

# RAZVOJ DIGITALNIH SISTEMOV

Izpit

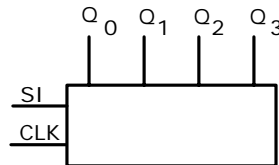
23. 06. 2014

1. Realizirajte funkcijo  $f$  s čim manj izbiralniki 4/1.

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

2. Pretvorite število  $59_{16}$  v BCD zapis z uporabo "double dabble" algoritma.

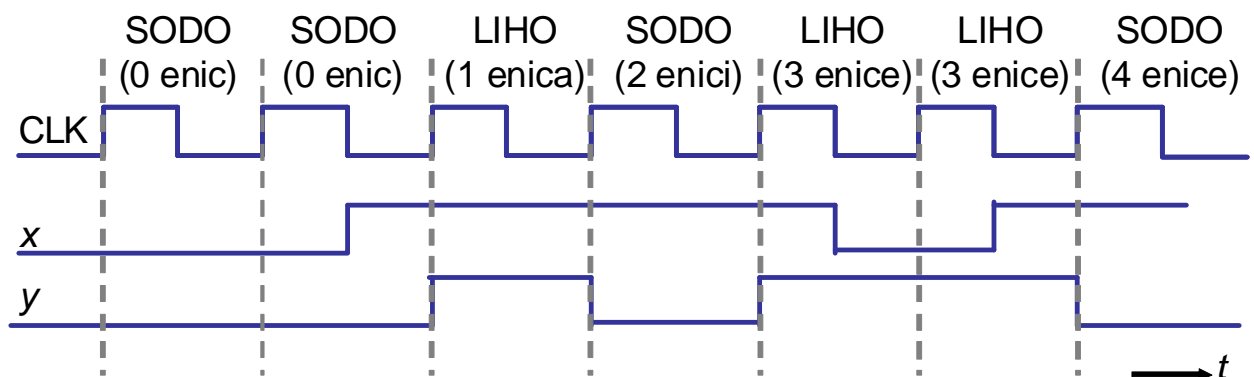
3. Sestavite 4-bitni pomikalni register s T-celicami in izbiralniki 2/1. Register ima zaporedni vhod  $SI$  (ang. serial input), in vzporedni izhod ( $Q_0, Q_1, Q_2, Q_3$ ). Uporabite poimenovanje signalov na spodnji sliki.



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:



Rešitev 1. naloge:

Funkcija  $f$  je podana v večnivojski (nenormalni) obliki:

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

zato jo najprej poenostavimo z uporabo pravil Boole-ove logike. Izpišemo desni člen funkcije in uporabimo lastnost Boole-ove logike  $x \cdot \bar{x} = 0$ , lastnost  $x \cdot x = x$  in lastnost  $0 + x = x$ .

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

Nad rezultatom ponovno uporabimo lastnost Boole-ove logike  $x \cdot \bar{x} = 0$  in lastnost  $x \cdot x = x$ .

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

Rezultat vnesemo v levi del funkcije in znova uporabimo omenjene lastnosti Boole-ove logike:

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

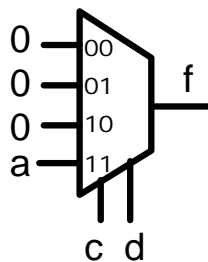
Dobimo dva člena in ju zapišemo v obliki PDNO, ki jo nato minimiziramo s pomočjo Veitch-evega diagrama ali z uporabo lastnosti združevanja Boole-ove algebre  $x + \bar{x} = 1$ :

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

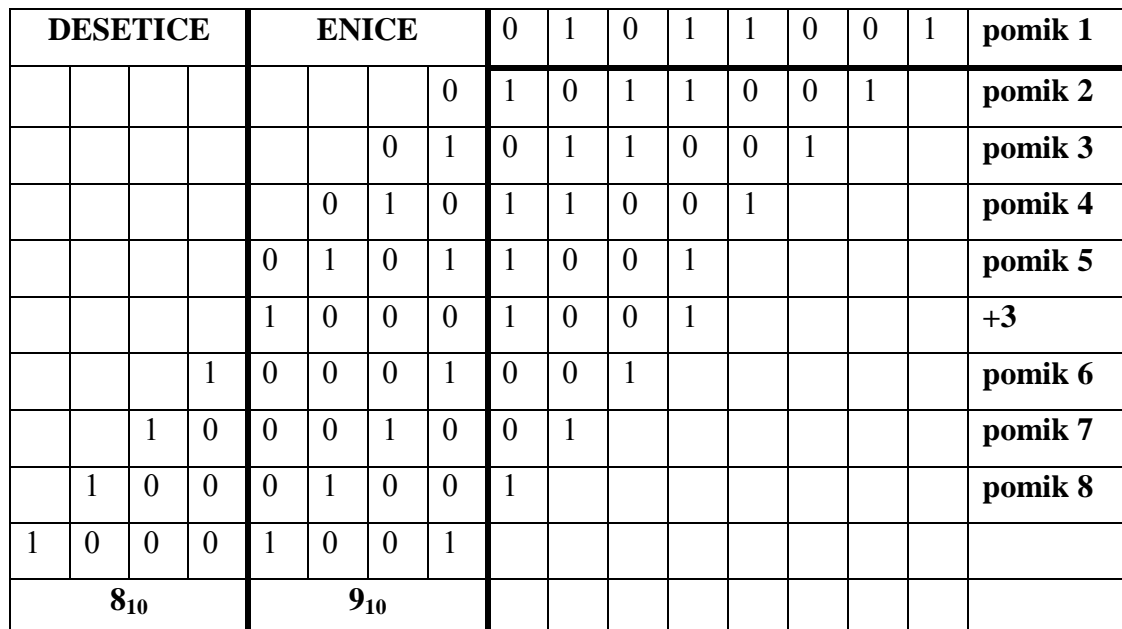
$$f_{PDNO}(a,b,c,d) = V(11,15)$$

$$f_{MDNO}(a,b,c,d) = a \cdot c \cdot d \cdot (\bar{b} + b) = a \cdot c \cdot d$$

in jo realiziramo z enim izbiralnikom 4/1, tako da naredimo Shannon-ov razvoj funkcije. Glede na kombinacijo naslovnih vhodov izbiralnika dobimo 6 možnih rešitev (ac, ca, ad, da, cd, dc).

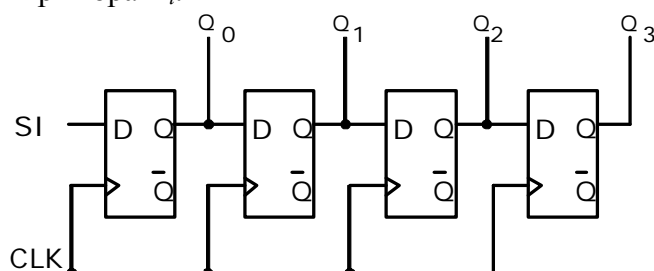


Pretvorimo šestnajstiko vrednost  $59_{16} = 89_{10}$ . Zapis posameznih števk: 0101 1001<sub>BCD</sub>. Pretvorbo po "double dabble" algoritmu opravimo po spodnjem algoritmu:



### 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:

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

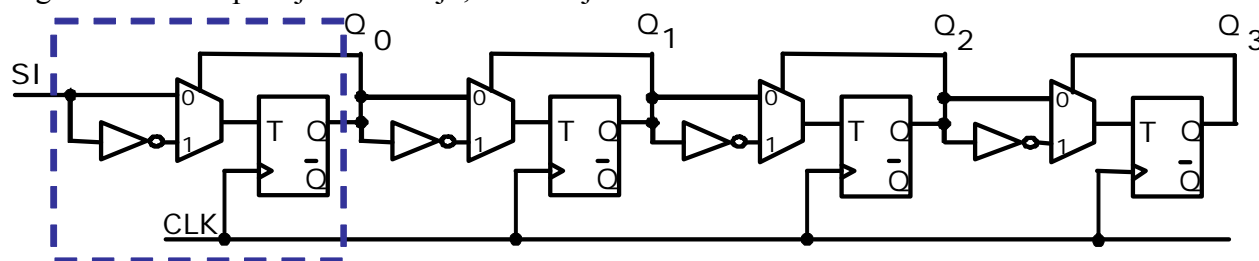
$D$	$Q(t)$	$Q(t+1)$	$T$
0	0	0	0
0	1	0	1
1	0	1	1
1	1	1	0

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.



### 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

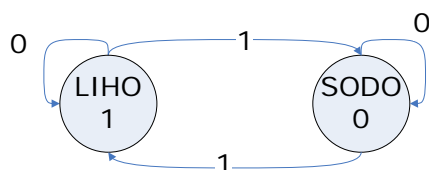


Diagram prehajanja stanj opišemo s tabelo prehajanja stanj:

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.

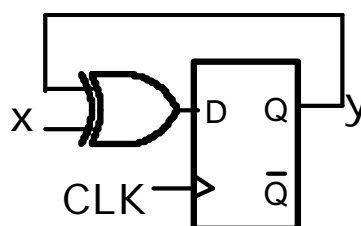
Iz aplikacijske tabele sledi:

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

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

$$y = Q(t)$$

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