

# RAZVOJ DIGITALNIH SISTEMOV

Izpit 26. 01. 2012

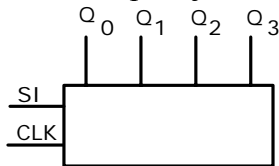
1. Pretvorite število  $3E7_{16}$  v BCD zapis z uporabo "double dabble" algoritma.

2. Uporabite PAL3L3 (namišljen čip) za realizacijo naslednjih funkcij:

- $f_1 = x_1 \oplus x_2$
- $f_2$  = konjunkcijo treh spremenljivk
- $f_3$  = funkcijo treh spremenljivk, ki vrne '1' pri vsaj dveh enicah na vseh vseh.

Vezje ima 3 vhode in 3 izhode. Vsaka disjunkcija (OR) ima 4 konjunkcije (AND). Oznaka L pomeni, da je izhod negiran. Programirane povezave označite s piko.

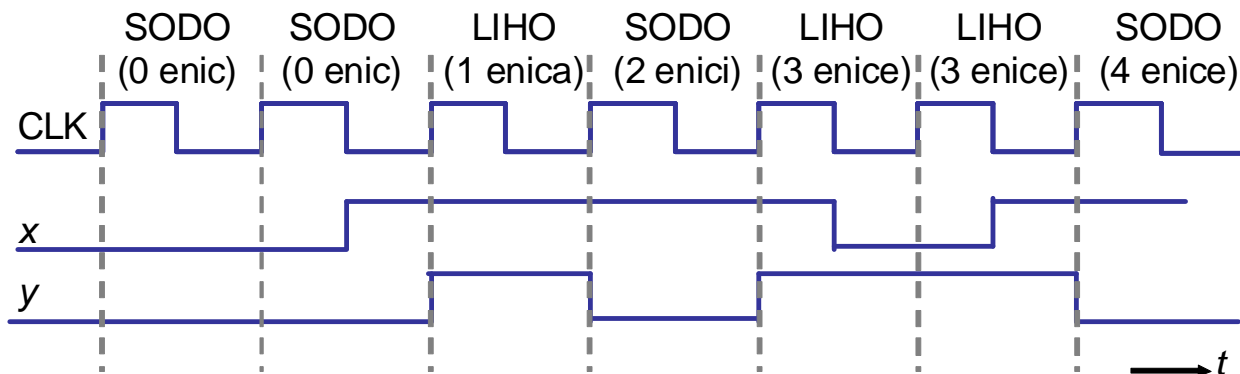
3. Sestavite 4-bitni pomikalni register s T-celicami in izbiralniki 2/1. Register ima zaporedni vhod *SI* (ang. serial input), in vzporedni izhod ( $Q_0$ ,  $Q_1$ ,  $Q_2$ ,  $Q_3$ ). Uporabite poimenovanje signalov na spodnji sliki.



4. 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:



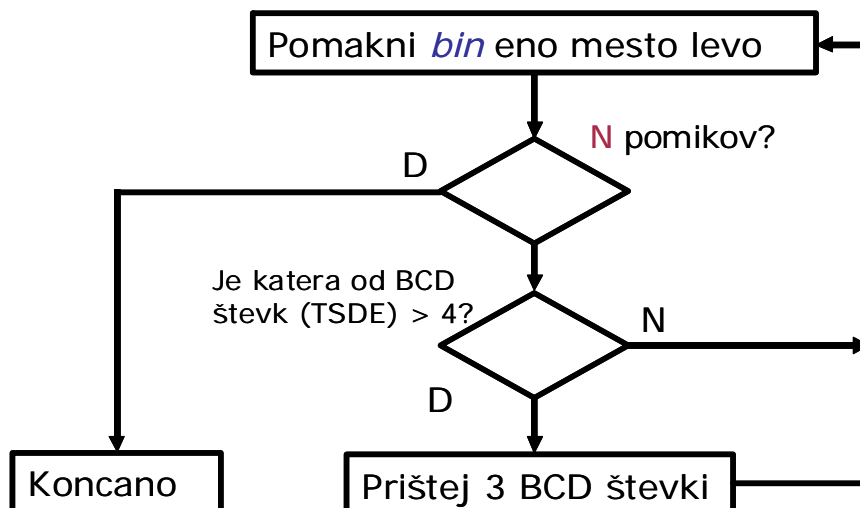
Čas pisanja je 60 minut. Vsaka naloga je vredna 10 točk.

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:  $3E7_{16} = 999_{10}$ . Zapis posameznih števk rezultata:  $1001\ 1001\ 1001_{BCD}$ .

Postopek pretvorbe poteka po spodaj prikazanem "Double dabble" algoritmu za pretvorbo  $N$  bitnega števila ( $bin$ ) v BCD zapis (TSDE)



STOTICE				DESETICE				ENICE				<i>1</i>	<i>1</i>	<i>1</i>	<i>1</i>	<i>1</i>	<i>0</i>	<i>0</i>	<i>1</i>	<i>1</i>	<i>1</i>	<i>1</i>	
											1	1	1	1	1	0	0	1	1	1			
										1	1	1	1	1	1	0	0	1	1	1			
									1	1	1	1	1	1	0	0	1	1	1				
								1	0	1	0	1	1	1	0	0	1	1	1				
							1	0	1	0	1	1	0	0	1	1	1						
							1	1	0	0	0	1	0	0	1	1	1						
						1	1	0	0	0	1	0	0	1	1	1							
						1	1	0	0	0	1	0	0	1	1	1							
					1	0	0	1	0	0	1	0	0	1	1	1							
			1	0	0	1	0	0	1	0	0	1	1	1									
		1	0	0	0	1	0	0	1	0	0	1	1										
		1	0	0	0	1	0	0	1	1	0	0	1	1									
	1	0	0	1	1	0	0	1	1	0	0	1	1										
	1	0	0	1	1	1	0	0	1	1	0	0	1	1									
1	0	0	1	1	0	0	1	1	0	0	1												
9 <sub>10</sub>				9 <sub>10</sub>				9 <sub>10</sub>															

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:

Za funkcije zapišemo najprej pravilnostno tabelo, nato narišemo Veitch–eve diagrame.

$x_1$	$x_2$	$x_3$	$f_1$	$f_2$	$f_3$
0	0	0	0	0	0
0	0	1	0	0	0
0	1	0	1	0	0
0	1	1	1	0	1
1	0	0	1	0	0
1	0	1	1	0	1
1	1	0	0	0	1
1	1	1	0	1	1

Vezje PAL ima negirane izhode, zato bomo pri realizaciji funkcij z Veitch–evimi diagrami realizirali  $\overline{f}$  in ne  $f$ .

$\overline{f_1}$ :

	$x_1$		
$x_2$	0	0	1
	1	1	0
	$x_3$		

Prvo funkcijo zapišemo enostavno, saj je negacija XOR funkcije dveh spremenljivk kar funkcija ekvivalence:

$$\overline{f_1} = x_1 \cdot x_2 + \overline{x_1} \cdot \overline{x_2}$$

Podobno lahko naredimo za drugo funkcijo, kjer za negacijo konjunkcije treh spremenljivk uporabimo De Morgan–ovo enakost.

$\overline{f_2}$ :

	$x_1$		
$x_2$	0	1	0
	0	0	0
	$x_3$		

$$\overline{f_2} = \overline{x_1 \cdot x_2 \cdot x_3}$$

$$\overline{f_2} = \overline{x_1} + \overline{x_2} + \overline{x_3}$$

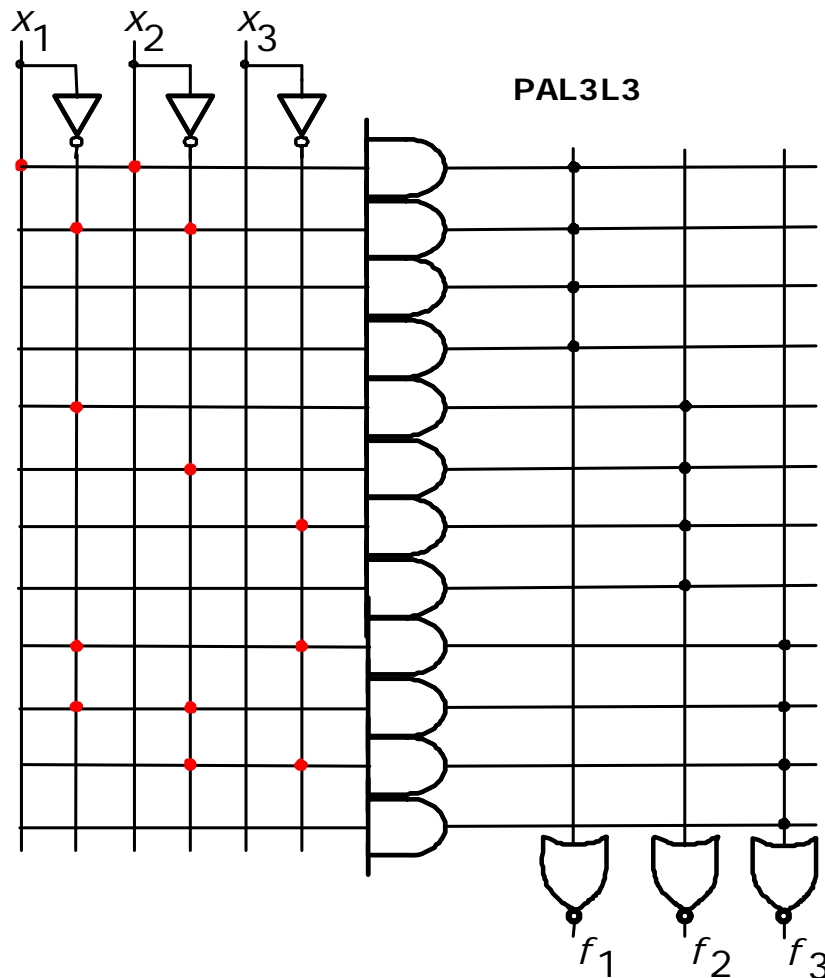
Zadnjo funkcijo minimiziramo z uporabo Veitch–evega diagrama, tako da zbiramo ničle.

$\overline{f_3}$ :

	$x_1$		
$x_2$	1	1	1
	0	1	0
	$x_3$		

$$\overline{f_3} = \overline{x_1 \cdot x_3} + \overline{x_1 \cdot x_2} + \overline{x_2 \cdot x_3}$$

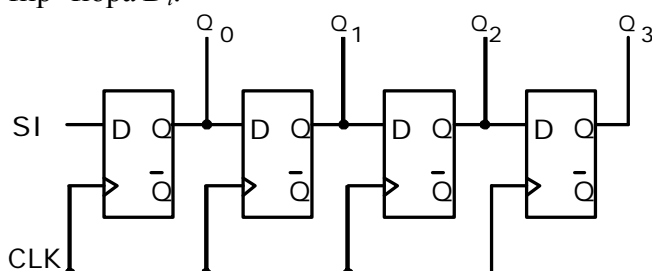
Pri realizaciji PAL vezja upoštevamo poenostavljeno strukturo, pri kateri ne vežemo vsake povezave na konjunkcije, saj so AND vrata na narisani strukturi 6-vhodna. Vezje PAL3L3 je AND–NOR arhitekture in vsebuje 4 konjunkcije na en NOR člen. Pri PAL vezju je programabilen samo AND del vezja.



Predstavljeno PAL3L3 vezje je sicer izmišljeno, vendar demonstrira strukturo in uporabo večjih (realnih) PAL vezij kot so npr. PAL14L4, GAL16V8 in GAL22V10. GAL vezja so nadgradnja osnovne PAL strukture. Slednji so izključno kombinacijski, GAL vezja pa imajo v OLMC (ang. Output Logic MacroCell) strukturi še D-FF, s katerim lahko realiziramo tudi sekvenčna vezja.

### Rešitev 3. 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  $D_i$ .



realiziramo.

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.

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

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

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

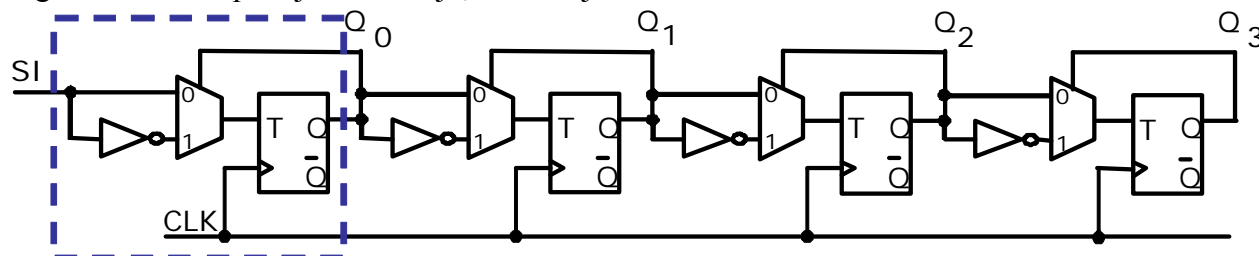
$D$	$Q(t)$	$Q(t+1)$	$T$
0	0	0	0
0	1	0	1
1	0	1	1
1	1	1	0

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

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.

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

#### Rešitev 4. 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 parnosti. 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

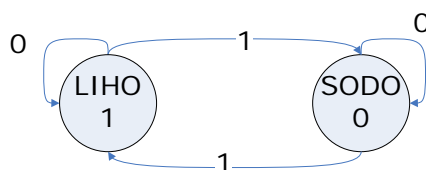


Diagram prehajanja stanj opišemo s tabelo prehajanja stanj:

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

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

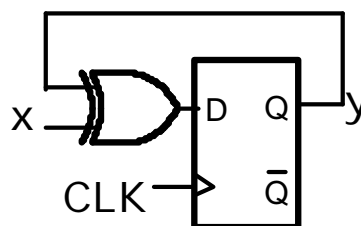
Iz tabele prehajanja stanj avtomata določimo enačbo za D-FF. Potrebno število FF je 1, saj sta stanji samo dve. Iz aplikacijske tabele sledi:

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

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

$$y = Q(t)$$

Izvedba vezja je:



Realizirali smo T-FF, prožen na sprednji rob signala ure.

Delovanje vezja si lahko ogledate v predlogah Logisim na domači strani predmeta:

Logisim\ff\T\_ff\_using\_D\_ff\_and\_xor.circ