

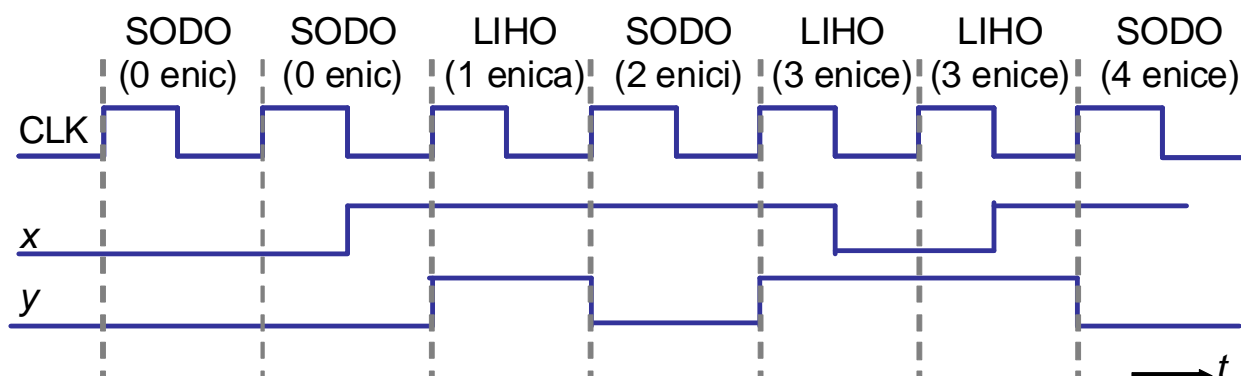
RAZVOJ DIGITALNIH SISTEMOV

Izpit 15. 04. 2011

1. Zgradite vezje, katerega izhod postane '1', ko se na vhodu pojavi eno izmed praštevil (2, 3, 5, 7, 11, 13). Števila na vhodu so kodirana s 4-bitno Gray-evo kodo. Za realizacijo uporabite samo NAND vrata. Kolikšna je COST funkcija realiziranega vezja?
2. Pretvorite število 999_{10} v dvojiški zapis z uporabo obratnega "double dabble" algoritma.
3. Realizirajte generator lihe parnosti (paritete) kot avtomat končnih stanj, ki šteje število enic v serijskem zaporedju bitov x na vhodu: Izhod vezja y naj bo '1', ko je na vhodu liho število enic in '0' ko je na vhodu sodo število enic. Ob resetu avtomata je število enic na vhodu sodo (nič enic). Za realizacijo uporabite D flip-flope, prožene na sprednji rob signala ure CLK .



Primer delovanja generatorja parnosti povzema spodnja slika:



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

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

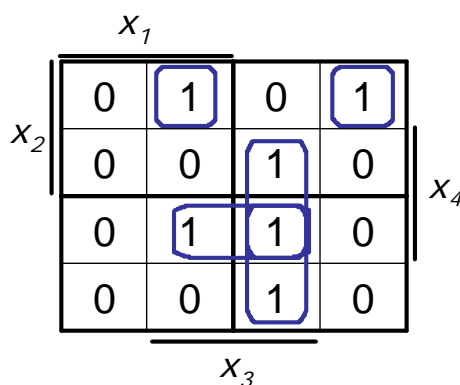
Na list z rešitvami se podpišite in napišite še vpisno številko ter kateri predmet pišete (VSŠ, UNI).

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

Rešitev 1. naloge:

S 4-bitno Grayevo kodo lahko kodiramo 16 števil (0 - 15). Najprej zapišemo pravilnostno tabelo v kateri desetiška števila kodiramo v Grayevi kodi (npr. $7_{10}=0100_{\text{Gray}}$). Izhod mora biti '1' vsakič, ko se na vhodu pojavi praštevilo. Izpisano funkcijo nato izpišemo v Veitch-ev diagram, od koder bomo lahko izvajali postopek minimizacije v PDNO, iz te oblike pa bomo prešli na PSNO (NAND) operatorji.

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



$$f_{MDNO} = \overline{x_1} \cdot \overline{x_2} \cdot x_3 + \overline{x_2} \cdot x_3 \cdot x_4 + \overline{x_1} \cdot x_3 \cdot x_4 + x_1 \cdot x_2 \cdot x_3 \cdot \overline{x_4} + \overline{x_1} \cdot x_2 \cdot \overline{x_3} \cdot \overline{x_4}$$

Da bi iz PDNO prešli na realizacijo s samimi NAND vrati, funkcijo v PDNO dvakrat negiramo in uporabimo de Morgan-ov teorem:

$$\overline{\overline{f_{MDNO}}} = \overline{\overline{x_1 \cdot \overline{x_2} \cdot x_3 + \overline{x_2} \cdot x_3 \cdot x_4 + \overline{x_1} \cdot x_3 \cdot x_4 + x_1 \cdot x_2 \cdot x_3 \cdot \overline{x_4} + \overline{x_1} \cdot x_2 \cdot \overline{x_3} \cdot \overline{x_4}}}$$

$$f_{PSNO} = \overline{(\overline{x_1 \cdot \overline{x_2} \cdot x_3}) \cdot (\overline{\overline{x_2} \cdot x_3 \cdot x_4}) \cdot (\overline{\overline{x_1} \cdot x_3 \cdot x_4}) \cdot (\overline{x_1 \cdot x_2 \cdot x_3 \cdot \overline{x_4}}) \cdot (\overline{\overline{x_1} \cdot x_2 \cdot \overline{x_3} \cdot \overline{x_4}})}$$

Za nastalo PSNO moramo določiti COST vezja (brez inverterjev):

OBLIKA	VRAT	VHODOV	COST
PSNO	6	22	28

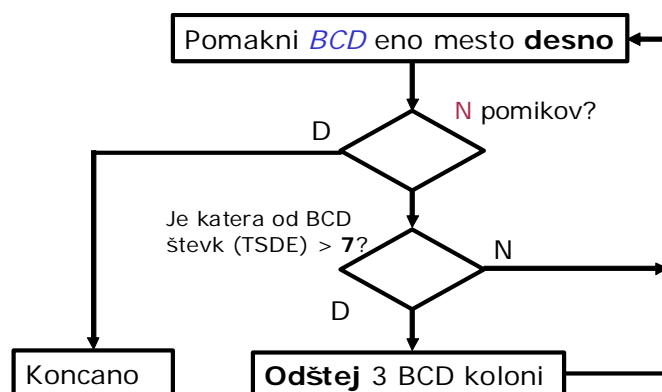
Čas pisanja je 60 minut. Vsaka naloga je vredna 10 točk.

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: Če vrednost 999_{10} zapišemo v šestnajstiški obliki dobimo $3E7_{16}$.

Postopek pretvorbe poteka po spodaj prikazanem obratnem "Double dabble" algoritmu za pretvorbo števila v BCD zapisu (TSDE) v N bitno število (bin). Začnemo tako, da zapišemo posamezne številke BCD števila: $1001\ 1001\ 1001_{BCD}$, jih razdelimo v 3 kolone BCD števk: Štotice, Desetice, Enice in njihove vrednosti po spodnjem algoritmu pomikamo v desno, ter jim odštevamo 3, če je njihova vrednost večja od 7. Po N pomikih je algoritem končan.



S	D	E	ŠTEVILO	operacija
1 0 0 1	1 0 0 1	1 0 0 1		POMIK1
1 0 0	1 1 0 0	1 1 0 0	1	-3
1 0 0	1 0 0 1	1 0 0 1	1	POMIK2
1 0	0 1 0 0	1 1 0 0	1 1	-3
1 0	0 1 0 0	1 0 0 1	1 1	POMIK3
1	0 0 1 0	0 1 0 0	1 1 1	POMIK4
	1 0 0 1	0 0 1 0	0 1 1 1	-3
	0 1 1 0	0 0 1 0	0 1 1 1	POMIK5
	0 1 1	0 0 0 1	0 0 1 1 1	POMIK6
	0 1	1 0 0 0	1 0 0 1 1 1	-3
	0 1	0 1 0 1	1 0 0 1 1 1	POMIK7
	0	1 0 1 0	1 1 0 0 1 1 1	-3
	0	0 1 1 1	1 1 0 0 1 1 1	POMIK8
		0 0 1 1	1 1 1 0 0 1 1 1	POMIK9
		0 0 1	1 1 1 1 0 0 1 1 1	POMIK10
		0 0	1 1 1 1 1 0 0 1 1 1	POMIK11
		0	0 1 1 1 1 1 0 0 1 1 1	POMIK12
				$3E7_{16}$

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 realiziramo avtomat končnih stanj Moore–ove izvedbe:

Najprej bomo razvili diagram prehajanja stanj, ki opisuje delovanje vezja za liho preverjanje paritete. Vezje je lahko v enem od dveh stanj: v sekvenci je bilo do tega trenutka liho ali sodo število enic. Kadar je na vhodu 1, je potrebno preklopiti v drugo stanje. Na primer, če je bilo do tega trenutka prisotnih liho število enic in je trenutni vhod 1, potem bomo imeli sedaj sodo število enic. Če pa bo na vhodu 0, ostane v istem stanju. Narisani diagram prehajanja stanj ima dve stanji, ki označujeta trenutno število enic na vhodu – torej LIHO in SODO. Izhod zapišemo pod stanjem (LIHO='1', SODO='0'). Vrednosti na vhodu x povzročajo spreminjanje stanj, ki so označene z usmerjenimi povezavami. Če je se na vhodu pojavi '0' (ne glede na to v katerem stanju smo) ostanemo v tem stanju. Jasno – saj štejemo samo '1'. Če smo v stanju LIHO in se na vhodu pojavi '1', preidemo v SODO. Če smo v stanju SODO in se na vhodu pojavi '1', preidemo v LIHO. Povedano povzema spodnji diagram prehajanja stanj

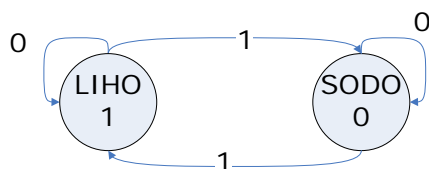


Diagram prehajanja stanj opišemo s tabelo prehajanja stanj:

vhod x	trenutno stanje $Q(t)$	naslednje stanje $Q(t+1)$
0	SODO	SODO
0	LIHO	LIHO
1	SODO	LIHO
1	LIHO	SODO

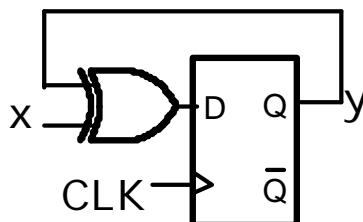
Iz tabele prehajanja stanj avtomata določimo enačbo za D–FF. Potrebno število FF je 1, saj sta stanji samo dve. Iz aplikacijske tabele sledi:

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

$$Q(t+1) = x \cdot \overline{Q(t)} + \overline{x} \cdot Q(t) = x \oplus Q(t)$$

$$y = Q(t)$$

Izvedba vezja je:



Če stanja kodiramo glede na njihov izhod LIHO '1' in SODO '0', potem dobimo novo aplikacijsko tabelo prehajanja stanj.

x	$Q(t)$	$Q(t+1)$	D	y
0	0	0	0	0
0	1	1	1	0
1	0	1	1	1
1	1	0	0	1

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:

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
F	E	D	0
G	F	G	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>

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	
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

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

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	
A	B	C	1
B	A	F	1
C	F	C	0
F	C	A	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>