

RAZVOJ DIGITALNIH SISTEMOV

Izpit

5. 7. 2018

1. Realizirajte podano funkcijo f z redundantnimi makstermi z enim izbiralnikom 4/1.

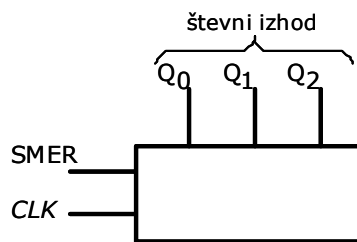
$$f(x_1, x_2, x_3, x_4) = \&(1, 2, 5-7, 9, 10, 14) \text{ in } \&_x(0, 4, 11, 13)$$

2. Uporabite ROM vezje za realizacijo funkcij:

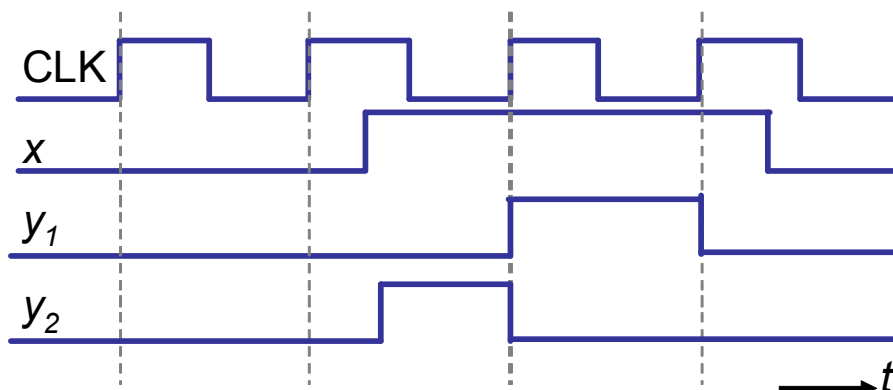
$$g_1 = x_1 + \overline{x_2} \cdot \overline{x_3} \quad g_2 = \overline{x_1} \cdot x_3 + x_1 \cdot x_2$$

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

3. Prikažite sintezo sinhronega dvosmernega 3-bitnega števca z uporabo T flip-flopov: Zapišite tabelo prehajanja stanj in določite enačbe flip-flopov. Števec ima vhod SMER, ki določa smer štetja: Če je SMER='0', števec šteje naraščajoče, sicer padajoče. Imena signalov so razvidna iz spodnje slike.



4. Z uporabo D flip-flopov, ki so proženi na sprednji rob signala ure (CLK) načrtajte Mooreov ali Mealyev avtomat končnih stanj, katerega izhod y postane '1' ko se vhod x spremeni iz logične '0' na '1'. Delovanje avtomata povzema spodnja slika. Narisani sta dve realizaciji izhoda avtomata y_1 in y_2 : Kateri izhod (y_1 ali y_2) pripada Moore-ovemu tip avtomata in kateri Mealy-evemu?



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 na: <https://studij.fe.uni-lj.si>

Rešitev 1. naloge:

Funkcija f je podana v obliki PKNO z redundancami.

$$f(x_1, x_2, x_3, x_4) = \&(1, 2, 5 - 7, 9, 10, 14) \text{ in } \&_x(0, 4, 11, 13)$$

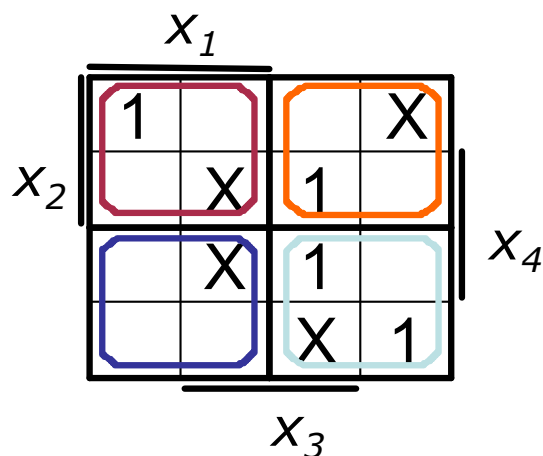
Za potrebe realizacije jo najprej pretvorimo v obliko PDNO. To storimo tako, da maksterme preslikamo v minterme. V pravilnostno tabelo funkcije najprej zapišemo številke mintermov (m) in pripadajoče številke makstermov (M). Vpišemo $f='0'$ za vse maksterme in $f='X'$ za vse redundantne maksterme. Na preostala mesta vpišemo $f='1'$ in preberemo pri katerih mintermih je $f='1'$ oz. $f='X'$ ter funkcijo izrazimo v obliki PDNO.

m	M	x_1	x_2	x_3	x_4	f
0	15	0	0	0	0	1
1	14	0	0	0	1	0
2	13	0	0	1	0	X
3	12	0	0	1	1	1
4	11	0	1	0	0	X
5	10	0	1	0	1	0
6	9	0	1	1	0	0
7	8	0	1	1	1	1
8	7	1	0	0	0	0
9	6	1	0	0	1	0
10	5	1	0	1	0	0
11	4	1	0	1	1	X
12	3	1	1	0	0	1
13	2	1	1	0	1	0
14	1	1	1	1	0	0
15	0	1	1	1	1	X

Dobimo:

$$f = V(0, 3, 7, 12) \text{ in } V_x(2, 4, 11, 15)$$

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 x_1 x_2 , potem dobimo:



V zgornjem Veitch-evem diagramu so označena vsa štiri polja funkcijskih ostankov, če izberemo vhodni spremenljivki x_1 x_2 . Zgornji levi kvadrat (rdeč) pomeni, da bo to polje izbrano ko bosta $x_1 x_2 = "11"$, oranžni kvadrat ko bo $x_1 x_2 = "01"$, temno modri ko bo $x_1 x_2 = "10"$ in svetlo modri ko bo $x_1 x_2 = "00"$. Vsakega od teh kvadratov poskušamo opisati s čimbolj enostavno funkcijo: Vrednost zgornjega levega kvadrata je komplicirana, saj moramo vsako '1' opisati posebej: Za zgornjo '1' v tem kvadratu velja $x_3' x_4'$. Če bi v tem kvadratu postavili redundanco na '1' jo bi opisali kot $x_3 x_4$. Funkcija bo torej $x_3' x_4' + x_3 x_4$, kar je enačba funkcije ekvivalence (XNOR). Podobno sklepanje velja za zgornji in spodnji desni kvadrat. Najbolj enostavna realizacija je spodnji levi kvadrat je konstanta '0', če postavimo redundanco na '0'. Za ostale možnosti realizacije moramo narisati še preostalih pet kombinacij dveh naslovnih vhodov.

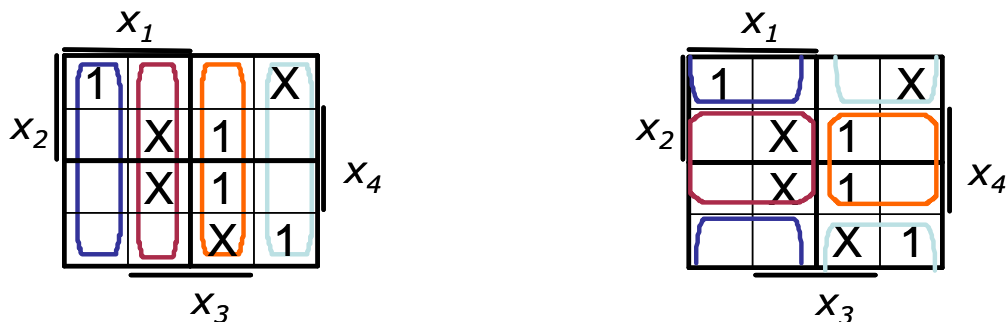
Če izberemo kot naslovni spremenljivki x_1 in x_3 , dobimo levi Veitchev diagram,

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

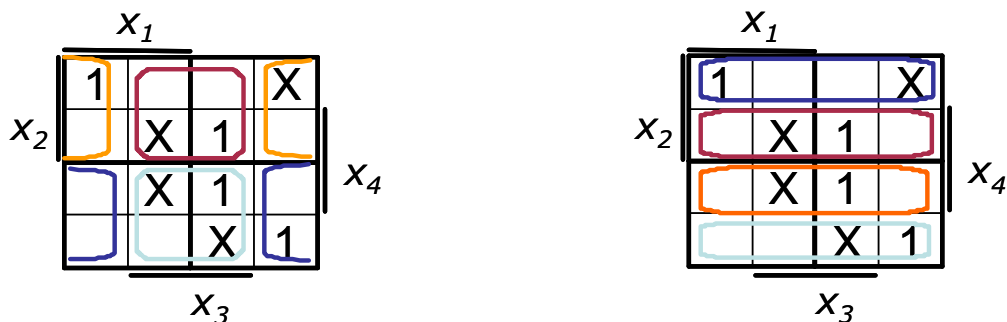
Na list z rešitvami se podpišite in napišite še vpisno številko ter kateri predmet pišete

Rezultati bodo objavljeni na: <https://studij.fe.uni-lj.si>

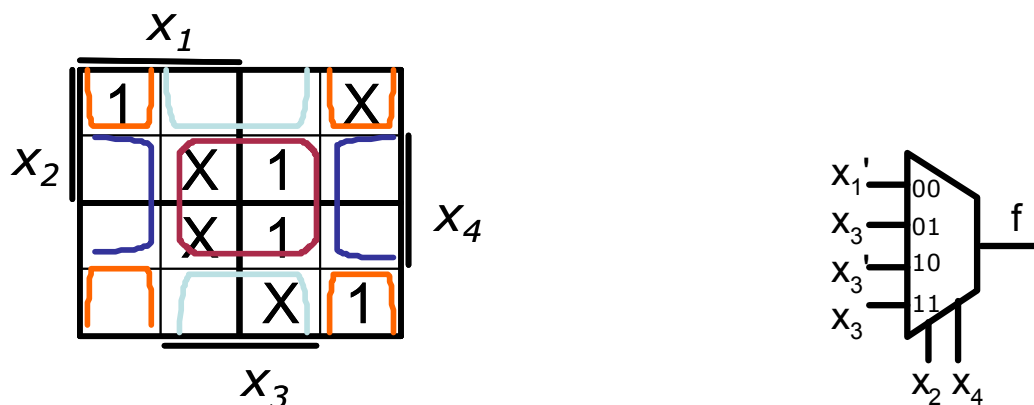
če x_1 in x_4 , pa desnega. Podobno kot v prejšnjem primeru poiščemo realizacije ustreznih kvadratov in iščemo najenostavnejšo realizacijo: Izigibamo se veliko različnim funkcijam in iščemo drugače kvadratov, ki vsebujejo konstante (samo '1' ali samo '0'). Pri razvoju po x_1 in x_3 imamo pri $x_1x_3="10"$ najneugodnejšo funkcijo, saj vsebuje eno samo '1'. Pri razvoju po x_1 in x_4 nastopa ena sama '1' pri kombinaciji $x_1x_4="10"$.



Nato izberemo naslovni spremenljivki x_2 in x_3 , (levi diagram) in x_2 in x_4 (desni diagram). Pri razvoju po x_2 in x_3 imamo pri $x_2x_3="00"$ najneugodnejšo funkcijo (moder), saj vsebuje eno samo '1'. Pri razvoju po x_2 in x_4 nikjer nimamo osamljene '1', zato se dajo funkcijski ostanki enostavno realizirati, če vse redundance postavimo na '1'.



Zadnja kombinacija naslovnih vhodov je x_3 in x_4 . Pri razvoju po x_3 in x_4 imamo pri $x_3x_4="00"$ najneugodnejšo funkcijo (oranžen), saj dve '1' opišemo s funkcijo ekvivalence. Možno rešitev torej predstavlja kombinacija naslovnih vhodov x_2 in x_4 .



Vezje izbiralnika je v predlogah avditornih vaj na domači strani predmeta:

Logisim\MUX\mux_4_1_f_V_0_3_7_12_in_Vx_2_4_11_15.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

Rezultati bodo objavljeni na: <https://studij.fe.uni-lj.si>

Rešitev 2. naloge:

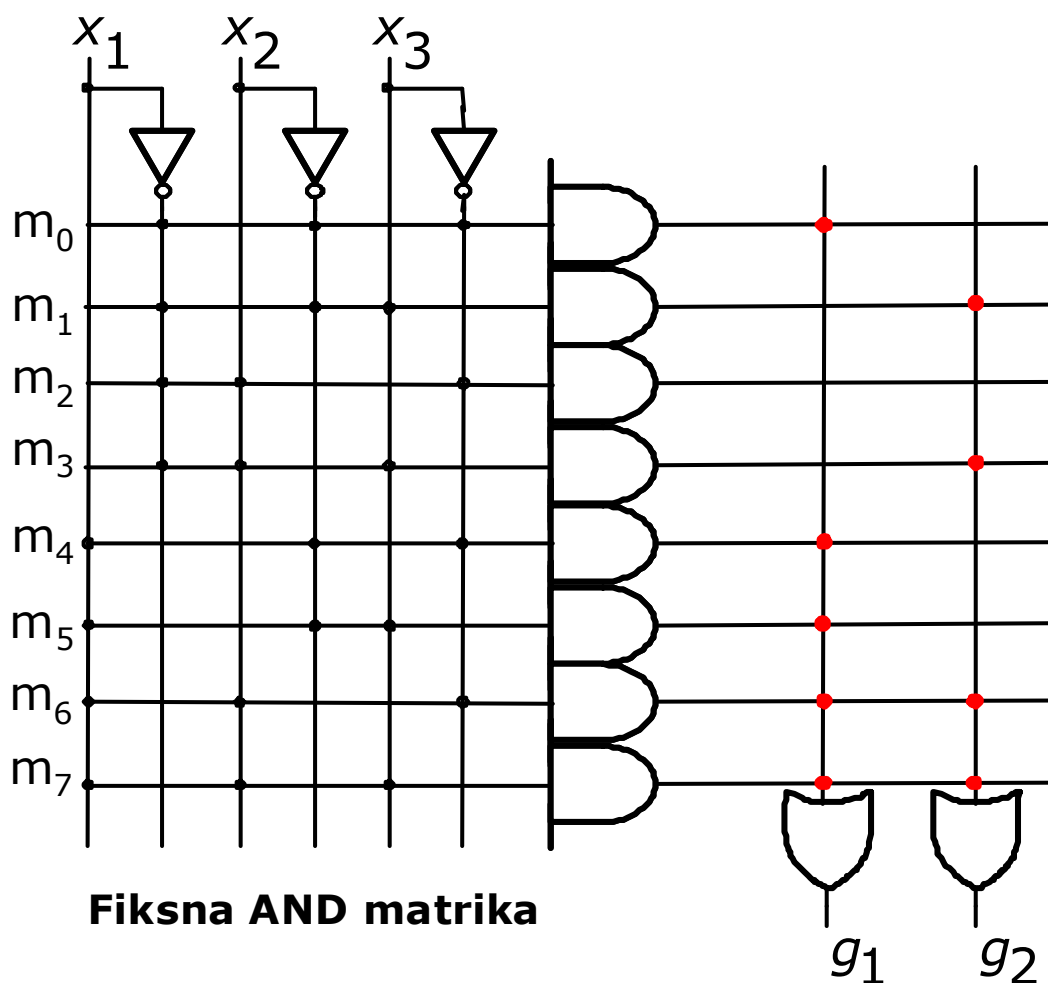
Če se funkcije ne nahajajo v popolni disjunktivni normalni obliki (PDNO), jih prevedemo v to obliko z uporabo pravil Boole-ove algebre. Funkcijo lahko tudi izpišemo v Veitch-ev diagram in izpišemo številke mintermov, kjer je funkcija enaka '1'.

$$\begin{aligned}g_1(x_1, x_2, x_3) &= x_1 + \overline{x_2} \cdot \overline{x_3} = x_1 \cdot (\overline{x_2} \cdot \overline{x_3} + x_2 \cdot \overline{x_3} + \overline{x_2} \cdot x_3 + x_2 \cdot x_3) + (\overline{x_1} + x_1) \cdot \overline{x_2} \cdot \overline{x_3} \\g_1(x_1, x_2, x_3) &= x_1 \cdot \overline{x_2} \cdot \overline{x_3} + x_1 \cdot x_2 \cdot \overline{x_3} + x_1 \cdot \overline{x_2} \cdot x_3 + x_1 \cdot x_2 \cdot x_3 + \overline{x_1} \cdot \overline{x_2} \cdot \overline{x_3} + x_1 \cdot \overline{x_2} \cdot \overline{x_3} \\g_1(x_1, x_2, x_3) &= x_1 \cdot \overline{x_2} \cdot \overline{x_3} + x_1 \cdot x_2 \cdot \overline{x_3} + x_1 \cdot \overline{x_2} \cdot x_3 + x_1 \cdot x_2 \cdot x_3 + \overline{x_1} \cdot \overline{x_2} \cdot \overline{x_3} \\g_1(x_1, x_2, x_3) &= V(4, 6, 5, 7, 0)\end{aligned}$$

Podobno storimo še za preostale funkcije:

$$\begin{aligned}g_2 &= \overline{x_1} \cdot x_3 + x_1 \cdot x_2 = \overline{x_1} \cdot (\overline{x_2} + x_2) \cdot x_3 + x_1 \cdot x_2 \cdot (\overline{x_3} + x_3) \\g_2(x_1, x_2, x_3) &= \overline{x_1} \cdot \overline{x_2} \cdot x_3 + \overline{x_1} \cdot x_2 \cdot x_3 + x_1 \cdot x_2 \cdot \overline{x_3} + x_1 \cdot x_2 \cdot x_3 \\g_2(x_1, x_2, x_3) &= V(1, 3, 6, 7)\end{aligned}$$

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.

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 na: <https://studij.fe.uni-lj.si>

Rešitev 3. naloge:

Postopek sinteze zahteva, da zapišemo tabelo prehajanja stanj števec:

SMER	Q ₂	Q ₁	Q ₀	Q ₂	Q ₁	Q ₀	T ₂	T ₁	T ₀
0	0	0	0	0	0	1	0	0	1
0	0	0	1	0	1	0	0	1	1
0	0	1	0	0	1	1	0	0	1
0	0	1	1	1	0	0	1	1	1
0	1	0	0	1	0	1	0	0	1
0	1	0	1	1	1	0	0	1	1
0	1	1	0	1	1	1	0	0	1
0	1	1	1	0	0	0	1	1	1
1	0	0	0	1	1	1	1	1	1
1	0	0	1	0	0	0	0	0	1
1	0	1	0	0	0	1	0	1	1
1	0	1	1	0	1	0	0	0	1
1	1	0	0	0	1	1	1	1	1
1	1	0	1	1	0	0	0	0	1
1	1	1	0	1	0	1	0	1	1
1	1	1	1	1	1	0	0	0	1

Normalna analiza bi zahtevala, da narišemo Veitch–eve diagrame za štiri spremenljivke za vsak vhod T–FF, vendar ker so T–FF po svoji naravi primerni za realizacijo števec, so praviloma njihove vhodne enačbe zelo enostavne. Iz tabele prehajanja stanj števec določimo enačbe T–FF: Iz stolpca T₀ se vidi, da je T₀='1'. Iz stolpca T₁ se vidi, da se ponavlja

vzorec 01, če je SMER='0' in 10, če je SMER='1'.

SMER	T ₁
0	Q ₀
1	Q ₀ '

kar lahko kratko zapišemo kot:

$$T_1 = \text{SMER} \cdot \overline{Q_0} + \overline{\text{SMER}} \cdot Q_0 = \text{SMER} \oplus Q_0$$

Za T₂ se da enostavno ugotoviti realizacijo iz Veitch–evega diagrama:

		SMER				
		1	0	0	0	
Q ₂	Q ₁	0	0	1	0	Q ₀
		0	0	1	0	
		1	0	0	0	

$$T_2 = \text{SMER} \cdot \overline{Q_1} \cdot \overline{Q_0} + \overline{\text{SMER}} \cdot Q_1 \cdot Q_0$$

V enačbi za T₂ poiščemo podobnosti z enačbo za T₁: Enačba za T₁ vsebuje konjunkciji SMER·Q₀' in SMER'·Q₀, ki sta vsebovani tudi v enačbi za T₂, kar nam dodatno poenostavi realizacijo števec. Obenem nam taka realizacija nakazuje osnovno strukturo, ki jo lahko s ponavljanjem razširimo v večbitni dvosmerni sinhroni števec.

Primer podobnega vezja 4-bitnega dvojiškega dvosmernega števec, ki ima še vzporedno nalaganje je 74191¹. Če boste primerjali našo realizacijo in realizacijo v podatkovnem listu, boste opazili, da je v dejanski realizaciji 74191 precej več večvhodnih AND vrat: Delno je razlog za to v dodani logiki za vzporedno nalaganje, delno pa tudi zato, da zagotovimo enakomerno zakasnitev med posameznimi stopnjami števec.

¹ <http://www.alldatasheet.com/view.jsp?Searchword=74191>

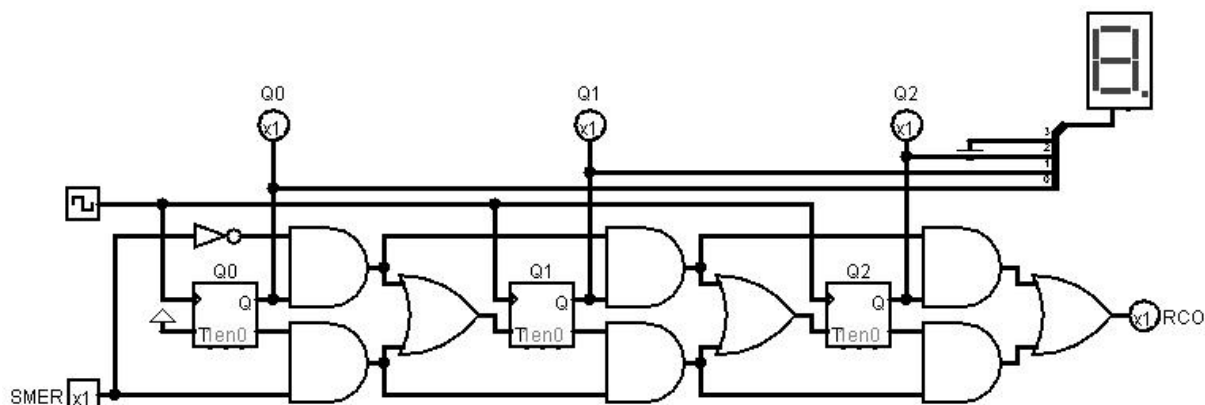
Ko enkrat narišemo vezje dvosmernega števca, zelo spominja na združitev sinhronnega števca za štetje navzgor in sinhronnega števca za štetje navzdol: Če bi števec vseboval samo zgornja AND vrata (vezanih neposredno na T vhod – brez OR) bi bil to števec navzgor, če pa samo spodnja AND bi bil števec navzdol. Signal SMER določa katera AND vrata so omogočena:

- zgornja AND vrata, ko je SMER='0' – štejemo naraščajoče,
- spodnja AND vrata, ko je SMER='1' – štejemo padajoče.

Pri tovrstnih števcih želimo realizirati tudi signal za proženje naslednjih stopenj števca (RCO – oz. ripple carry out, včasih tudi TC – terminal count). RCO je signal, ki postane '1' ob prehodu iz najvišjega stanja števca (v našem primeru je to "111") v stanje "000" pri štetju navzgor in ob prehodu "000" v najvišje stanje števca pri štetju navzdol:

SMER	RCO
0	$Q_0 \cdot Q_1 \cdot Q_2$
1	$Q_0' \cdot Q_1' \cdot Q_2'$

Tak signal uporabljamo pri realizaciji večbitnih števcov tako, da izdelane 3 bitne števec vežemo kaskadno – torej da signal RCO vežemo na EN signal naslednjega vezja. Za realizacijo takega signala bi narisali enako kombinacijo AND in OR vrat še na izhodu Q_2 , kot kaže spodnja slika:



Opis delovanja in vezje števca je v predlogah vaj na domači strani predmeta v imeniku Logisim\counter\ counter_up_down_3_bit_using_T_FF.circ

Večina števcov je realizirana v 4-bitni zasnovi, tako da glede na vrednost RCO signala ločimo dve skupini števcov:

- desetiški (BCD) števci, katerih RCO se postavi na '1' takrat, ko števec preide iz stanja "1001" v "0000" in
- dvojiški (binarni), katerih RCO se postavi na '1' takrat, ko števec preide iz stanja "1111" v "0000". Več o delovanju RCO najdete v opisu delovanja števcov 74161².

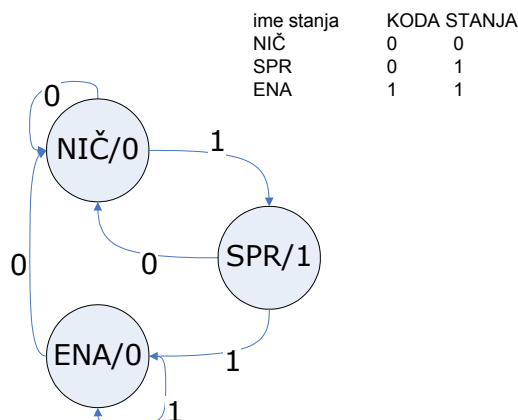
² <http://www.alldatasheet.com/view.jsp?Searchword=74161>

Rešitev 4. naloge:

Moore-ova realizacija avtomata končnih stanj.

Diagram stanj:

Stanja avtomata določimo glede na to, v katerem stanju se je vhodni signal nahajal v



prejšnji periodi signala ure CLK. Ta stanja so NIČ, ENA in SPR, ki pomeni spremembo signala. Če se signal nahaja v stanju NIČ, potem ob logični '1' na vhodu preidemo v stanje sprememba (SPR). Izhod vezja postane '1' torej ob naslednji periodi signala ure. Stanje SPR traja natanko eno periodo signala ure, zato ob vhodu $x=0$ preidemo v stanje '0', ob vhodu '1' pa v stanje ENA. V obeh primerih je izhodni signal v teh stanjih enak '0'. V vsakem od teh

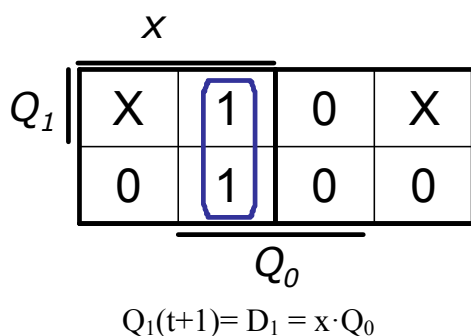
stanj vztrajamo, dokler ne pride do ponovne spremembe vhodnega signala. Iz povedanega sledi, da potek izhodnega signala y_1 ustreza Moore-ovi izvedbi avtomata.

Narišemo tabelo prehajanja stanj

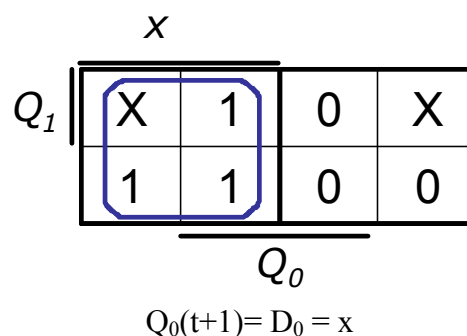
vhod in trenutno stanje			naslednje stanje		enačbe FF in izhoda		
x	Q_1	Q_0	Q_1	Q_0	D_1	D_0	y
0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	1
0	1	0	X	X	X	X	X
0	1	1	0	0	0	0	0
1	0	0	0	1	0	1	0
1	0	1	1	1	1	1	1
1	1	0	X	X	X	X	X
1	1	1	1	1	1	1	0

Iz tabele prehajanja stanj avtomata določimo enačbe D-FF:

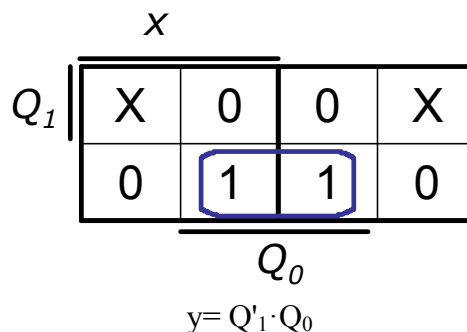
Za Q_1 narišemo Veitchev diagram



Podobno za Q_0 narišemo Veitchev diagram



In še za izhod y:



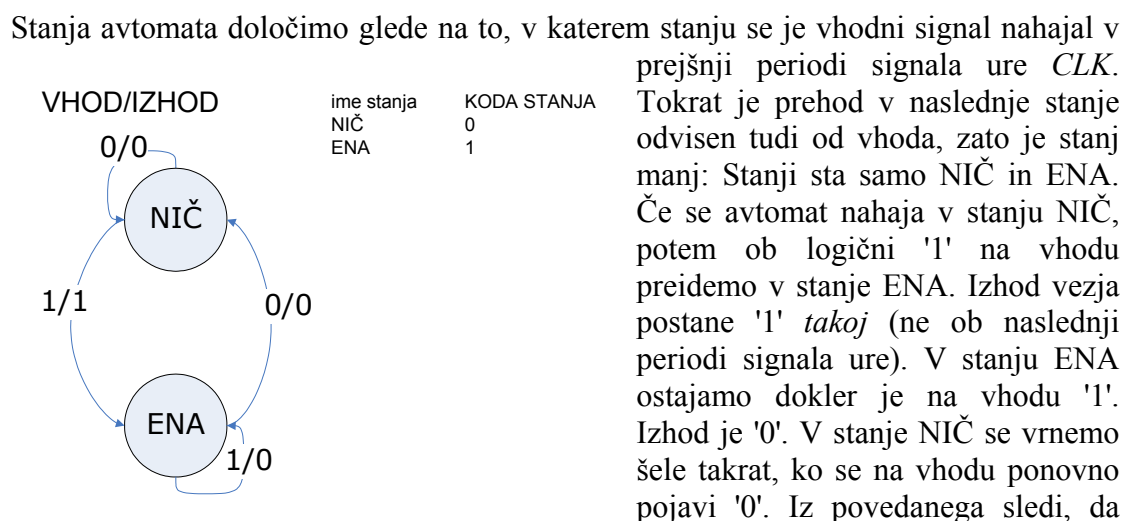
Čas pisanja je 60 minut. Vsaka naloga je vredna 10 točk.

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>

Mealy–eva realizacija avtomata končnih stanj.

Diagram stanj:



potek izhodnega signala y_2 ustreza Mealy–evi izvedbi avtomata.

Narišemo tabelo prehajanja stanj:

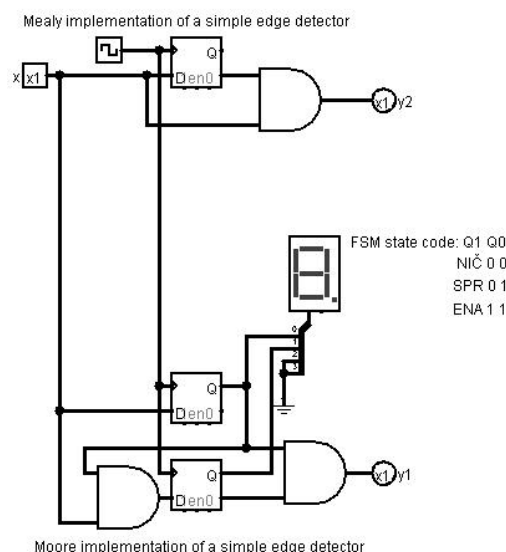
vhod in trenutno stanje		naslednje stanje	D–FF in izhod	
x	Q	Q	D	y
0	0	0	0	0
0	1	0	0	0
1	0	1	1	1
1	1	1	1	0

Veitchevih diagramov nam ni treba risati, saj enačbi D–FF in izhoda neposredno sledita iz tabele.

$$Q(t+1)=x \quad \text{in} \quad y=x \cdot Q'(t)$$

Vezje se nahaja v Logisim predlogah rešenih nalog na domači strani predmeta:

Logisim\fsm\front_edge_detector_mealy_moore.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>