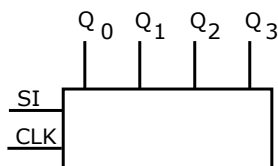


# RAZVOJ DIGITALNIH SISTEMOV

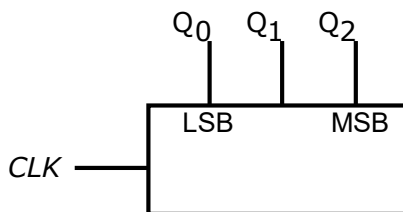
2. kolokvij

18.1.2024

1. Realizirajte JK-flip flop z uporabo RS-flip flopa in logičnih vrat.
2. Narišite vezje 4-bitnega pomikalnega registra z RS flip flopi in logičnimi vrati. Register ima zaporedni vhod  $SI$  (ang. serial input) in vzporedni izhod ( $Q_0$ ,  $Q_1$ ,  $Q_2$ ,  $Q_3$ ).



3. Prikažite sintezo 3-bitnega sinhronega števca, ki šteje po Grayevi kodi 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. Števno zaporedje po Grayevi kodi navzdol:  $4 \rightarrow 5 \rightarrow 7 \rightarrow 6 \rightarrow 2 \rightarrow 3 \rightarrow 1 \rightarrow 0 \rightarrow 4 \rightarrow 5 \dots$



4. Narišite diagram stanj za Mealyev avtomat končnih stanj, ki ima vhod  $w$  in izhod  $z$ . Avtomat končnih stanj postavi izhod  $z=1$ , ko se na vhodu pojavi zaporedje **110** ali **101**, sicer je  $z=0$ . Prekrivanje vzorcev je dovoljeno.

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 domači strani predmeta.

## 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 RS-FF opravlja samo dve (RESET, SET).

J	K	Q(t)	Q(t+1)	R	S	funkcija RS	funkcija JK
0	0	0	0	0	0	HOLD	HOLD
0	0	1	1	0	0	HOLD	HOLD
0	1	0	0	X	0	HOLD/RESET	RESET
0	1	1	0	1	0	RESET	RESET
1	0	0	1	0	1	SET	SET
1	0	1	1	0	X	HOLD/SET	SET
1	1	0	1	0	1	SET	INVERT
1	1	1	0	1	0	RESET	INVERT

Iz tabele narišemo Veitchev diagram za vhoda R in S v odvisnosti od vhodov J, K in trenutnega stanja Q(t).

*R:*

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

$$R = K \cdot Q(t)$$

***S:***

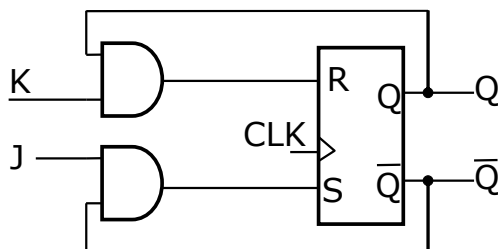
Diagram illustrating a matrix structure with rows labeled  $K$  and columns labeled  $Q(t)$ . The matrix is labeled  $J$  above it. The matrix contains the following values:

1	0	0	0
1	X	0	0

The first column is highlighted with a blue box.

$$S = J \cdot \overline{Q(t)}$$

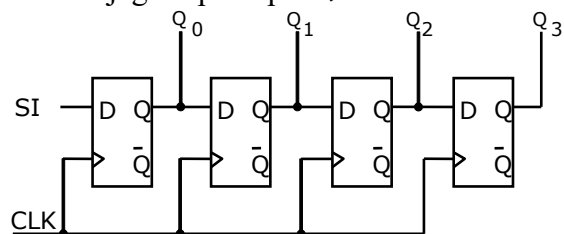
Vezje narišemo:



**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 domači strani predmeta.**

## Rešitev 2. 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  $D_i$ .



Če želimo pomikalni register sestaviti iz RS–FF in logičnih vrat, je en način realizacije realizirati celico D–FF s pomočjo RS–FF in logičnih vrat.

V ta namen zapišemo tabelo D–FF, pri kateri dodamo izhodni stolpec RS vhodov.

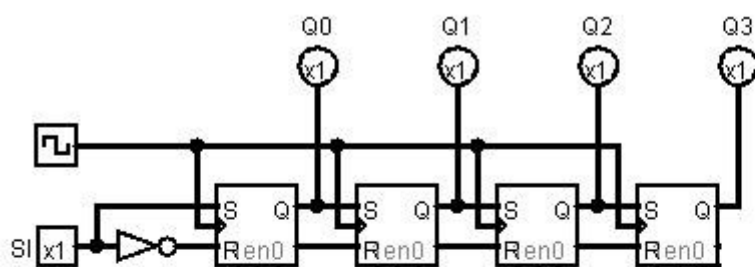
$D$	$Q(t)$	$Q(t+1)$	$S$	$R$	Pomen stanja
0	0	0	0	X	RESET/HOLD
0	1	0	0	1	RESET
1	0	1	1	0	SET
1	1	1	X	0	SET/HOLD

Iz tabele izrazimo R in S vhoda s pomočjo  $D$  vhoda in trenutnega stanja  $Q(t)$ . Če redundance (X) postavimo primerno, dobimo:

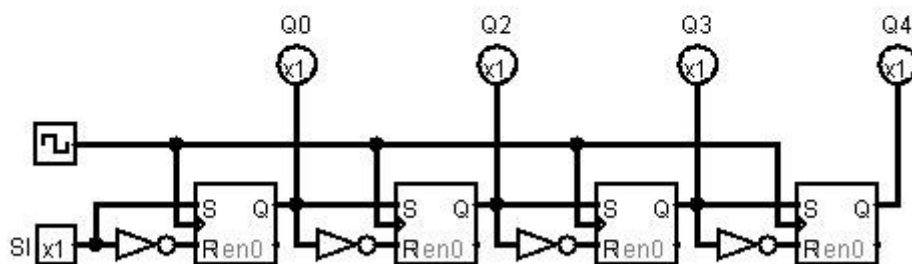
$$S = D \text{ in } R = \overline{D}$$

Če nastali D–FF sestavimo skupaj v 4–bitni pomikalni register dobimo spodnjo realizacijo.

Rešitev je opisana realizacija D–FF z RS–FF, ki je prikazana na spodnji sliki.



Neposredna izvedba pomikalnega registra z RS–FF



Izvedba pomikalnega registra s pomočjo D–FF, ki so realizirani z RS–FF

Razlika med zgornjo in spodnjo rešitvijo je v dodatnih inverterjih na mestih nad vhodom SI pri spodnji rešitvi, medtem ko so pri zgornji rešitvi uporabljeni negirani izhodi Q.

Rešitev 3. naloge:

Desetiška števna sekvenca po 3-bitni Grayevi kodi navzdol se glasi: ...  
4→5→7→6→2→3→1→0→4→5 ...

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	0	0	1	0	0
0	0	1	0	0	0	0	0	1
0	1	0	0	1	1	0	0	1
0	1	1	0	0	1	0	1	0
1	0	0	1	0	1	0	0	1
1	0	1	1	1	1	0	1	0
1	1	0	0	1	0	1	0	0
1	1	1	1	1	0	0	0	1

Normalna analiza bi zahtevala, da narišemo Veitcheve diagrame za tri spremenljivke za vsak vhod T–FF, vendar ker so T–FF po svoji naravi primerni za realizacijo števcov, so praviloma njihove vhodne enačbe zelo

enostavne. Iz tabele prehajanja stanj števec določimo enačbe T–FF:

Za vhod T<sub>0</sub> narišemo Veitchev diagram:



Funkcija na Veitchevem diagramu je linearna - če izračunamo koeficiente dobimo:

$$T_0 = Q_2 \oplus Q_1 \oplus Q_0$$

Če izpišemo vrednost T<sub>1</sub>, dobimo:

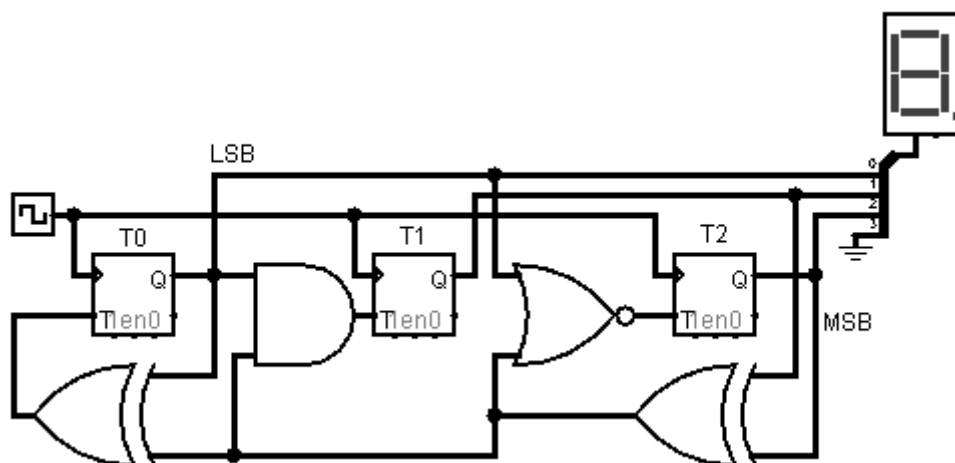
$$T_1 = \overline{Q_2} \cdot Q_1 \cdot Q_0 + Q_2 \cdot \overline{Q_1} \cdot Q_0 = (Q_2 \oplus Q_1) \cdot Q_0$$

Podobno lahko določimo T<sub>2</sub>:

$$T_2 = \overline{Q_2} \cdot \overline{Q_1} \cdot \overline{Q_0} + Q_2 \cdot Q_1 \cdot \overline{Q_0} = (\overline{Q_2 \oplus Q_1}) \cdot \overline{Q_0}$$

$$T_2 = \overline{(Q_2 \oplus Q_1)} + Q_0$$

Opis delovanja in vezje števec je v predlogah vaj na domači strani predmeta v imeniku Logisim\counter\counter\_down\_3\_bit\_Gray\_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 domači strani predmeta.

#### Rešitev 4. naloge:

Naloga zahteva realizacijo z Mealyevim tipom avtomata. Zapišemo začetno stanje A, v katerem ostajamo toliko časa, dokler se ne začne ena od sekvenc, ki ju zaznavamo. Obe sekvenci se začneta z '1', zato v stanje B preidemo, ko je na vhodu prva '1'. V stanju B ne moremo ostati, saj se na vhodu lahko pojavi '0' ali '1' – v obeh primerih gre za del zaznavanega zaporedja "10X" ali "11X". Iz stanja B preidemo v stanje C, če se vmes pojavi '1', tako da v tem stanju pomeni detekcijo sekvence "11X", v stanje D pa preidemo če se pojavi na vhodu '0', kar pomeni detekcijo sekvence "10X".

Prekrivanje zaporedij: Če se v stanju C pojavi '1' na vhodu, potem gre za sekvenco "111" na vhodu – kar še vedno pomeni, da ostajamo v stanju C, saj je prekrivanje vzorcev dovoljeno. Drugače se diagram obnaša, ko smo v stanju D in pride na vhod še ena '0' – takrat smo imeli na vhodu sekvenco "100", tako da se moramo vrniti v stanje A, saj se nobena od zaznavanih zaporedij ne začneja z '0'.

