

# RAZVOJ DIGITALNIH SISTEMOV

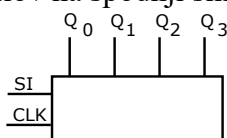
Izpit

21. 01. 2022

1. Realizirajte podano funkcijo  $f$  z redundantnimi makstermi s čim manj izbiralniki 4/1.

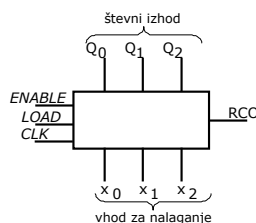
$$f(x_1, x_2, x_3, x_4) = \&(1, 5, 7, 8, 9, 11, 12) \text{ in } \&_x(3, 4, 10, 15)$$

2. Sestavite 4-bitni pomikalni register s T-celicami in izbiralniki 2/1. Register ima zaporedni vhod  $SI$  (ang. serial input), in vzporedni izhod ( $Q_0, Q_1, Q_2, Q_3$ ). Uporabite poimenovanje signalov na spodnji sliki.



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. Z uporabo D flip-flopov, ki so proženi na sprednji rob signala ure CLK, načrtajte Moore-ove avtomat končnih stanj, ki deluje kot krmilje za kavni avtomat. Kava stane 15 centov, plačujemo pa lahko s kovancema za 5 in 10 centov.

Krmilje ima:

- vhod  $5cent$ , ki postane '1', ko uporabnik vrže v avtomat kovanec za 5 centov,
- vhod  $10cent$  ki postane '1', ko uporabnik vrže v avtomat kovanec za 10 centov,
- izhod  $p$ , ki postane '1', ko uporabnik vrže v avtomat skupno 15 centov.

Avtomat ne vrača drobiža in se ob detekciji plačila 15 centov ne vrača nazaj v začetno stanje, ampak ostane v končnem stanju. Vnos dveh kovancev naenkrat ni mogoč.

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 PKNO z redundancami.

$$f(x_1, x_2, x_3, x_4) = \&(1, 5, 7 - 9, 11, 12) \text{ in } \&_x(3, 4, 10, 15)$$

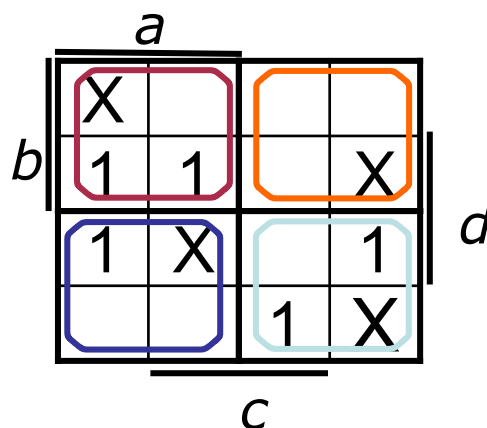
Za potrebe realizacije jo najprej pretvorimo v obliko PDNO. To storimo tako, da maksterme preslikamo v minterme. V pravilnostno tabelo funkcije najprej zapišemo številke mintermov ( $m$ ) in pripadajoče številke makstermov ( $M$ ). Vpišemo  $f=0$  za vse maksterme in  $f=X$  za vse redundantne maksterme.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

Dobimo:

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

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  $a$  in  $b$ , potem dobimo:

$m$	$M$	$a$	$b$	$c$	$d$	$f$
0	15	0	0	0	0	X
1	14	0	0	0	1	1
2	13	0	0	1	0	1
3	12	0	0	1	1	0
4	11	0	1	0	0	0
5	10	0	1	0	1	X
6	9	0	1	1	0	0
7	8	0	1	1	1	0
8	7	1	0	0	0	0
9	6	1	0	0	1	1
10	5	1	0	1	0	0
11	4	1	0	1	1	X
12	3	1	1	0	0	X
13	2	1	1	0	1	1
14	1	1	1	1	0	0
15	0	1	1	1	1	1



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.

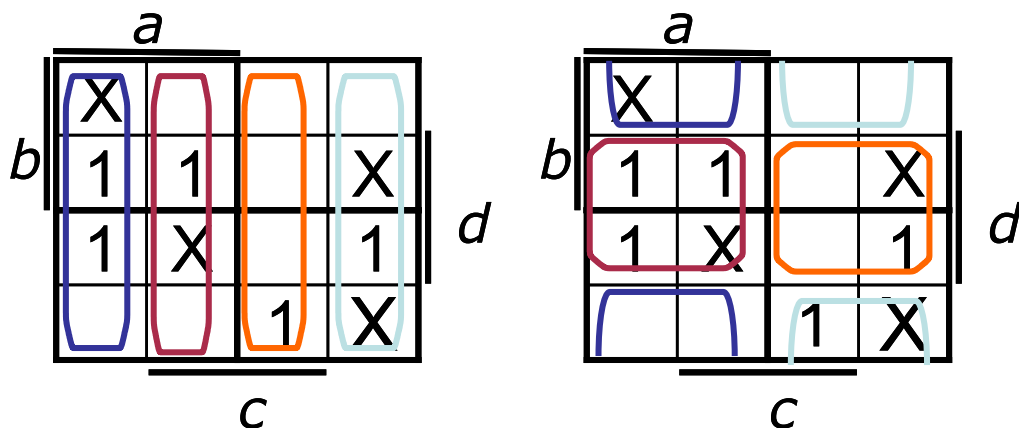
Če izberemo kot naslovni spremenljivki  $a$  in  $c$ , dobimo levi Veitchev diagram, če  $a$  in  $d$ , pa desnega. Podobno kot v prejšnjem primeru poiščemo realizacije ustreznih kvadratov in iščemo najcenejšo realizacijo: Izogibamo se veliko različnim funkcijam

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

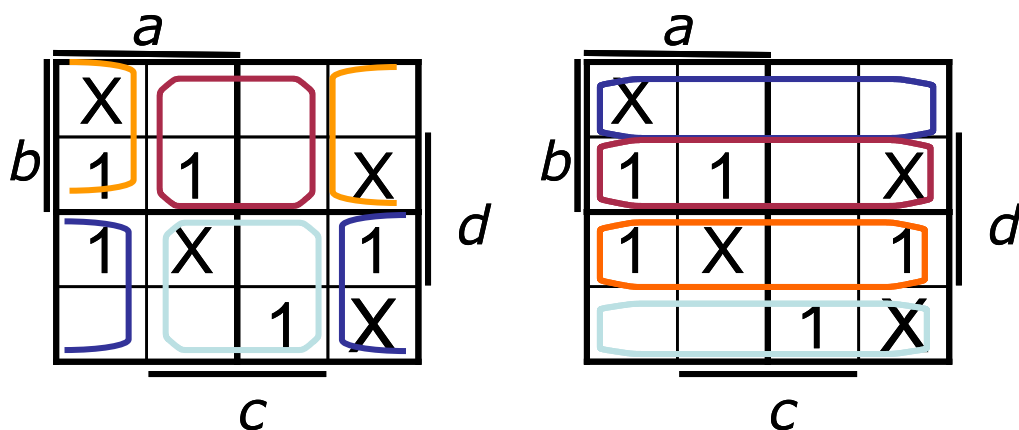
Na list z rešitvami se podpišite in napišite še vpisno številko ter kateri predmet pišete (VSŠ, UNI).

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

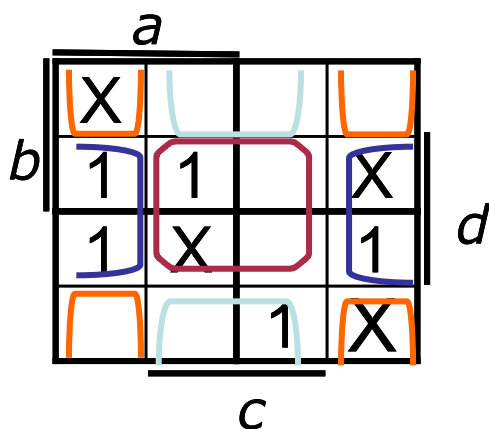
in iščemo inačice kvadratov, ki vsebujejo same '1' ali same '0'. Pri razvoju po a in c imamo pri ac="01" najneugodnejšo funkcijo, saj vsebuje eno samo '1'; medtem ko je razvoju po a in d nikjer ne nastopa ena sama '1' ali tri '1' ali diagonala (XOR) dveh '1'.



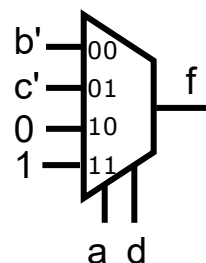
Nato izberemo naslovni spremenljivki b in c, (levi Veitchev diagram) in b in d (desni diagram). Pri razvoju po b in c imamo pri bc="11" najneugodnejšo funkcijo (rdeč), saj vsebuje eno samo '1'; medtem ko imamo pri razvoju po b in d pri bd="11" (rdeč) najneugodnejšo funkcijo, saj vsebuje tri '1'.



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:



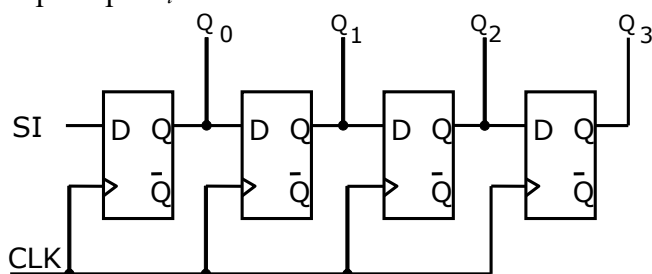
Čas pisanja je 60 minut. Vsaka naloga je vredna 10 točk.

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 2. 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  $D_i$ .



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

Če želimo pomikalni register sestaviti iz T–FF in 2/1 izbiralnikov, moramo pravzaprav realizirati celico D–FF s pomočjo T–FF in 2/1 izbiralnikov. V ta namen zapišemo tabelo D–FF, pri kateri dodamo izhodni stolpec  $T$  vhoda.

XOR vrata moramo realizirati s pomočjo 2/1 izbiralnikov, zato zapišemo enačbo XOR funkcije:

$$f = x \oplus y = \bar{x} \cdot y + x \cdot \bar{y}$$

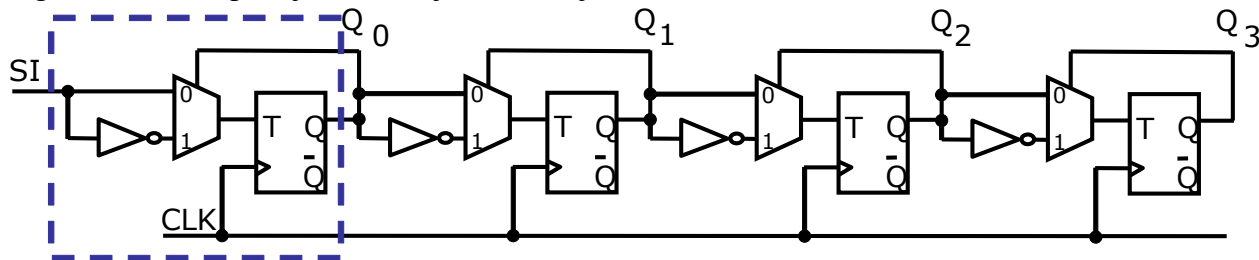
$D$	$Q(t)$	$Q(t+1)$	$T$
0	0	0	0
0	1	0	1
1	0	1	1
1	1	1	0

Funkcijo  $f$  realiziramo z izbiralnikom tako, da naredimo razvoj po spremenljivki  $x$  in dobimo:

$x$	$f$
0	$y$
1	$y'$

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

Če nastali D–FF iz T–FF in 2/1 izbiralnika sestavimo skupaj v 4–bitni pomikalni register dobimo spodnjo realizacijo, v kateri je izvedba D–FF označena črtkano.



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:

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<sub>0</sub> = Q<sub>0</sub>'.

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

D<sub>1</sub>:

	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<sub>2</sub>:

	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:

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

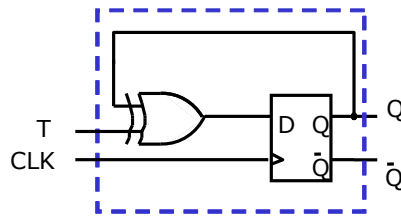
Upoštevamo definicijo XOR operacije (a ⊕ b = a' · b + a · b')

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

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".

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

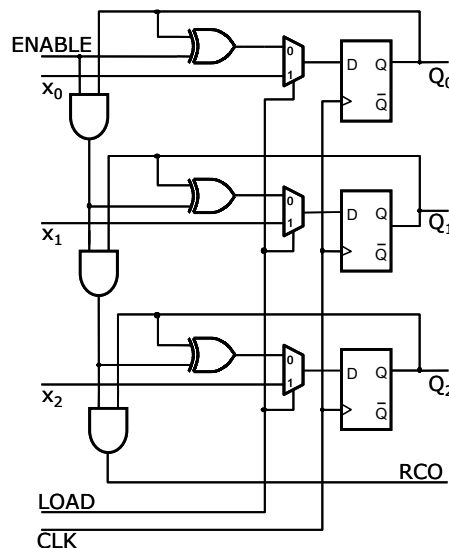
Naloga pravi, da moramo izdelati števec, ki ima vhod za omogočanje štetja (ENABLE). Če vezje analiziramo, vidimo, da smo pravzaprav realizirali T–FF s pomočjo D–FF in XOR vrat, kot kaže spodnja slika:



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 sinhronega 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 števnega 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 števnici 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.

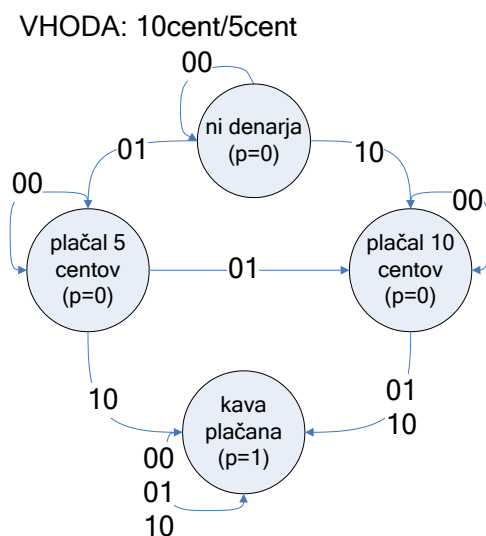
Čas pisanja je 60 minut. Vsaka naloga je vredna 10 točk.

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:

Moore-ova realizacija avtomata končnih stanj. Opis diagrama stanj:



vrgel bo izhod v teh dveh stanjih enak  $p=0$ , ker še ni plačal celotne cene kave. Če smo v stanju "plačal 5 centov" in uporabnik vrže v avtomat 10 centov (10cent/5cent=10), potem preide v stanje "kava plačana", kjer postavimo izhod ( $p=1$ ). Stanje "kava plačana" je končno in tam tudi ostanemo za vse možne kombinacije. Če smo v stanju "plačal 10 centov" in uporabnik vrže v avtomat 5 ali 10 centov (10cent/5cent=10 oz. 01), potem podobno preidemo v stanje "kava plačana", kjer postavimo izhod ( $p=1$ ).

Naredimo tabelo prehajanja stanj:

trenutno stanje	10cent	5cent	naslednje stanje	izhod p
ni denarja	0	0	ni denarja	0
ni denarja	0	1	plačal 5 centov	0
ni denarja	1	0	plačal 10 centov	0
ni denarja	1	1	X	X
plačal 5 centov	0	0	plačal 5 centov	0
plačal 5 centov	0	1	plačal 10 centov	0
plačal 5 centov	1	0	kava plačana	0
plačal 5 centov	1	1	X	X
plačal 10 centov	0	0	plačal 10 centov	0
plačal 10 centov	0	1	kava plačana	0
plačal 10 centov	1	0	kava plačana	0
plačal 10 centov	1	1	X	X
kava plačana	0	0	kava plačana	1
kava plačana	0	1	kava plačana	1
kava plačana	1	0	kava plačana	1
kava plačana	1	1	X	X

Izberemo kodiranje stanj:

stanje	$Q_1$	$Q_0$
ni denarja	0	0
plačal 5 centov	0	1
plačal 10 centov	1	0
kava plačana	1	1

Na začetku se nahajamo v stanju "ni denarja", v katerem je izhod  $p=0$ . Vhoda v avtomat sta dva: 10cent in 5cent, kar na diagramu kodiramo kot 10cent/5cent.

Mehanizem za vnos kovancev preprečuje hkraten vnos dveh kovancev, torej je kombinacija (10cent/5cent=11) nemogoča, zato bo avtomat od tu lahko prešel v poljubno stanje (X). Če uporabnik ni vrgel denarja v avtomat (10cent/5cent=00), potem ostaja v stanju "ni denarja". Če uporabnik vrže v avtomat 5 centov (10cent/5cent=01), potem preide v stanje "plačal 5 centov". Če uporabnik vrže v avtomat 10 centov (10cent/5cent=10), potem preide v stanje "plačal 10 centov". Ne glede na to koliko je

Nad tabelo prehajanja stanj uporabimo predlagano kodiranje stanj:

$Q_1$	$Q_0$	10cent	5cent	$Q_1$	$Q_0$	izhod p
0	0	0	0	0	0	0
0	0	0	1	0	1	0
0	0	1	0	1	0	0
0	0	1	1	X	X	X
0	1	0	0	0	1	0
0	1	0	1	1	0	0
0	1	1	0	1	1	0
0	1	1	1	X	X	X
1	0	0	0	1	0	0
1	0	0	1	1	1	0
1	0	1	0	1	1	0
1	0	1	1	X	X	X
1	1	0	0	1	1	1
1	1	0	1	1	1	1
1	1	1	0	1	1	1
1	1	1	1	X	X	X

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>

Naloga zahteva realizacijo z D-FF:

t				t+1				
Q <sub>1</sub>	Q <sub>0</sub>	10cent	5cent	Q <sub>1</sub>	Q <sub>0</sub>	D <sub>1</sub>	D <sub>0</sub>	izhod p
0	0	0	0	0	0	0	0	0
0	0	0	1	0	1	0	1	0
0	0	1	0	1	0	1	0	0
0	0	1	1	X	X	X	X	X
0	1	0	0	0	1	0	1	0
0	1	0	1	1	0	1	0	0
0	1	1	0	1	1	1	1	0
0	1	1	1	X	X	X	X	X
1	0	0	0	1	0	1	0	0
1	0	0	1	1	1	1	1	0
1	0	1	0	1	1	1	1	0
1	0	1	1	X	X	X	X	X
1	1	0	0	1	1	1	1	1
1	1	0	1	1	1	1	1	1
1	1	1	0	1	1	1	1	1
1	1	1	1	X	X	X	X	X

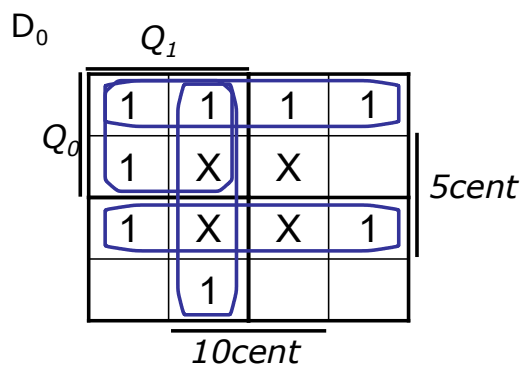
Iz dobljene tabele narišemo Veitch-eve diagrame za oba D-FF in izhod p:

$$D_0 = V(1, 4, 6, 9, 10, 12-14) \text{ in } Vx(3, 7, 11, 15)$$

$$D_1 = V(2, 5, 6, 8, 9, 10, 12-14) \text{ in } Vx(3, 7, 11, 15)$$

$$p = V(12-14) \text{ in } Vx(3, 7, 11, 15)$$

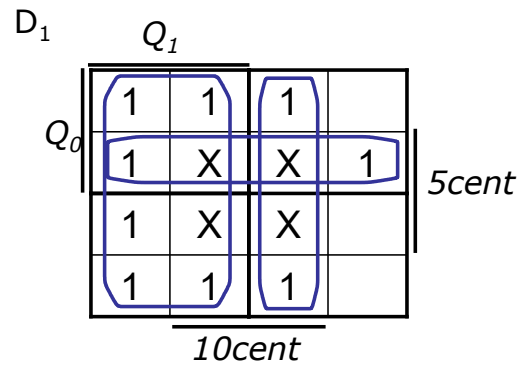
Veitch-ev diagram za D<sub>0</sub>:



$$D_0 = Q_1 \cdot Q_0 + \overline{5cent} \cdot Q_0 + 10cent \cdot Q_1 + 5cent \cdot \overline{Q_0}$$

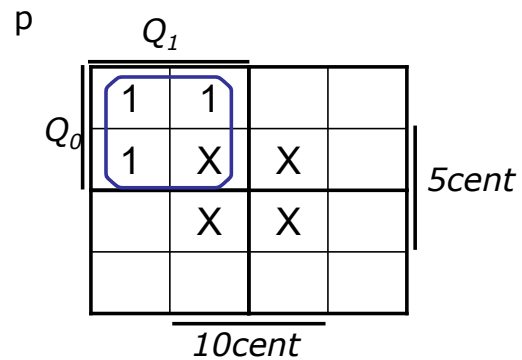
$$D_0 = Q_1 \cdot (Q_0 + 10cent) + 5cent \oplus Q_0$$

Veitch-ev diagram za D<sub>1</sub>:



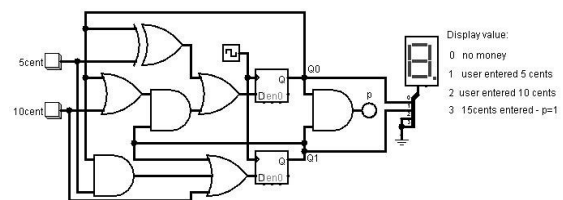
$$D_1 = Q_1 + 5cent \cdot Q_0 + 10cent$$

Veitch-ev diagram za izhod p:



$$p = Q_1 \cdot Q_0$$

Enačbo za izhod p bi lahko napisali tudi samo s sklepanjem, saj se izhod p postavi samo, ko je avtomat v stanju "kava plačana", ki ima kodo Q<sub>1</sub>Q<sub>0</sub>="11" – torej ko bosta Q<sub>1</sub> in Q<sub>0</sub> enaka '1', bo izhod p=1.



Vezje se nahaja v Logisim predlogah rešenih nalog na domači strani predmeta: Logisim\fsm\Vending\_machine\_d\_ff\_5\_10\_cents\_price\_15cents.circ