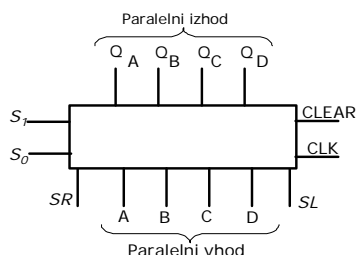


# RAZVOJ DIGITALNIH SISTEMOV

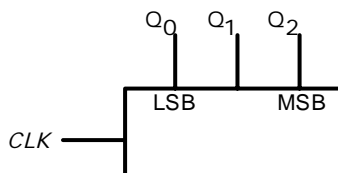
2. kolokvij  
22. 1. 2016

1. Realizirajte JK-flip flop z uporabo D- flip flopa in logičnih vrat.
2. Z uporabo D flip-flopov in izbiralnikov 4/1 prikažite sintezo univerzalnega 4-bitnega pomikalnega registra, ki ima dva funkcijska vhoda  $S_0$  in  $S_1$  in opravlja funkcije po spodnji tabeli. Register ima tudi zaporedna vhoda za pomik v levo (SL – serial left), pomik v desno (SR – serial right) in asinhroni vhod za brisanje CLEAR (aktiven nizek).



$S_1$	$S_0$	<i>funkcija</i>
0	0	drži stanje
0	1	pomik vsebine eno mesto desno
1	0	pomik vsebine eno mesto levo
1	1	nalaga vsebino z vhodov ABCD

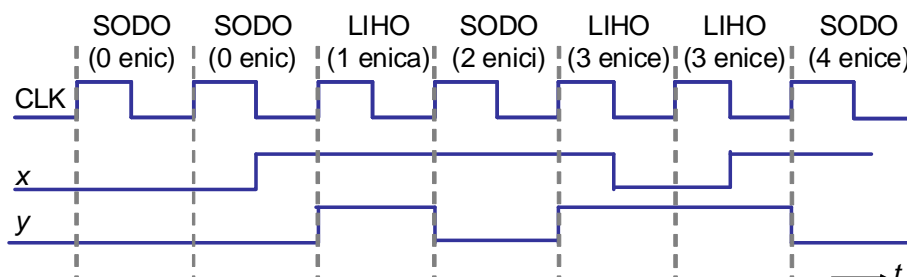
3. Prikažite sintezo sinhronnega 3-bitnega števca navzdol z uporabo T flip-flopov: Zapišite tabelo prehajanja stanj in določite enačbe flip-flopov ter vezje narišite. Imena signalov so razvidna iz spodnje slike.



4. Realizirajte generator lihe parnosti (paritete) kot Moore-ov 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.

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

## Rešitev 1. naloge

Za vezje JK-FF narišemo pravilnostno tabelo, pri čemer na vhodni strani zberemo vhode J, K in trenutno stanje  $Q(t)$ , na izhodni pa naslednje stanje  $Q(t+1)$ . JK-FF opravlja štiri funkcije (HOLD, RESET, SET, INVERT) glede na kombinacijo vhodnih signalov, medtem ko D-FF opravlja samo dve (SET, RESET).

$J$	$K$	$Q(t)$	$Q(t+1)$	$D$	funkcija $D$	funkcija $JK$
0	0	0	0	0	RESET	HOLD
0	0	1	1	1	SET	HOLD
0	1	0	0	0	RESET	RESET
0	1	1	0	0	RESET	RESET
1	0	0	1	1	SET	SET
1	0	1	1	1	SET	SET
1	1	0	1	1	SET	INVERT
1	1	1	0	0	RESET	INVERT

Iz tabele narišemo Veitchev diagram za vhod D v odvisnosti od vhodov J, K in trenutnega stanja  $Q(t)$ .

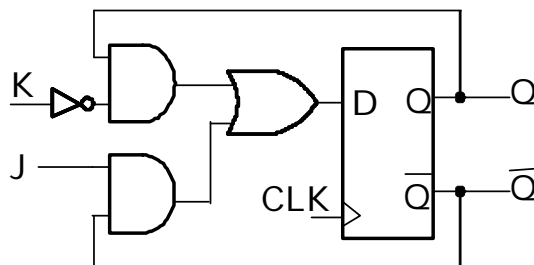
$D$ :

	$J$			
$K$	1	0	0	0
	1	1	1	0
	$Q(t)$			

Dobljeni diagram minimiziramo in zapišemo enačbo vhoda D:

$$D = J \cdot \overline{Q(t)} + \bar{K} \cdot Q(t)$$

Vezje narišemo:



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 2. naloge:

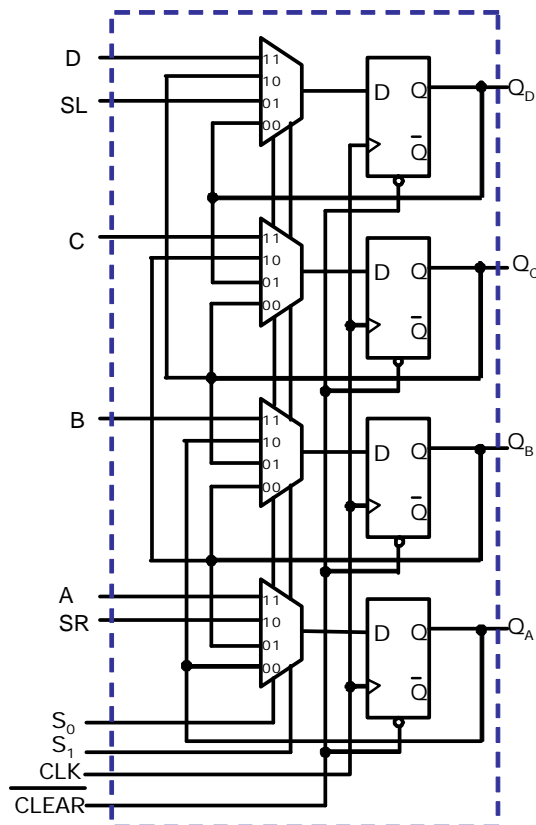
Vsako od operacij izpišemo v pravilnostno tabelo v kateri združimo funkcijska bita  $S_1$  in  $S_0$  in trenutno stanje na  $i$ -tem mestu registra  $Q_i(t)$ . Mesta registra od leve proti desni so  $Q_i = (Q_A, Q_B, Q_C, Q_D)$ . Realizacija z D flip-flopi nam analizo močno poenostavi, zaradi enačbe D flip-flopa:  $D = Q(t+1)$ .

Tabela 1: Prehajanje stanj univerzalnega registra.

$S_1$	$S_0$	$Q_i(t+1)$	funkcija
0	0	$Q_i(t)$	HOLD
0	1	$Q_{i+1}(t)$	LSR
1	0	$Q_{i-1}(t)$	LSL
1	1	$x_i$	LOAD

Iz poenostavljene tabele prehajanja stanj univerzalnega registra sestavimo realizacijo, ki bo vključevala izbiralnike MUX 4/1 in D-FF.

Na naslovna vhoda vseh MUX 4/1 vodimo funkcijska signala  $S_1$  in  $S_0$ . Potem na vsakem podatkovnem vhodu realiziramo ustrezno funkcijo.



Stanje  $S_1S_0 = "00"$  pomeni držanje stanja (*HOLD*), torej bodo trenutne vrednosti D-FF ohranile vrednost  $Q_i(t+1) = Q_i(t)$ . Na sliki to realiziramo tako, da vodimo izhod D-FF nazaj na vhod pri podatkovnem vhodu 00. Stanje  $S_1S_0 = "01"$  pomeni pomik desno (*LSR* – ang. logic shift right), torej bodo D-FF pomaknili vsebino eno mesto desno. Pomik desno pomeni, da na mesto skrajno levega bita vpišemo vrednost zaporednega vhoda SR, nato  $Q_A$  vodimo na vhod  $Q_B$  in tako do skrajno desnega bita. Stanje  $S_1S_0 = "10"$  pomeni pomik levo (*SHL* – ang. shift left), torej bodo D-FF pomaknili vsebino eno mesto levo. Pomik levo pomeni, da na mesto skrajno desnega bita vpišemo vrednost zaporednega vhoda SL, nato  $Q_D$  vodimo na vhod  $Q_C$  in tako do skrajno levega bita.  $S_1S_0 = "11"$  pomeni vzporedno nalaganje z vhodov (*LOAD*)  $Q_D(t+1) = D$ ,  $Q_C(t+1) = C$ ,  $Q_B(t+1) = B$ ,  $Q_A(t+1) = A$ . Na tabeli smo  $i$ -ti vhod za vzporedno nalaganje označili kot  $x_i = (A, B, C, D)$ .

### Rešitev 3. naloge:

Postopek sinteze zahteva, da zapišemo tabelo prehajanja stanj števec:

Trenutno stanje			Naslednje stanje			Enačbe 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>	T <sub>2</sub>	T <sub>1</sub>	T <sub>0</sub>
0	0	0	1	1	1	1	1	1
0	0	1	0	0	0	0	0	1
0	1	0	0	0	1	0	1	1
0	1	1	0	1	0	0	0	1
1	0	0	0	1	1	1	1	1
1	0	1	1	0	0	0	0	1
1	1	0	1	0	1	0	1	1
1	1	1	1	1	0	0	0	1

Normalna analiza bi zahtevala, da narišemo Veitch-eve diagrame za tri spremenljivke za vsak vhod T-FF, vendar ker so T-FF po svoji naravi primerni za realizacijo števec, so praviloma njihove vhodne enačbe zelo enostavne. Iz tabele prehajanja stanj števca določimo enačbe T-FF:

Iz stolpca T<sub>0</sub> se vidi:

$$T_0 = 1$$

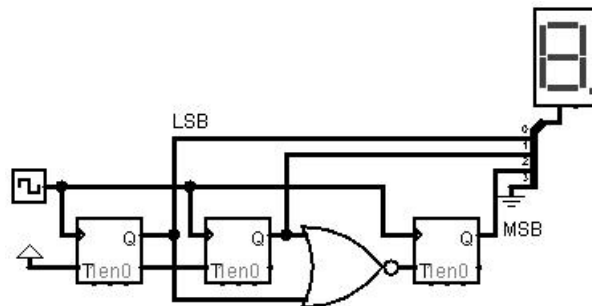
Z opazovanjem stolpcev trenutnega stanja določimo T<sub>1</sub>:

$$T_1 = \overline{Q_0}$$

Podobno lahko določimo T<sub>2</sub>:

$$T_2 = \overline{Q_0} \cdot \overline{Q_1} = \overline{Q_0 + Q_1}$$

Opis delovanja in vezje števca je v predlogah vaj na domači strani predmeta v imeniku Logisim\counter\ counter\_7\_0\_using\_T\_FF.circ:



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

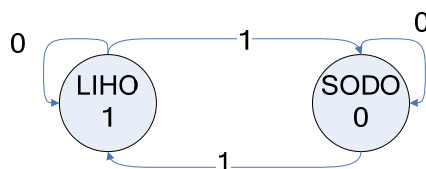


Diagram prehajanja stanj opišemo s tabelo prehajanja stanj:

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

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

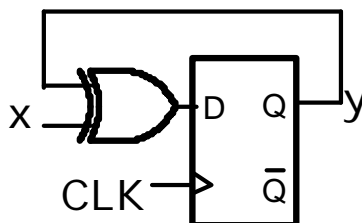
Iz aplikacijske tabele sledi:

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

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

$$y = Q(t)$$

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

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>