

# RAZVOJ DIGITALNIH SISTEMOV

Izpit

05. 01. 2019

1. Določite popolno konjunktivno normalno obliko (PKNO) in popolno disjunktivno normalno obliko (PDNO) funkcije  $f$ .

$$f(x_1, x_2, x_3, x_4) = (x_1 \downarrow x_2) \cdot \overline{x_3} + ((\overline{x_2} \equiv x_4) \downarrow \overline{x_1})$$

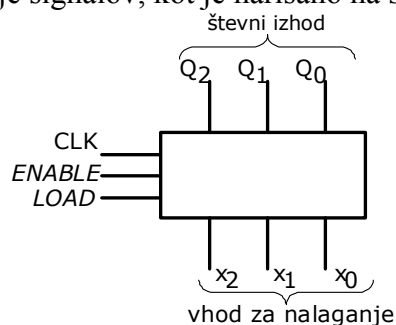
2. Realizirajte podano funkcijo  $f$  z redundancami s čim manj 4-bitnimi aritmetičnimi–logičnimi enotami (ALU). Negacije vhodnih spremenljivk izvedite z ALU.

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

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. Logika vseh krmilnih signalov je pozitivna.

S tovrstnim števcem realizirajte števec, ki šteje 2, 3, 4, 5, 2, 3, 4, 5 ...

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 in
- vhod *10cent* ki postane '1', ko uporabnik vrže v avtomat kovanec za 10 centov ter
- 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.

## Rešitev 1. naloge

Funkcija je zapisana v večnivojski obliki, torej jo izrazimo v normalno obliko.

$$f(x_1, x_2, x_3, x_4) = (x_1 \downarrow x_2) \cdot \overline{x_3} + ((\overline{x_2} \equiv x_4) \downarrow \overline{x_1})$$

Funkciji NOR ( $\downarrow$ ) in ekvivalence ( $\equiv$ ) izpišemo:

$$f(x_1, x_2, x_3, x_4) = (\overline{x_1 + x_2}) \cdot \overline{x_3} + \left( \overline{(\overline{x_2 \oplus x_4}) + x_1} \right)$$

Ekvivalenco smo izrazili kot negacijo XOR. Uporabimo De Morganov teorem:

$$f(x_1, x_2, x_3, x_4) = \overline{x_1} \cdot \overline{x_2} \cdot \overline{x_3} + (\overline{x_2 \oplus x_4}) \cdot x_1$$

Izpišemo enačbo funkcije XOR ( $a \oplus b = \overline{a} \cdot b + a \cdot \overline{b}$ ) in dobimo:

$$f(x_1, x_2, x_3, x_4) = \overline{x_1} \cdot \overline{x_2} \cdot \overline{x_3} + (\overline{x_2} \cdot \overline{x_4} + x_2 \cdot x_4) \cdot x_1$$

Razširimo še zadnjo konjunkcijo in rezultat je oblika MDNO:

$$f(x_1, x_2, x_3, x_4) = \overline{x_1} \cdot \overline{x_2} \cdot \overline{x_3} + \overline{x_2} \cdot \overline{x_4} \cdot x_1 + x_1 \cdot x_2 \cdot x_4$$

Če uporabimo lastnost Boole-ove algebre ( $\overline{\overline{a}} + a = 1$ ) lahko zapišemo:

$$f(x_1, x_2, x_3, x_4) = \overline{x_1} \cdot \overline{x_2} \cdot \overline{x_3} \cdot x_4 + \overline{x_1} \cdot \overline{x_2} \cdot \overline{x_3} \cdot \overline{x_4} + x_1 \cdot \overline{x_2} \cdot \overline{x_3} \cdot x_4 + x_1 \cdot \overline{x_2} \cdot \overline{x_3} \cdot \overline{x_4} + x_1 \cdot x_2 \cdot x_3 \cdot x_4 + x_1 \cdot x_2 \cdot \overline{x_3} \cdot x_4$$

Kar lahko zapišemo v obliki PDNO:

$$f(x_1, x_2, x_3, x_4) = \overline{x_1} \cdot \overline{x_2} \cdot \overline{x_3} \cdot \overline{x_4} + \overline{x_1} \cdot \overline{x_2} \cdot \overline{x_3} \cdot x_4 + x_1 \cdot \overline{x_2} \cdot \overline{x_3} \cdot \overline{x_4} + x_1 \cdot \overline{x_2} \cdot \overline{x_3} \cdot x_4 + x_1 \cdot x_2 \cdot \overline{x_3} \cdot x_4 + x_1 \cdot x_2 \cdot x_3 \cdot x_4$$

$$f_{PDNO}(x_1, x_2, x_3, x_4) = V(0, 1, 8, 10, 13, 15)$$

PDNO pretvorimo v PKNO tako, da pregledamo manjkajoče minterme: 2, 3, 4, 5, 6, 7, 9, 11, 12, 14. Te minterme preslikamo preko tabele:

m <sub>i</sub>	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
M <sub>i</sub>	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

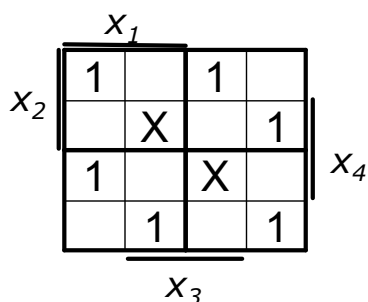
Funkcija v PKNO se torej glasi:

$$f_{PDNO}(x_1, x_2, x_3, x_4) = V(0, 1, 8, 10, 13, 15)$$

$$f_{PKNO}(x_1, x_2, x_3, x_4) = \&(13, 12, 11, 10, 9, 8, 6, 4, 3, 1)$$

## Rešitev 2. 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 ugotovljamo 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).

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  $1 \oplus 1 \oplus k_4=1 \rightarrow k_4=1$ .

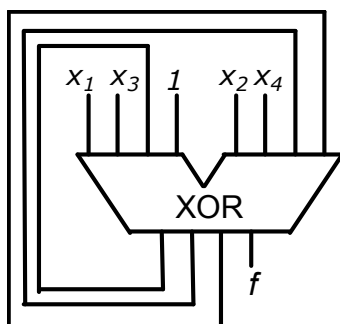
Analiziramo naprej in dobimo  $k_0 \oplus k_1 \oplus k_2=1$ , kar pomeni  $1 \oplus k_1 \oplus 1=0 \rightarrow k_1=1$ .

Vstavimo dobljene koeficiente v enačbo za splošno izražavo in dobimo:

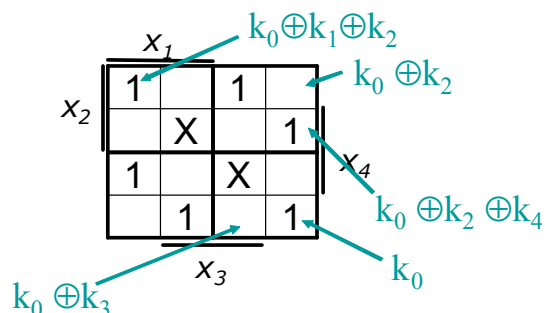
$$f(x_1, x_2, x_3, x_4) = 1 \oplus x_1 \oplus x_2 \oplus x_3 \oplus x_4$$

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.

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



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:

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

### Rešitev 3. naloge:

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

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:

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

Iz tabele prehajanja stanj števca določimo enačbe D-FF:

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

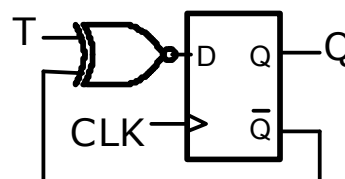
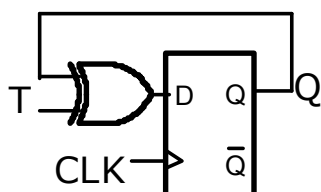
Za D<sub>1</sub> narišemo Veitchev diagram:

$D_1$ :

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

$$D_1 = Q_0 \oplus Q_1$$

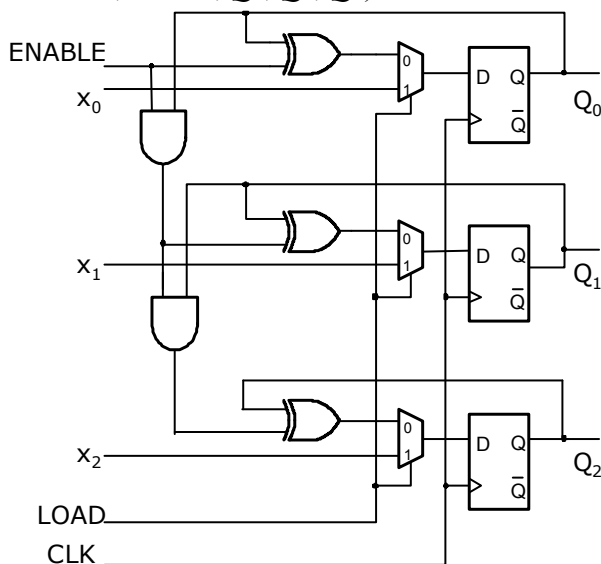
Če enačbo  $D_0 = Q_0'$  zapišemo kot  $D_0 = 1 \oplus Q_0$  in jo narišemo v vezju, smo pravzaprav realizirali T-FF s pomočjo D-FF, kot kaže levi del spodnje slike:



Slika 1: Realizacija T-FF s pomočjo D-FF in XOR vrat (levo) in XNOR vrat (desno)

Naloga pravi, da moramo izdelati števec, ki ima vhod za omogočanje štetja (*ENABLE*): Če prvemu "T-FF" (D-FF z XOR vrati) postavimo vhod  $T_0 = '0'$  namesto  $T_0 = '1'$ , vsi FF ne bodo štel, 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 FF vezani tako, da so odvisni od prvega 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 za 5 spremenljivk (*ENABLE*, *LOAD*,  $Q_2$ ,  $Q_1$ ,  $Q_0$ ).



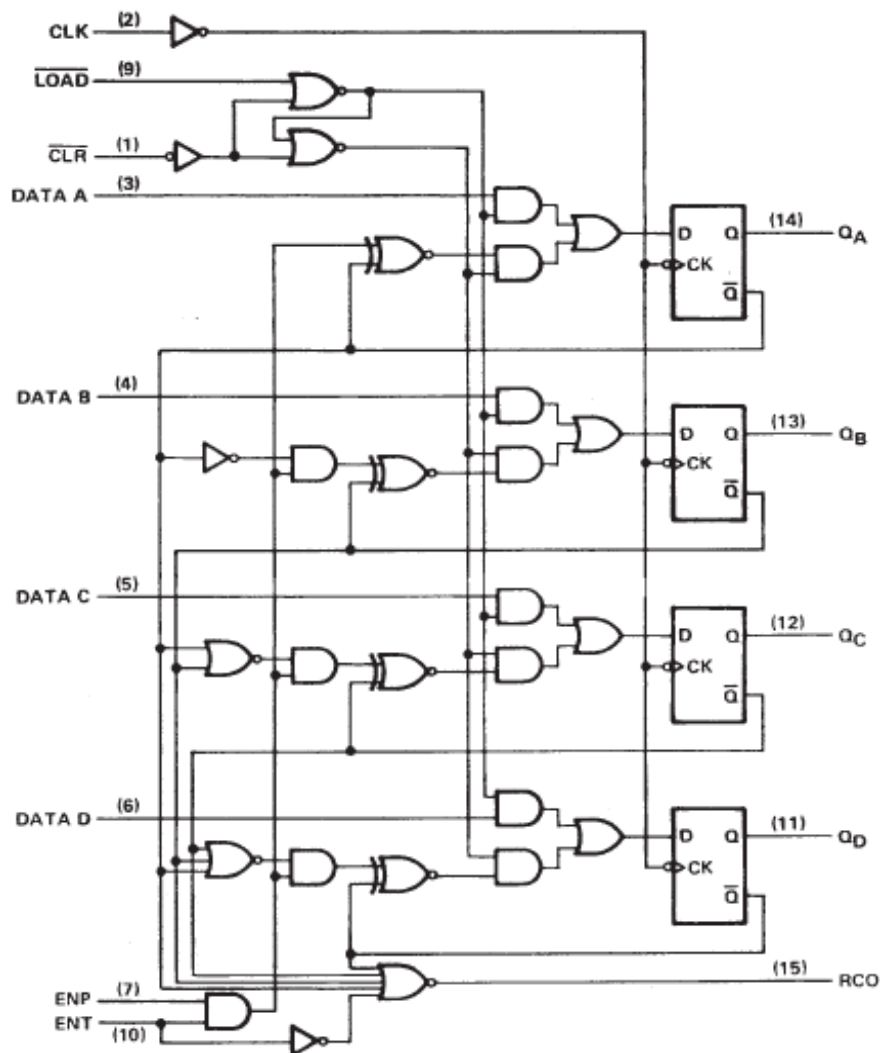
**Slika 2: Sinhroni števec z vzporednim nalaganjem (*LOAD*) in omogočanjem štetja (*ENABLE*) (3-bitna izvedba).**

Če želimo z nastalim števcem šteti naraščajoče 2, 3, 4, 5 ..., moramo ko števec pride do stanja 5 ( $Q_2Q_1Q_0=101_2$ ) števec postaviti nazaj na stanje v stanje 2 ( $Q_2Q_1Q_0=x_2x_1x_0=010_2$ ), torej signal *LOAD* dekodiramo s pomočjo 2–vhodnih 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 en signal *RESET*, s katerim postavimo števec v začetno stanje, kar dosežemo tako, da na vhod D–FF za brisanje asinhrono priključimo signal *RESET*.

Nastali strukturi števca bi lahko na isti način dodali še četrti bit. Tako bi dobili podobno strukturo kot je 4-bitni sinhroni števec z vzporednim nalaganjem 74163<sup>1</sup>. Pri spodnji realizaciji tega vezja so uporabljeni D–FF, proženi na negativni rob signala ure (*CLK*). Štetje omogočimo s signaloma *ENP*, *ENT* (ang. enable parallel, enable transfer). Štetje je izvedeno tako, da se D–FF spremenijo v T–FF, kar dosežemo s pomočjo XNOR vrat, ki imajo en vhod vezan na števeni signal ( $ENT=ENP='1'$ ), drug vhod pa na izhod  $Q'$  FF. AND vrata pred XNOR zagotavljajo prenehanje štetja, če velja  $ENT \cdot ENP \neq '1'$ . Na vhodu D–FF je vezan enostaven izbiralnik iz dveh AND vrat, vezanih na OR vrata. Ta izbiralnik določa, ali bo števec štel, ali bo vzporedno naložil vrednost. Zgornja *AND* vrata izbiralnika so krmiljena s signalom  $LOAD \cdot CLR'$ , spodnja pa z  $LOAD' \cdot CLR'$ . Čim velja, da je  $CLR' = '1'$  in  $LOAD' = '0'$ , se v D–FF asinhrono naloži vsebina na vseh  $Q_D Q_C Q_B Q_A = DATA_D DATA_C DATA_B DATA_A$ , medtem ko dokler velja  $LOAD' = '1'$  in  $CLR' = '1'$ , bo števec štel navzgor. Če je  $CLR' = '0'$ , je na obeh vseh vseh AND vrat izbiralnika '0' in stanje vseh D–FF se asinhrono postavi na  $Q_D Q_C Q_B Q_A = "0000"$ . Števeni izhodi so  $Q_D Q_C Q_B Q_A$ . Izhod *RCO* (*ripple carry out*) se postavi na '1', ko je števec v stanju "1111" ob tem, da

<sup>1</sup> <http://www.alldatasheet.com/view.jsp?Searchword=74163>

je  $ENT='1'$ . Signal RCO je izveden z NOR vrati na vhode katerih so priključeni negirani izhodi D-FF, kar je po De Morgan-ovem teoremu ekvivalent AND vratom.

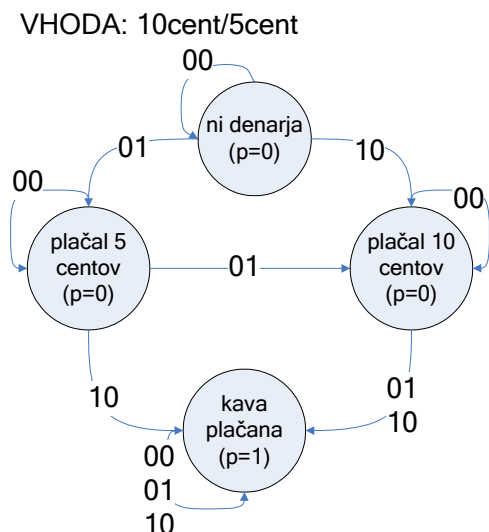


Slika 3: Struktura 4-bitnega MSI sinhronega števca z vzporednim nalaganjem (74163).

#### Rešitev 4. naloge:

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

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 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 <sub>1</sub>	Q <sub>0</sub>
--------	----------------	----------------

ni denarja	0	0
plačal 5 centov	0	1
plačal 10 centov	1	0
kava plačana	1	1

Nad tabelo prehajanja stanj uporabimo predlagano kodiranje stanj:

Q <sub>1</sub>	Q <sub>0</sub>	10cent	5cent	Q <sub>1</sub>	Q <sub>0</sub>	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

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

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

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

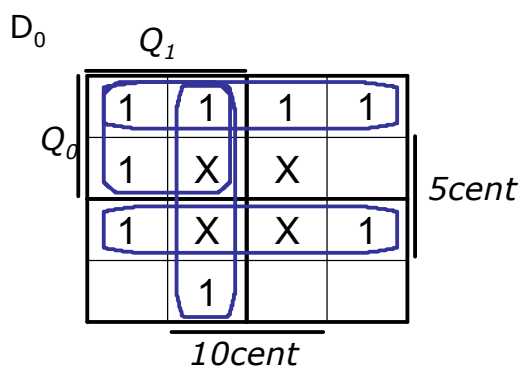
Iz dobljene tabele narišemo Veitch–eve diagrame za oba D–FF in izhod p:

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

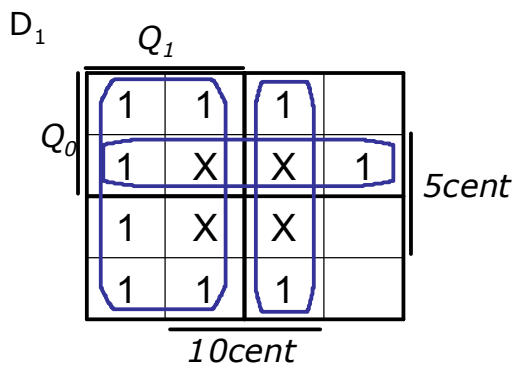
$$D_1 = V(2, 5, 6, 8, 9, 10, 12-14) \text{ in } Vx(3, 7, 11, 15)$$

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

Veitch–ev diagram za D<sub>0</sub>:

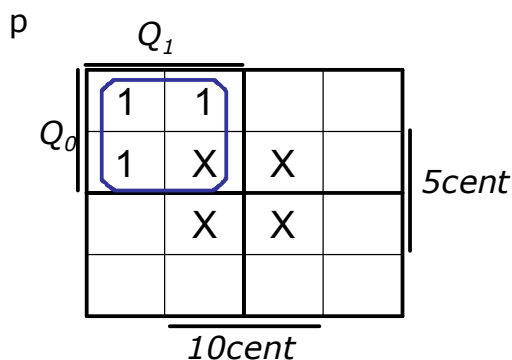


Veitch–ev diagram za D<sub>1</sub>:



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

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.

Iz dobljenih enačb minimizacije bi lahko narisali vezje krmilja.