

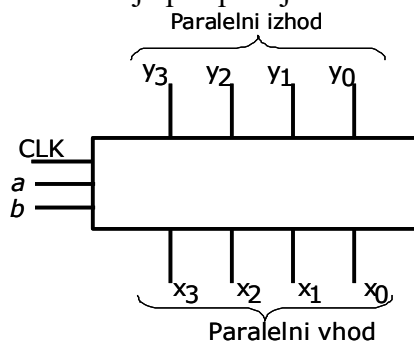
RAZVOJ DIGITALNIH SISTEMOV

2. kolokvij

12.1. 2022

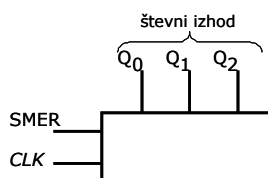
1. Realizirajte JK-flip flop z uporabo T flip flopa in logičnih vrat.

2. Z uporabo univerzalnih logičnih modulov prikažite sintezo univerzalnega 4-bitnega pomikalnega registra, ki ima dva funkcijska vhoda a in b in opravlja funkcije po spodnji tabeli.



a	b	<i>funkcija</i>
0	0	briše vsebino ($y_i = 0$)
0	1	rotira vsebino eno mesto levo ($y_0 \rightarrow y_1 \rightarrow y_2 \rightarrow y_3 \rightarrow y_0$)
1	0	rotira vsebino eno mesto desno ($y_0 \rightarrow y_3 \rightarrow y_2 \rightarrow y_1 \rightarrow y_0$)
1	1	naloži vsebino s paralelnega vhoda ($y_i = x_i$)

3. Prikažite sintezo dvosmernega sinhronnega 3-bitnega števca z uporabo D flip-flopov: Zapišite tabelo prehajanja stanj in določite enačbe flip-flopov. Števec ima vhod SMER, ki določa smer štetja: Če je SMER='0', števec šteje naraščajoče po Grayevi kodi, sicer šteje padajoče dvojiško. Imena signalov so razvidna iz spodnje slike.



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.

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 JK	funkcija T
0	0	0	0	0	HOLD	HOLD
0	0	1	1	0	HOLD	HOLD
0	1	0	0	0	RESET	HOLD
0	1	1	0	1	RESET	INVERT
1	0	0	1	1	SET	INVERT
1	0	1	1	0	SET	HOLD
1	1	0	1	1	INVERT	INVERT
1	1	1	0	1	INVERT	INVERT

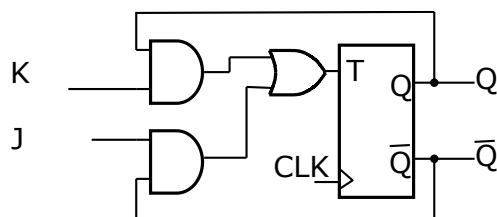
Iz tabele narišemo Veitchev diagram za vhod T v odvisnosti od vhodov J, K in trenutnega stanja $Q(t)$.

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

Vezje narišemo.

T :

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



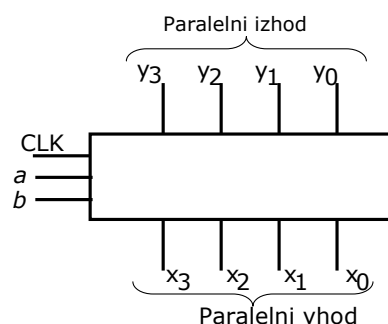
Rešitev 2. naloge:

Naloga zahteva realizacijo univerzalnega registra s funkcijami

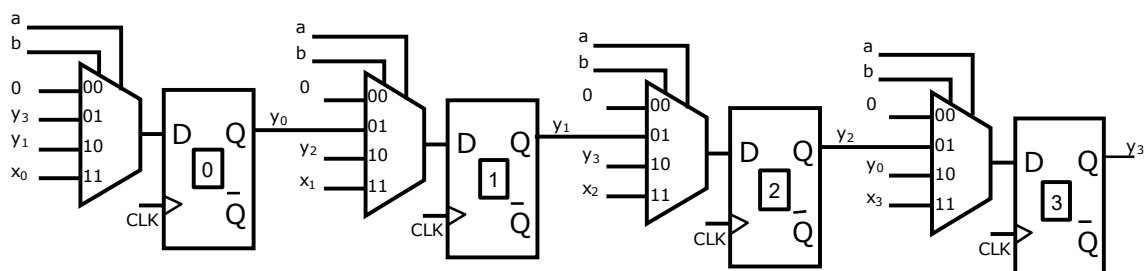
a	b	<i>funkcija</i>
0	0	briše vsebino registra (CLEAR)
0	1	rotira vsebino eno mesto levo (ROL)
1	0	rotira vsebino eno mesto desno (ROR)
1	1	vpiše vsebino s paralelnega vhoda (LOAD)

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 levo, torej nima serijskega vhoda in izhoda, ampak MSB bit y_3 vodimo na LSB bit y_0 . Podobno velja za rotacijo desno, kjer ampak LSB bit y_0 vodimo na MSB bit y_3 . Naloga zahteva realizacijo z 4/1 izbiralniki, s katerimi povežemo naslednje vhode glede na izbrane 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. Navzgor (SMER=0) šteje po Grayevi kodi, navzdol pa šteje dvojiško.

m _j	SMER	Trenutno			Naslednje		
		Q ₂	Q ₁	Q ₀	D ₂	D ₁	D ₀
					Q ₂	Q ₁	Q ₀
0	0	0	0	0	0	0	1
1	0	0	0	1	0	1	1
2	0	0	1	0	1	1	0
3	0	0	1	1	0	1	0
4	0	1	0	0	0	0	0
5	0	1	0	1	1	0	0
6	0	1	1	0	1	1	1
7	0	1	1	1	1	0	1
8	1	0	0	0	1	1	1
9	1	0	0	1	0	0	0
10	1	0	1	0	0	0	1
11	1	0	1	1	0	1	0
12	1	1	0	0	0	1	1
13	1	1	0	1	1	0	0
14	1	1	1	0	1	0	1
15	1	1	1	1	1	1	0

Naraščajoča 3-bitna Grayeva koda v desetiškem zapisu bi se glasila: 0,1,3,2,6,7,5,4,0.... Tabela prehajanja stanj števca za SMER='0' izpišemo tako, da gledamo prehode navedenega zaporedja.

Dodatnih stolpcev za vhode D-FF nam ni treba risati, saj velja da je $Q(t+1) = D$.

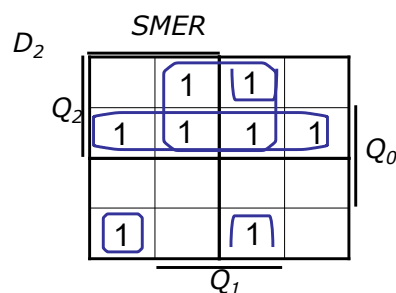
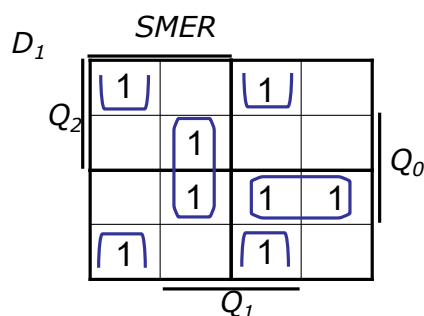
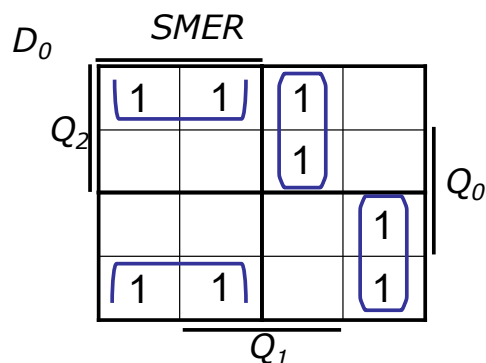
Iz tabele izrazimo funkcije za vse tri vhode:

$$D_0 = V(0,1,6-8,10,12,14)$$

$$D_1 = V(1,2,3,6,8,11,12,15)$$

$$D_2 = V(2,5-8,13-15)$$

Izrišemo pripadajoče Veitcheve diagrame:



Funkcije v diagramih minimiziramo in dobimo:

$$D_0 = \text{SMER} \cdot \overline{Q_0} + \overline{\text{SMER}} \cdot Q_2 \cdot Q_1 + \overline{\text{SMER}} \cdot \overline{Q_2} \cdot Q_0$$

$$D_1 = \text{SMER} \cdot Q_1 \cdot Q_0 + \text{SMER} \cdot \overline{Q_1} \cdot \overline{Q_0} + \overline{\text{SMER}} \cdot Q_1 \cdot \overline{Q_0} + \overline{\text{SMER}} \cdot \overline{Q_2} \cdot Q_0$$

$$D_2 = Q_2 \cdot Q_1 + Q_2 \cdot Q_0 + \overline{\text{SMER}} \cdot Q_2 \cdot Q_1 + \text{SMER} \cdot \overline{Q_2} \cdot \overline{Q_1} \cdot \overline{Q_0}$$

Rešitev 4. naloge:

Naloga zahteva realizacijo z Mealy-evim 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 sekvenc ne začneja z '0'.

