

RAZVOJ DIGITALNIH SISTEMOV

Izpit

09. 09. 2024

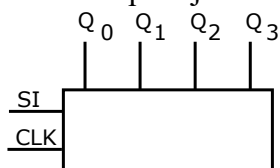
1. Izrazite podano logično funkcijo samo s Shefferjevimi operatorji. Morebitne negacije realizirajte s Shefferjevim operatorjem.

$$f(a,b,c) = \overline{\left((a+b) \downarrow \bar{c}\right)} \equiv (a \downarrow c)$$

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

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

3. 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.



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

<i>Trenutno stanje</i>	<i>Naslednje stanje</i>		<i>Izhod</i>
	$w=0$	$w=1$	
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

Rešitev 1. naloge

Realizacija s samimi Shefferjevimi (NAND, oziroma \uparrow) operatorji najprej zahteva pretvorbo funkcije v disjunktivno obliko (če se da, normalno).

$$f(a, b, c) = \overline{\left((a + b) \downarrow \bar{c} \right)} \equiv (a \downarrow c)$$

V ta namen moramo najprej operatorje (EQU, Pierce) v podani funkciji izpisati z disjunktijami in konjunkcijami. Začnemo na najvišjem nivoju, pri negaciji "čez vse" in EQU funkciji - oboje skupaj predstavlja XOR na najvišjem nivoju, ki jo izpišemo po definiciji XOR $\bar{x} \cdot y + x \cdot \bar{y} = x \oplus y$:

$$\begin{aligned} f(a, b, c) &= \left((a + b) \downarrow \bar{c} \right) \oplus (a \downarrow c) = \\ &= \left((a + b) \downarrow \bar{c} \right) \cdot \overline{(a \downarrow c)} + \overline{\left((a + b) \downarrow \bar{c} \right)} \cdot (a \downarrow c) \end{aligned}$$

Izpišemo še oba Pierceva (NOR) operatorja, pri čemer izničimo nastali dvojni negacijo drugega in tretjega člena ($x + y = x \downarrow y$):

$$f(a, b, c) = \overline{\left((a + b) + \bar{c} \right)} \cdot (a + c) + \left((a + b) + \bar{c} \right) \cdot \overline{(a + c)}$$

Uporabimo DeMorganov teorem nad negiranimi členi in nekaj ostalih osnovnih lastnosti Boole-ove logike ($x \cdot x = x$ in $x \cdot \bar{x} = 0$):

$$\begin{aligned} f(a, b, c) &= \overline{(a + b)} \cdot c \cdot (a + c) + \left((a + b) + \bar{c} \right) \cdot \bar{a} \cdot \bar{c} = \\ &= \bar{a} \cdot \bar{b} \cdot c \cdot (a + c) + (a + b + \bar{c}) \cdot \bar{a} \cdot \bar{c} = \\ &= \bar{a} \cdot \bar{b} \cdot c \cdot a + \bar{a} \cdot \bar{b} \cdot c \cdot c + \bar{a} \cdot b \cdot \bar{c} + \bar{a} \cdot \bar{c} = \bar{a} \cdot \bar{b} \cdot c \cdot a + \bar{a} \cdot \bar{b} \cdot c + \bar{a} \cdot b \cdot \bar{c} + \bar{a} \cdot \bar{c} = \\ &= \bar{a} \cdot \bar{b} \cdot c + \bar{a} \cdot \bar{b} \cdot \bar{c} + \bar{a} \cdot b \cdot \bar{c} + \bar{a} \cdot \bar{c} = \bar{a} \cdot \bar{b} + \bar{a} \cdot \bar{c} \end{aligned}$$

Dobili smo disjunktivno normalno obliko (DNO):

$$f(a, b, c) = \bar{a} \cdot \bar{b} \cdot c + \bar{a} \cdot \bar{c}$$

Dobljeno DNO pretvorimo v Shefferjevo (NAND) obliko tako, da dvakrat negiramo posamezne člene:

$$\begin{aligned} f(a, b, c) &= \bar{a} \cdot \bar{b} + \bar{a} \cdot \bar{c} = \overline{\overline{\bar{a} \cdot \bar{b}}} + \overline{\overline{\bar{a} \cdot \bar{c}}} \\ &= \overline{\overline{\bar{a} \cdot \bar{b}}} \uparrow \overline{\overline{\bar{a} \cdot \bar{c}}} = (\bar{a} \uparrow \bar{b}) \uparrow (\bar{a} \uparrow \bar{c}) \end{aligned}$$

Preostale negacije spremenljivk izrazimo z uporabo lastnosti $\bar{x} = x \uparrow x$:

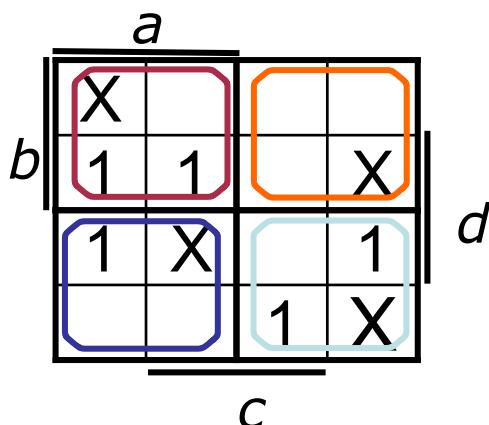
$$\begin{aligned} f(a, b, c) &= (\bar{a} \uparrow \bar{b}) \uparrow (\bar{a} \uparrow \bar{c}) \\ &= ((a \uparrow a) \uparrow (b \uparrow b)) \uparrow ((a \uparrow a) \uparrow (c \uparrow c)) \end{aligned}$$

Rešitev 2. naloge:

Funkcija f je podana v obliki PDNO z redundancami.

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

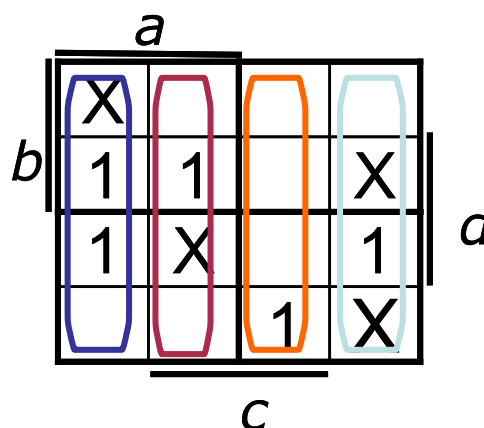
Dobljeno funkcijo vrišemo v Veitchev diagram. Ker iščemo najcenejšo realizacijo z izbiralnikom 4/1, bomo naredili razvoj po vseh kombinacijah naslovnih spremenljivk v Veitchevem diagramu. Če izberemo kot naslovni spremenljivki a in b , potem dobimo:



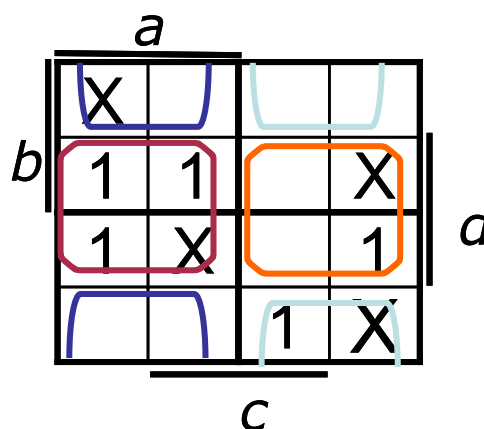
V zgornjem Veitchevem diagramu so označena vsa štiri polja štirih mintermov, če izberemo vhodni spremenljivki a in b . Zgornji levi kvadrat (rdeč) pomeni, da bo to polje izbrano ko bosta $ab="11"$, oranžni kvadrat ko bo $ab="01"$, temno modri ko bo $ab="10"$ in svetlo modri ko bo $ab="00"$. Vsakega od 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.

Čas pisanja je 60 minut. Vsaka naloga je vredna 10 točk. Na list z rešitvami se podpišite in napišite še vpisno številko. Rezultati bodo objavljeni na domači strani predmeta.

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 drugače 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'.

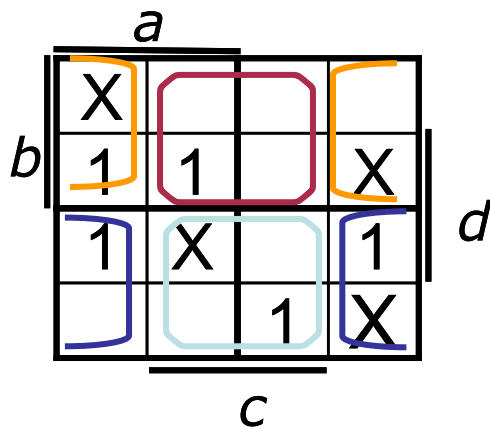


Pri razvoju po a in d nikjer ne nastopa ena sama '1' ali tri '1' ali diagonalna (XOR) dveh '1'.

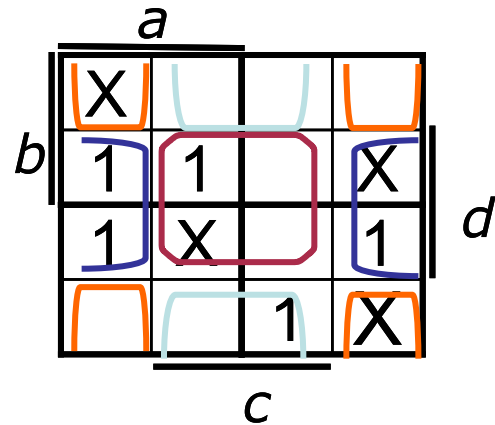
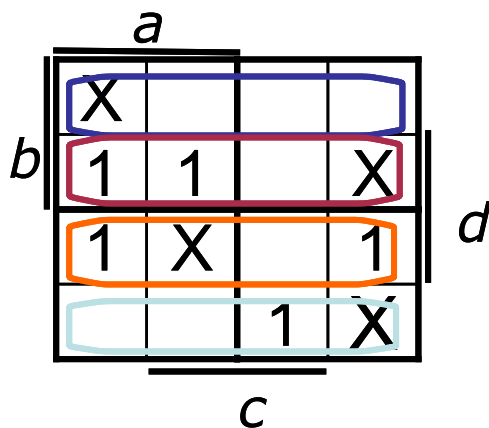


Nato izberemo naslovni spremenljivki b in c , (levi Veitchev diagram) in b in d (desni diagram). Pri razvoju po b in c

imamo pri $bc="11"$ najneugodnejšo funkcijo (rdeč), saj vsebuje eno samo '1'.

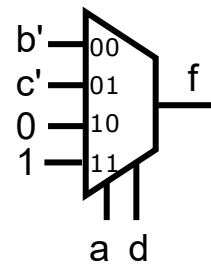


Pri razvoju po b in d dobimo eno (od dveh) možno realizacijo s funkcijskimi ostanki: $F_{00}=a'$; $F_{01}=c'$; $F_{10}=0$ in $F_{11}=a$.



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 .

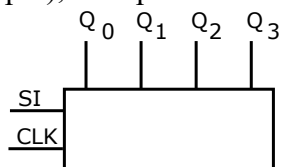
Končna realizacija funkcije je lahko:



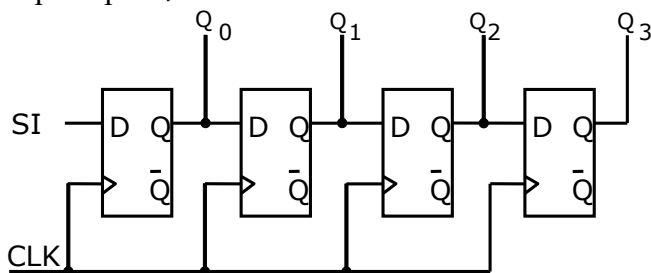
Druga možna rešitev je kombinacija naslovnih spremenljivk b in d .

Rešitev 3. naloge:

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)



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 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.

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

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

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

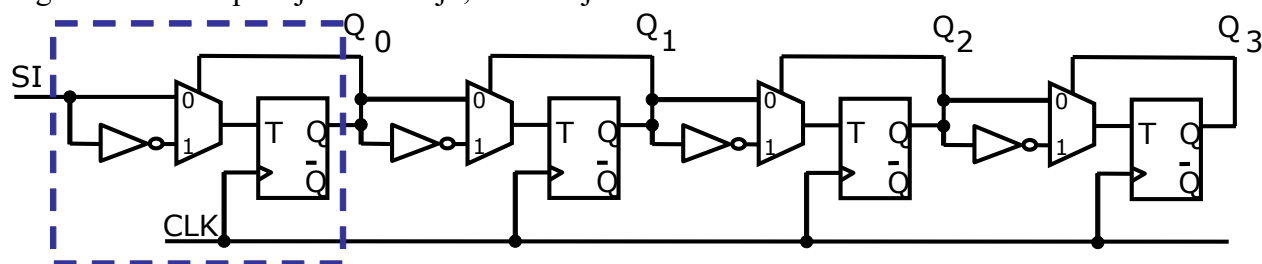
Iz tabele sledi, da je T vhod XOR operacija $Q(t)$ in vhoda D-FF, ki ga realiziramo.

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

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

x	f
0	y
1	y'

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.



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
<u>F</u>	E	<u>D</u>	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.

Rezultati bodo objavljeni na domači strani predmeta.

<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>