q <sub>2</sub>			
Q <sub>1</sub>	1	0	1	0
	1	1	0	0
	Q <sub>0</sub>			

Za D<sub>2</sub> sledi:

$$D_2 = Q_2 \cdot Q_1' + Q_2 \cdot Q_0' + Q_2' \cdot Q_1 \cdot Q_0$$

iz česar lahko izpostavimo:

$$D_2 = Q_2 \cdot (Q_1' + Q_0') + Q_2' \cdot Q_1 \cdot Q_0$$

Uporabimo De Morganovo enakost:

$$D_2 = Q_2 \cdot (Q_1 Q_0)' + Q_2' \cdot Q_1 \cdot Q_0$$

iz česar sledi:

$$D_2 = Q_2 \cdot (Q_1 \cdot Q_0)' + Q_2' \cdot Q_1 \cdot Q_0$$

Upoštevamo definicijo XOR operacije ( $a \oplus b = a' \cdot b + a \cdot b'$ )

$$D_2 = Q_2 \oplus Q_1 \cdot Q_0$$

Signal RCO postane '1' takrat, ko števec prešteje do svoje največje vrednosti – v našem primeru postane '1', ko gre stanje števca iz "111" na "000".

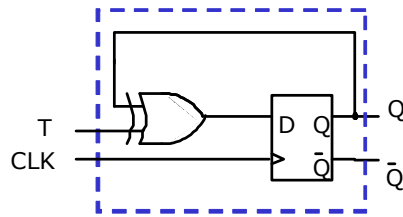
$$RCO = Q_2 \cdot Q_1 \cdot Q_0$$

Naloga pravi, da moramo izdelati števec, ki ima vhod za omogočanje štetja (ENABLE). Če vezje analiziramo, vidimo, da smo pravzaprav realizirali T–FF s pomočjo D–FF in XOR vrat, kot kaže spodnja slika:

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).

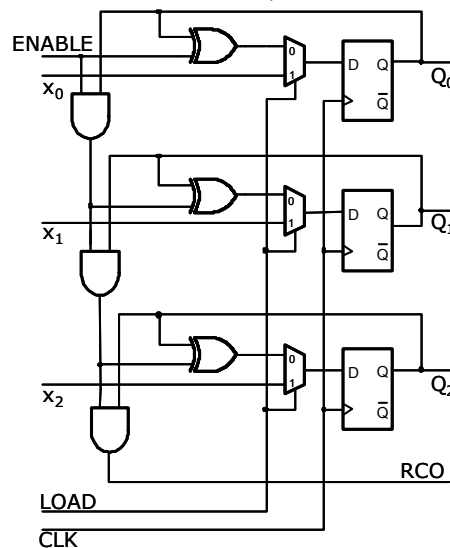
Rezultati bodo objavljeni na: <https://estudent.fri.uni-lj.si>



Slika 1: Realizacija T-FF s pomočjo D-FF.

Če prvemu "T-FF" (D-FF z XOR vrati) postavimo vhod  $T_0 = '0'$  namesto  $T_0 = '1'$ , vsi T-FF ne bodo šteli, ampak bodo ohranjali stanje. Torej, če na vhod  $T_0$  postavimo zunanji signal ENABLE, števec ne bo štel, ampak ohranjal stanje, če bo  $ENABLE = '0'$ . V verigi sinhronnega števca so namreč vsi T-FF vezani tako, da so odvisni od prvega T-FF: Če stanje ohranja prvi, ga bodo tudi vsi ostali.

Za realizacijo signala za vzporedno nalaganje pa izkoristimo osnovno lastnost D-FF (pomnjenje). To storimo tako, da na vhod vsakega D-FF postavimo 2/1 izbiralnik, s katerim določimo, ali se bo dana informacija vpisala s števnege vhoda ali preko zunanjih priključkov. Do iste realizacije bi prišli, če bi v osnovni analizi upoštevali ta dva krmilna signala – analiza je veliko bolj zapletena, saj vsebuje Veitcheve diagrame 5 spremenljivk (ENABLE, LOAD,  $Q_2$ ,  $Q_1$ ,  $Q_0$ ).



Slika 2: Sinhroni števec z vzporednim nalaganjem (LOAD) in omogočanjem štetja (ENABLE) (3-bitna izvedba).

Če želimo z nastalim števcem šteti naraščajoče ... 2, 3, 4, 5, 2, 3, 4, 5 ..., moramo števec, ko le-ta pride do stanja 5 ( $Q_2Q_1Q_0 = 101_2$ ) postaviti nazaj na stanje v stanje 2 ( $Q_2Q_1Q_0 = x_2x_1x_0 = 010_2$ ), torej na LOAD vhod pripeljemo s pomočjo dodatnih dvovhodnih AND vrat.

Pomembno pri tem je, da se pri dekodiranju zavedamo, da se ( $Q_2 = 1$  in  $Q_0 = 1$  v števeni sekvenci pojavlja samo enkrat – če bi se večkrat bi morali dekodirati tudi  $Q_1$ ).

Pri tovrstnih števcih ponavadi uporabljamo še zunanji signal RESET, s katerim postavimo števec v začetno stanje, kar dosežemo tako, da na vhod izbiralnikov vodimo LOAD OR RESET.

Rešitev 4. naloge:

V prvi iteraciji zberemo skupaj vsa stanja v enem razdelku:  $P_1 = (ABCDEFGG)$

Trenutno stanje	Naslednje stanje		Izhod z
	w=0	w=1	
A	B	C	1
B	D	F	1
C	F	E	0
D	B	G	1
E	F	C	0
F	E	D	0
G	F	G	0

Naslednja iteracija loči stanja, ki imajo različne izhode:  $P_2 = (ABD)(CEFG)$

- Pregledamo vsa naslednja stanja pri vhodu 0 in 1 v vsakem bloku:
    - Blok (ABD):
      - Naslednja stanja pri w=0 (BDB)
      - Naslednja stanja pri w=1 (CFG)
    - Blok (CEFG):
      - Naslednja stanja pri w=0 (FFEF)
      - Naslednja stanja pri w=1 (ECDG)
- Vsa stanja niso v enem bloku. Problem je pri stanju F, ki ima naslednje stanje D. Zato bo stanje F NEEKVIVALENTNO ostalim CEG.
- Novo stanje F zato postavimo v svojo skupino.

Naslednja iteracija loči stanje F od ostalih  $P_3 = (ABD)(CEG)(F)$

- Blok (ABD):
  - Naslednja stanja pri w=0 (BDB)  
So vsa v istem bloku
  - Naslednja stanja pri w=1 (CFG) Niso v istem bloku, ker je F v drugem bloku kot C in G. Zato bo stanje B v novem bloku.
- Blok (CEG):
  - Naslednja stanja pri w=0 (FFF)
  - Naslednja stanja pri w=1 (ECG) C, E in G imamo lahko še vedno za ekvivalentna

Trenutno stanje	Naslednje stanje		Izhod z
	w=0	w=1	
A	B	C	1
B	D	F	1
C	F	E	0
D	B	G	1
E	F	C	0

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>

<i>F</i>	<i>E</i>	<i>D</i>	<i>0</i>
<i>G</i>	<i>F</i>	<i>G</i>	<i>0</i>

Naslednja iteracija loči stanje B od ostalih  $P_4=(AD)(B)(CEG)(F)$

- Blok (*AD*)
  - Naslednja stanja pri  $w=0$  (*BB*)
  - Naslednja stanja pri  $w=1$  (*CG*)
  - So vsa v istem bloku.
- Blok (*CEG*)
  - Naslednja stanja pri  $w=0$  (*FFF*)
  - Naslednja stanja pri  $w=1$  (*ECG*) So vsa v istem bloku.

Trenutno stanje	Naslednje stanje		Izhod z
	w=0	w=1	
<i>A</i>	<i>B</i>	<i>C</i>	<i>1</i>
<i>B</i>	<i>D</i>	<i>F</i>	<i>1</i>
<i>C</i>	<i>F</i>	<i>E</i>	<i>0</i>
<i>D</i>	<i>B</i>	<i>G</i>	<i>1</i>
<i>E</i>	<i>F</i>	<i>C</i>	<i>0</i>
<i>F</i>	<i>E</i>	<i>D</i>	<i>0</i>
<i>G</i>	<i>F</i>	<i>G</i>	<i>0</i>

$P_5=(AD)(B)(CEG)(F)$

Iteraciji  $P_5$  in  $P_4$  sta enaki, zato se postopek minimizacije zaključi. Stanji *A* in *D* sta ekvivalentni. Stanja *C*, *E* in *G* so ekvivalentna.

- Tabelo stanj zapišemo na novo
- Izbrišemo vrstice za *D*, *E* in *G*
- Zamenjamo stanja:  $D \rightarrow A$  in vse  $E \rightarrow C$  ter  $G \rightarrow C$

Rezultat je nova tabela stanj minimiziranega avtomata:

Trenutno stanje	Naslednje stanje		Izhod z
	w=0	w=1	
<i>A</i>	<i>B</i>	<i>C</i>	<i>1</i>
<i>B</i>	<i>A</i>	<i>F</i>	<i>1</i>
<i>C</i>	<i>F</i>	<i>C</i>	<i>0</i>
<i>F</i>	<i>C</i>	<i>A</i>	<i>0</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 (VSŠ, UNI).

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