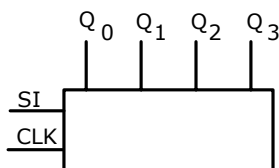


RAZVOJ DIGITALNIH SISTEMOV

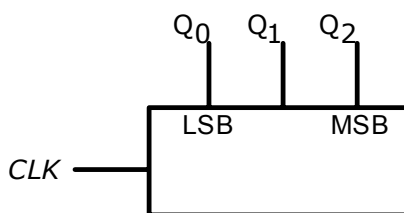
2. kolokvij

27. 1. 2017

1. Realizirajte JK-flip flop z uporabo T- flip flopa in logičnih vrat.
2. Narišite vezje 4-bitnega pomikalnega registra z JK celicami 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 sinhronnega 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. Načrtajte diagram stanj Moore-ovega avtomata končnih stanj, ki krmili delovanje garažnih vrat: Garažna vrata imajo vhod VRATA ter vhod ZAŠČITA, ki postane '1' vedno, ko preko motorja steče dovolj velik tok. Z meritvijo toka na motorju obenem izdelamo funkcijo detekcije obeh končnih položajev, kot tudi zaščito proti oviram na poti vrat. Vezje ima 2-bitni izhod za enosmerni motor:

Koda operacije		Funkcija izhoda
OP ₁	OP ₀	
0	0	motor stoji
0	1	motor pomika vrata navzgor
1	0	motor pomika vrata navzdol

Če pritisnemo gumb VRATA, se vrata začno pomikati navzgor. Če na poti naletijo na oviro ali pridejo do zgornje končne lege, se motor ustavi. Če pritisnemo gumb VRATA ponovno, se začnejo gibati v smeri navzdol. Podobno je v obratni smeri: Če na poti naletijo na oviro ali pridejo do spodnje končne lege, se motor ustavi. Če pritisnemo gumb VRATA ponovno, se začnejo pomikati v smeri navzgor.

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.

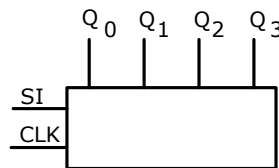
Rezultati bodo objavljeni domači strani predmeta.

DEVELOPMENT OF DIGITAL SYSTEMS

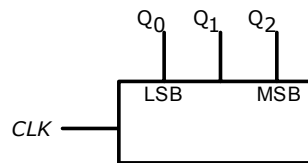
Midterm examination

27. 01. 2017

1. Implement a JK-flip flop using a T- flip flop and logic gates.
2. Draw the circuit diagram of a four bit shift SIPO register, using JK type flip-flops and any logical gates. The SIPO register has a serial input SI (ang. serial input) and a parallel output (Q_0, Q_1, Q_2, Q_3). Name the signals according to figure below.



3. Synthesize a 3 bit Gray down counter using T type flip-flops: Draw the state transition table, determine T flip-flop input equations and draw the resulting counter circuit using signal names in the figure below.
Note: 3-bit Gray down counting sequence: $4 \rightarrow 5 \rightarrow 7 \rightarrow 6 \rightarrow 2 \rightarrow 3 \rightarrow 1 \rightarrow 0 \rightarrow 4 \rightarrow 5 \dots$



4. Draw the state transition diagram of a Moore type finite state machine, which controls the movement of a garage door. The control circuit has an input (DOOR) and an input (PROTECTION), which is set to '1' whenever a current limit of the motor is reached. The motor current limit input is used to detect both door end positions as well as for protection against obstacles in the door path. The control circuit has a two bit motor output:

Operation code		Motor operation
OP ₁	OP ₀	
0	0	motor stop
0	1	door moving upwards
1	0	door moving downwards

When the door knob is pressed ($DOOR=1$), the door starts to move upwards. If an obstacle is in the door path or the door reaches its upper end position ($PROTECTION=1$), the motor stops. When the door knob is pressed again, the door starts to move downwards until the protection limit is reached again. After the door knob is pressed again, the process is repeated.

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 T-FF opravlja samo dve (HOLD, INVERT).

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

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

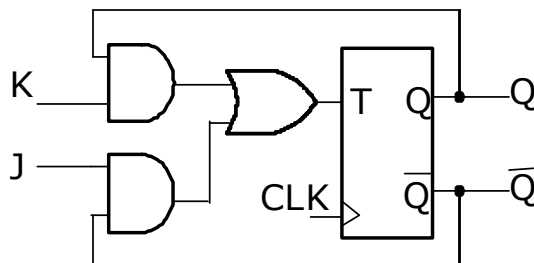
$T:$

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

Dobljeni diagram minimiziramo in zapišemo enačbo vhoda T:

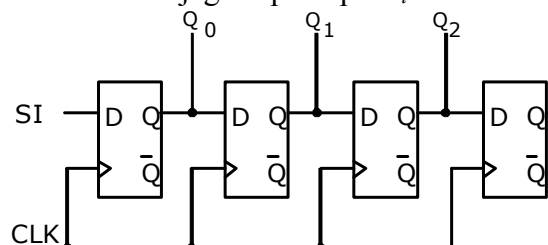
$$T = J \cdot \overline{Q(t)} + K \cdot Q(t)$$

Vezje narišemo:



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 JK-FF in logičnih vrat, je en način realizacije realizirati celico D-FF s pomočjo JK-FF in logičnih vrat.

V ta namen zapišemo tabelo D-FF, pri kateri dodamo izhodni stolpec JK

vhodov.

D	$Q(t)$	$Q(t+1)$	J	K	Pomen stanja
0	0	0	0	X	RESET/HOLD
0	1	0	X	1	INVERT/RESET
1	0	1	1	X	INVERT/SET
1	1	1	X	0	SET/HOLD

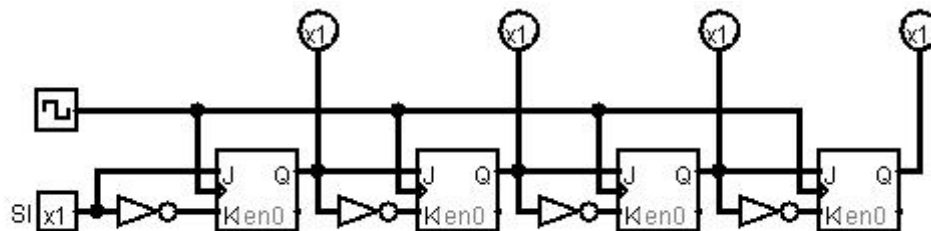
Iz tabele izrazimo J in K vhoda s pomočjo D vhoda in trenutnega stanja $Q(t)$. Če redundance (X) postavimo primerno, dobimo:

$$J = D$$

$$K = \bar{D}$$

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

Ena možna rešitev je opisana realizacija D-FF z JK-FF, ki je prikazana na spodnji sliki.

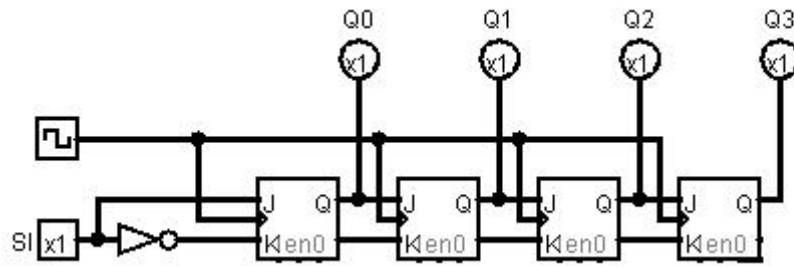


Drugo možnost predstavlja neposredna realizacija s pomočjo tabele pomikanja med opazovanim mestom Q_i in naslednjim mestom Q_{i+1} pomikalnega registra. Ponovno zapišemo vzbujalno tabelo za opazovano mesto in določimo vhode glede na spremembo stanja ($Q_{i+1}(t) \rightarrow Q_{i+1}(t+1)$) na $(i+1)$ mestu.

Trenutna vsebina registra		Vsebinska registra ob pomiku	Vhoda FF na mestu $i+1$		Pomen stanja
$Q_i(t)$	$Q_{i+1}(t)$	$Q_{i+1}(t+1)$	J_{i+1}	K_{i+1}	
0	0	0	0	X	RESET/HOLD
0	1	0	X	1	INVERT/RESET
1	0	1	1	X	INVERT/SET
1	1	1	X	0	SET/HOLD

Če v dobljenih vseh vhodi za mesto $(i+1)$ izberemo primerne redundance, dobimo vhoda $J_{i+1} = Q_i(t)$ in $K_{i+1} = Q_i(t)'$. To velja tudi za vhodno mesto, torej bomo na vhodni JK-FF vezali vhod $J_0 = x(t)$ in $K_0 = x(t)'$.

Neposredna realizacija pomikalnega registra z JK–FF je prikazana na spodnji sliki.



Vezje se nahaja v Logisim predlogah rešenih nalog na domači strani predmeta:
Logisim\shift_reg\shift_reg_4bit_using_jkff.circ

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 števca:

Trenutno stanje			Naslednje stanje			Enačbe FF		
Q ₂	Q ₁	Q ₀	Q ₂	Q ₁	Q ₀	T ₂	T ₁	T ₀
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 Veitch-eve 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 števca določimo enačbe T-FF:

Za vhod T₀ narišemo Veitchev diagram:

T_0 :

	Q_2			
Q_1	0	1	0	1
	1	0	1	0
	Q_0			

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

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

Če izpišemo vrednost T₁, dobimo:

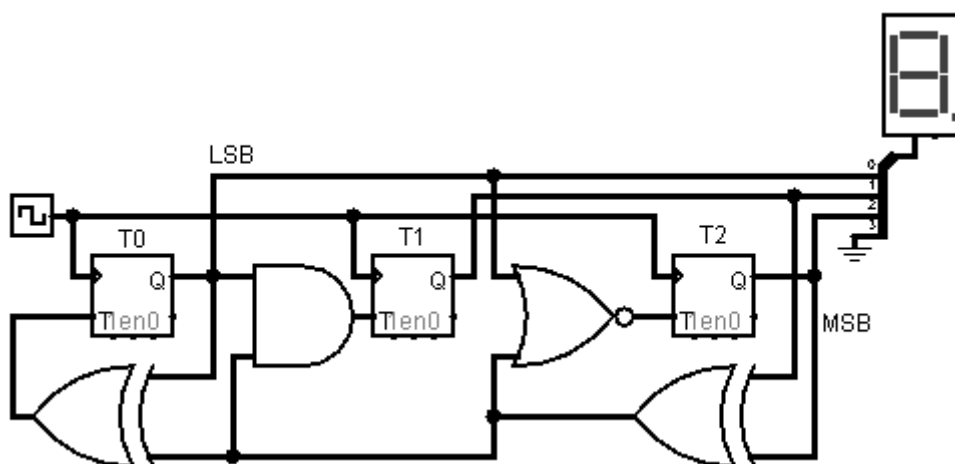
$$T_1 = \overline{Q_2} \cdot Q_1 \cdot Q_0 + Q_2 \cdot \overline{Q_1} \cdot Q_0 = (Q_2 \oplus Q_1) \cdot Q_0$$

Podobno lahko določimo T₂:

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

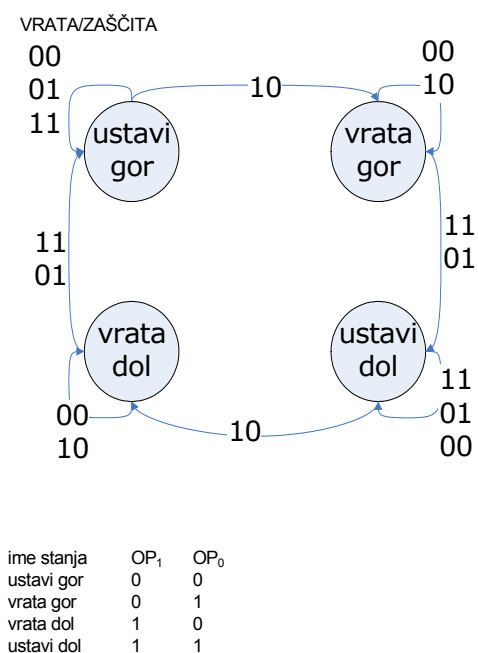
$$T_2 = (\overline{Q_2 \oplus Q_1}) + Q_0$$

Opis delovanja in vezje števca je v predlogah vaj na domači strani predmeta v imeniku Logisim\counter\counter_down_3_bit_Gray_using_T_FF.circ:



Rešitev 4. naloge:

Narišemo Moore-ov diagram stanj:



Iz opisa naloge je razvidno, da stanje "ustavi" ni samo eno, ker si moramo zapomniti v katero smer so se gibala vrata, da bi lahko šli v nasprotni smeri. Glede na to imamo stanji "ustavi gor", ki določa, da se bodo vrata ob naslednjem pritisku na gumb gibala gor in stanje "ustavi dol", ki določa, da se bodo vrata ob naslednjem pritisku na gumb gibala dol. Če stanja ločimo tako, potem v stanju "ustavi gor" ostajamo toliko časa, dokler ne pritisnemo VRATA in jasno na motorju ni napake, se pravi kombinacija "10". Vrata se nato pomikajo gor (preidemo v stanje "vrata gor"). V tem stanju lahko tipko spustimo in vrata se pomikajo navzgor. To se dogaja toliko časa, dokler ne naletimo na pogoj ZAŠČITA='1' (se pravi kombinaciji

"11" in "01". Ko postane pogoj ZAŠČITA='1' se postavimo v stanje "ustavi dol" in v tem stanju ostajamo dokler vztraja pogoj ZAŠČITA='1' oz. dokler ne pritisnemo tipke VRATA='1' (kombinacija "10"). Takrat na podoben način preidemo v stanje "vrata dol", kjer ostanemo dokler ne naletimo na oviro (tla prostora recimo), ko preidemo v stanje "ustavi gor".

Takšna realizacija še zdaleč ni optimalna: Bolje bi bilo, če bi avtomat realizirali kot Mealy-ev tip. Dejanska realizacija ne vsebuje avtomata, temveč en T-FF in relejno logiko.