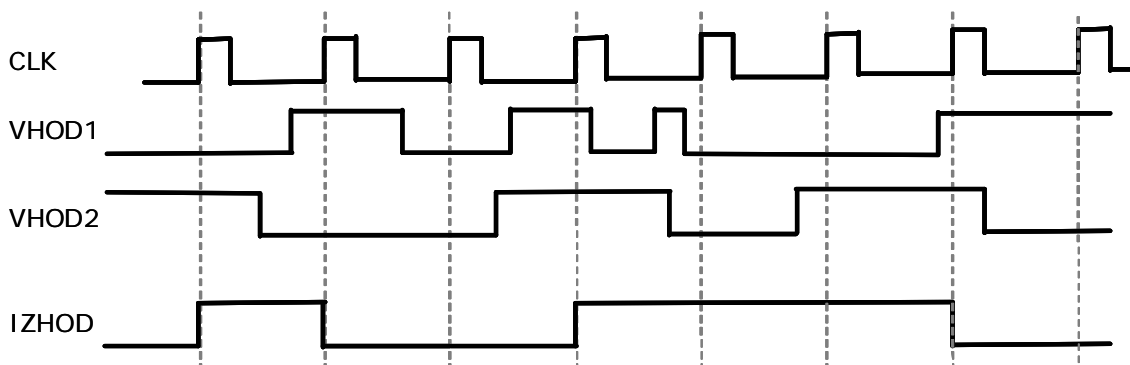


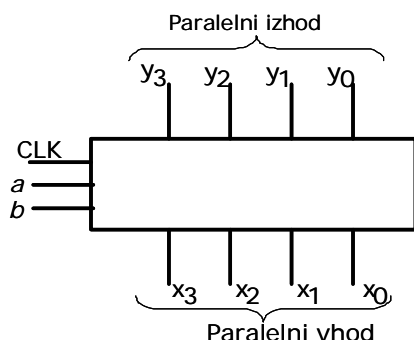
RAZVOJ DIGITALNIH SISTEMOV

2. kolokvij 19. 1. 2011

1. Prikazano je časovno zaporedje signalov, ki popolnoma podaja delovanje nekega sekvenčnega vezja. Narišite sekvenčno vezje, ki realizira spodnje zaporedje signalov.

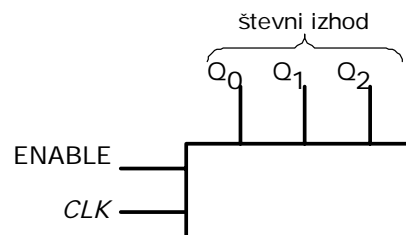


2. S pomočjo 4/1 izbiralnikov, inverterjev in D flip-flopov realizirajte univerzalni register, ki ima dva funkcijska vhoda a in b in opravlja funkcije po spodnji tabeli:



a	b	funkcija
0	0	briše vsebino registra (CLEAR)
0	1	rotira vsebino eno mesto desno (ROR)
1	0	negira vsebino (CPL)
1	1	vpiše vsebino s paralelnega vhoda (LOAD)

3. Prikažite sintezo 3-bitnega sinhronega števca navzgor z omogočanjem štetja (ENABLE) s T flip-flopi in logičnimi vrati. Če je ENABLE='1', števec šteje, če je ENABLE='0' števec ohranja stanje.



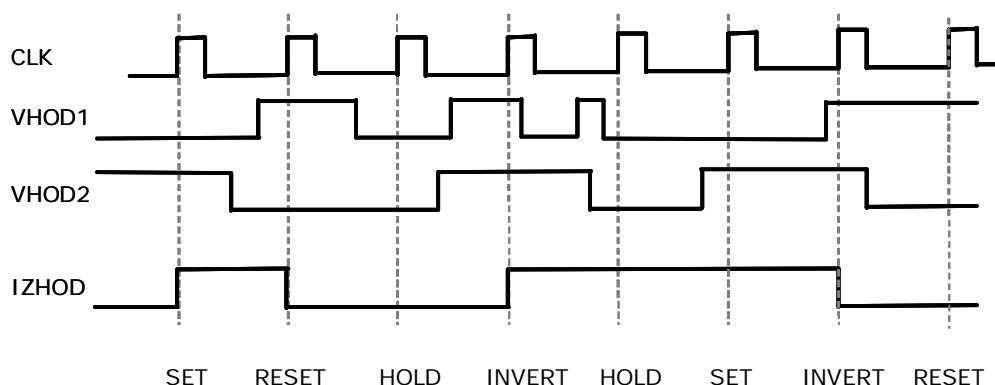
4. Narišite diagram prehajanja stanj avtomata končnih stanj, ki ima vhod w in izhod z . Avtomat končnih stanj postavi izhod $z=1$, ko se na vhodu pojavi zaporedje **101**, sicer je $z=0$. Prekrivanje vzorcev je dovoljeno. Delovanje avtomata končnih stanj povzema spodnje časovno zaporedje vhoda in izhoda.

CLK	t_0	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}	t_{11}	t_{12}	t_{13}	t_{14}	t_{15}
w	0	0	1	0	0	1	0	1	0	1	1	1	0	1	1	1
z	-	0	0	0	0	0	0	1	0	1	0	0	0	1	0	0

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 <http://estudent.fri.uni-lj.si/fe.html>

Rešitev 1. naloge

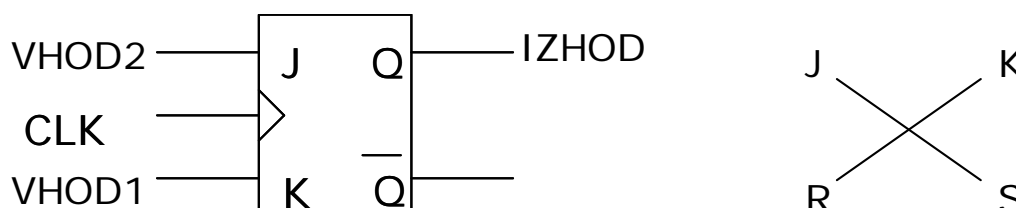
Podano je časovno zaporedje signalov. S črtkano črto je označen rob spremembe, saj gre očitno za robno proženo vezje (pozitivni rob).. Vezje je sekvenčno, torej je delovanje lahko odvisno od vhodov in prejšnjega stanja vezja. Časovni diagram mora vsebovati toliko ciklov signala ure, da je delovanje *popolnoma podano*. V našem primeru je to 8 ciklov, torej lahko sklepamo, da vezje nima več kot eno spominsko stanje, saj dobimo 8 kombinacij s 3 vhodnimi spremenljivkami.



Za vezje narišemo pravilnostno tabelo, v kateri združimo oba vhoda in prejšnje stanje, saj naloga pravi, da gre za sekvenčno vezje.

<i>VHOD1</i>	<i>VHOD2</i>	<i>IZHOD(t)</i>	<i>IZHOD(t+1)</i>	<i>funkcija vezja</i>	<i>JK-FF</i>	
0	0	0	0	HOLD	J=0	K=0
0	0	1	1	HOLD	J=0	K=0
0	1	0	1	SET	J=1	K=0
0	1	1	1	SET	J=1	K=0
1	0	0	0	RESET	J=0	K=1
1	0	1	0	RESET	J=0	K=1
1	1	0	1	INVERT	J=1	K=1
1	1	1	0	INVERT	J=1	K=1

Iz tabele lahko povzamemo, da gre za JK flip-flop, prožen na pozitivni rob signala. VHOD2 ima funkcijo J vhoda (opravlja funkcijo postavljanja – set), medtem ko ima VHOD1 funkcijo K vhoda (opravlja funkcijo brisanja – reset). IZHOD je Q.



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 <http://estudent.fri.uni-lj.si/fe.html>

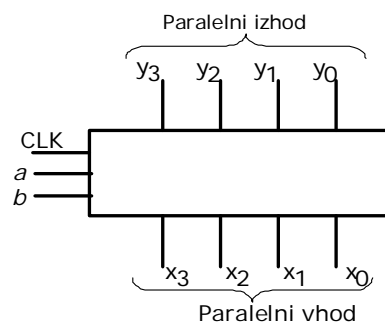
Rešitev 2. naloge:

Naloga zahteva realizacijo univerzalnega registra s funkcijami

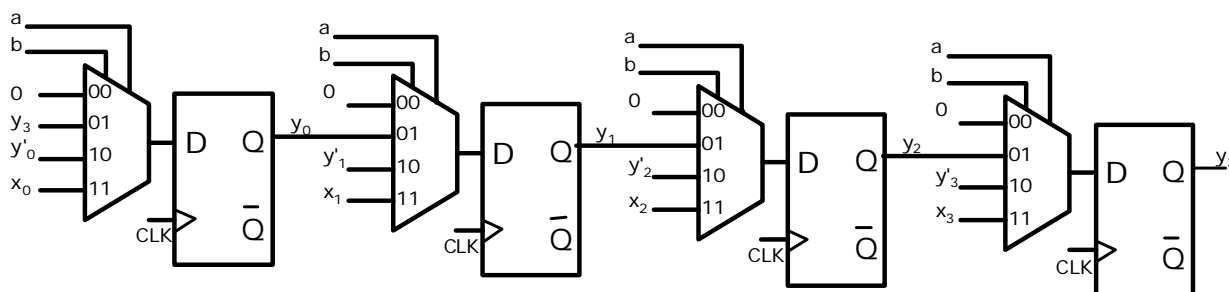
a	b	<i>funkcija</i>
0	0	briše vsebino registra (CLEAR)
0	1	rotira vsebino eno mesto desno (ROR)
1	0	negira vsebino (CPL)
1	1	vpiše vsebino s paralelnega vhoda (LOAD)

Vsako od operacij izpišemo v pravilnostno tabelo v kateri združimo funkcijska bita a , b in trenutno stanje na i -tem mestu registra $y_i(t)$. Realizacija z D flip-flopi nam analizo močno poenostavi, zaradi enačbe D flip-flopa: $D = y_{i+1}(t)$

a	b	$y_i(t)$	$y_i(t+1)$	D
0	0	0	0	0
0	0	1	0	0
0	1	0	$y_{i-1}(t)$	$y_{i-1}(t)$
0	1	1	$y_{i-1}(t)$	$y_{i-1}(t)$
1	0	0	y_i'	y_i'
1	0	1	y_i'	y_i'
1	1	0	x_i	x_i
1	1	1	x_i	x_i



Register izvaja rotacijo, torej nima serijskega vhoda in izhoda, ampak LSB bit y_3 vodimo na MSB bit y_0 . Naloga zahteva realizacijo z 4/1 izbiralniki, s katerimi ločimo 4 operacije registra.



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 <http://estudent.fri.uni-lj.si/fe.html>

Rešitev 3. naloge:

Postopek sinteze zahteva, da zapišemo tabelo prehajanja stanj števca:

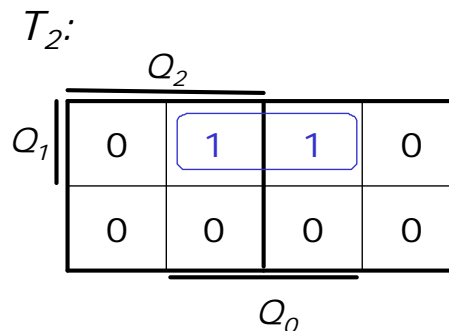
trenutno stanje			naslednje stanje			T-FF		
Q ₂	Q ₁	Q ₀	Q ₂	Q ₁	Q ₀	T ₂	T ₁	T ₀
0	0	0	0	0	1	0	0	1
0	0	1	0	1	0	0	1	1
0	1	0	0	1	1	0	0	1
0	1	1	1	0	0	1	1	1
1	0	0	1	0	1	0	0	1
1	0	1	1	1	0	0	1	1
1	1	0	1	1	1	0	0	1
1	1	1	0	0	0	1	1	1

Iz tabele prehajanja stanj števca določimo enačbe T-FF:

Za T₀ se iz tabele vidi T₀ = 1

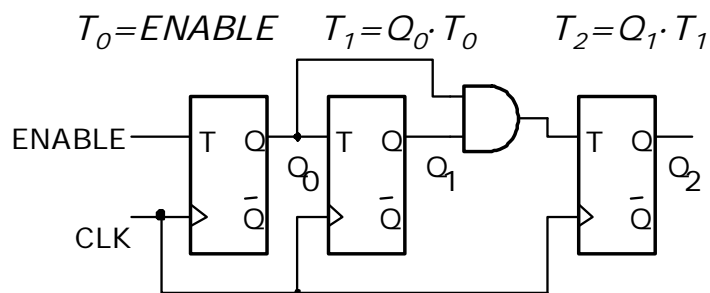
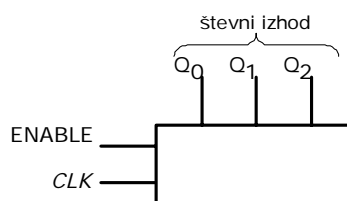
Za T₁ se iz tabele vidi T₁ = Q₀·1 = Q₀·T₀

Za T₂ narišemo Veitchev diagram



$$T_2 = Q_0 \cdot Q_1 = T_1 \cdot Q_1$$

Naloga pravi, da moramo dodati še vhod za omogočanje štetja (ENABLE). Če prvemu T-FF postavimo T₀ = '0' namesto T₀ = '1', flip-flopi števca ne bodo štel, ampak bodo ohranjali stanje. Torej, če na vhod T₀ postavimo zunanji signal ENABLE, števec ne bo štel, ampak ohranjal stanje, če bo ENABLE='0'. V verigi sinhronnega števca so namreč vezani vsi T-FF tako, da so odvisni od prvega T-FF. Do istega sklepa bi prišli, če bi risali Veitch-eve diagrame za 4 spremenljivke (ENABLE, Q₂, Q₁, Q₀).



Rešitev 4. naloge:

Iz tabele, ki ponazarja delovanje avtomata končnih stanj se vidi, da to ne more biti Moore–ov tip avtomata ampak Mealy–ev tip, saj se izhod spremeni *takoj*, ko na vhod pride druga '1' sekvence "101", ki jo zaznavamo.

CLK	t ₀	t ₁	t ₂	t ₃	t ₄	t ₅	t ₆	t ₇	t ₈	t ₉	t ₁₀	t ₁₁	t ₁₂	t ₁₃	t ₁₄	t ₁₅
w	0	0	1	0	0	1	0	1	0	1	1	1	0	1	1	1
z	-	0	0	0	0	0	0	1	0	1	0	0	0	1	0	0

Avtomat začnemo risati v nekem začetnem stanju. Za vsako stanje imamo dve možnosti, saj je vhod samo eden. Vezje ostaja v stanju A toliko časa, dokler ne pride na vhod prva '1' zaporedja. Ko se to zgodi, preide avtomat v stanje B, kjer ostane dokler je na vhodu '1' (glej tabelo – trenutek t₉, t₁₀, t₁₁). Ko se na vhodu pojavi '0' preide v zadnje stanje, od koder sta dve možnosti: Če se naslednji cikel pojavi '0' moramo do sedaj zaznano zaporedje "10X" zavreči (glej tabelo – trenutek t₂, t₃, t₄), tako da se vrnemo v stanje A, če pa se na vhodu pojavi '1', pa postavimo izhod na '1' in se vrnemo na začetek:

Takšna realizacija ne dovoljuje prekrivanja vzorcev – če dovolimo tudi prekrivanje vzorcev (glej tabelo – trenutek t₇, t₈, t₉), potem mora avtomat preiti iz stanja C ob pogoju w='1' v stanje B in ne v stanje A. Na spodnji sliki sta prikazani obe rešitvi za detektor zaporedja "101": Zgornja slika predstavlja rešitev brez prekrivanja zaporedij, spodnja s prekrivanjem zaporedij.

