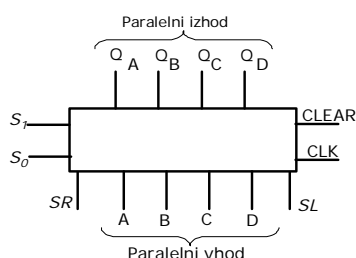


# RAZVOJ DIGITALNIH SISTEMOV

2. kolokvij

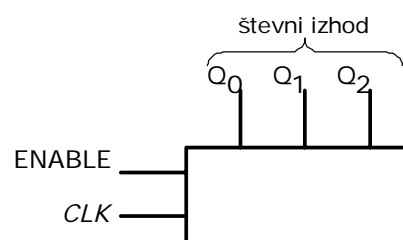
23. 1. 2015

1. Realizirajte JK-flip flop z uporabo RS- 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).



$S_1$	$S_0$	funkcija
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 3-bitnega sinhronnega števca navzgor z omogočanjem štetja (ENABLE) s T flip-flopi. Zapišite tabelo prehajanja stanj, določite enačbe vhodov T-FF in narišite vezje z uporabo signalov na sliki.



4. Narišite **diagram prehajanja stanj** za avtomat končnih stanj, ki ima vhod  $w$  in izhod  $z$ . Avtomat končnih stanj postavi izhod  $z='1'$  takoj, ko se na vhodu pojavi zaporedje **110** ali **101**, sicer je  $z='0'$ . Prekrivanje vzorcev je dovoljeno. Delovanje avtomata končnih stanj povzema spodnje časovno zaporedje vhoda in izhoda. Delovanje avtomata je povzema podano časovno zaporedje, pri čemer  $t_0 \dots t_{12}$  označujejo aktivne prehode signala ure  $CLK$ :

$CLK$	$t_0$	$t_1$	$t_2$	$t_3$	$t_4$	$t_5$	$t_6$	$t_7$	$t_8$	$t_9$	$t_{10}$	$t_{11}$	$t_{12}$
$w$	0	1	1	0	1	1	0	1	1	1	0	0	0
$z$	–	0	0	1	1	0	1	1	0	0	1	0	0

Čas pisanja je 60 minut. Vsaka naloga je vredna 10 točk.

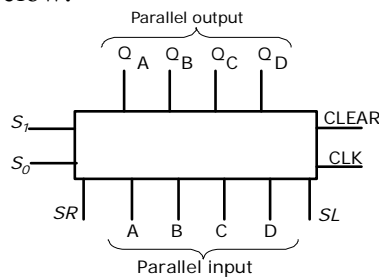
Na list z rešitvami se podpišite in napišite še vpisno številko ter kateri predmet pišete.

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

# DEVELOPMENT OF DIGITAL SYSTEMS

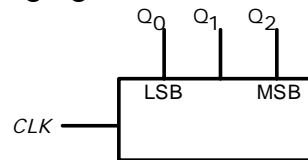
2<sup>nd</sup> Midterm Examination  
23. 1. 2015

1. Implement a JK type flip flop using a single RS flip flop and logic gates.
2. Draw the circuit of a 4 bit universal shift register using D type flip-flops and 4/1 multiplexers. The register comprises four content outputs  $Q_A$ ,  $Q_B$ ,  $Q_C$ ,  $Q_D$  and two function inputs  $S_0$  and  $S_1$ . It also has two shift operation inputs:  $SL$  (shift left) and  $SR$  (shift right). A LOAD operation will load the inputs  $A$ ,  $B$ ,  $C$ ,  $D$  into register locations  $Q_A$ ,  $Q_B$ ,  $Q_C$ ,  $Q_D$  respectively. An active low asynchronous  $CLEAR$  input sets the contents of register to zero. Register operations are specified in a table below.



$S_1$	$S_0$	function
0	0	HOLD
0	1	SHIFT RIGHT
1	0	SHIFT LEFT
1	1	LOAD

3. Synthesize a 3 bit binary up counter with ENABLE 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.



4. Draw a **state transition diagram** of a sequence detector FSM. Sequence detector has an input  $w$  and an output  $z$ . The FSM sets the output  $z='1'$  *immediately* after a sequence **110** or **101** is detected at its input. The sequences may overlap (i.e. input sequence 1101 must be detected twice). FSM operation is described in a test sequence given below, where  $t_0 \dots t_{12}$  denote active edge transitions of clock signal  $CLK$ :

$CLK$	$t_0$	$t_1$	$t_2$	$t_3$	$t_4$	$t_5$	$t_6$	$t_7$	$t_8$	$t_9$	$t_{10}$	$t_{11}$	$t_{12}$
$w$	0	1	1	0	1	1	0	1	1	1	0	0	0
$z$	—	0	0	1	1	0	1	1	0	0	1	0	0

Examination duration is 60 minutes. Each assignment is worth 10 points.

Please sign your answer sheet using your enrollment number. Solutions will be published on the course web page. Examination results will be announced on the course web page.

## 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 tri (HOLD, RESET, SET). Posamezno kombinacijo prehodov iz  $Q(t)$  v  $Q(t+1)$  je možno doseči na več načinov. Primer je prva kombinacija  $J=K=Q(t)=0$  pri čemer je  $Q(t+1)=0$ . Če je bil izhod FF prej 0 in je tudi v naslednjem stanju 0 sta vzroka za to lahko dva: HOLD ( $R=0, S=0$ ) ali RESET ( $R=1, S=0$ ), kar opišemo s kombinacijo ( $R=X, S=0$ ). Podobno lahko sklepamo za ostale kombinacije vhodov.

J	K	$Q(t)$	$Q(t+1)$	R	S	funkcija RS	funkcija JK
0	0	0	0	X	0	HOLD/RESET	HOLD
0	0	1	1	0	X	HOLD/SET	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 Veitcheva diagrama za vhoda R in S v odvisnosti od vhodov J, K in trenutnega stanja  $Q(t)$ .

R:

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

S:

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

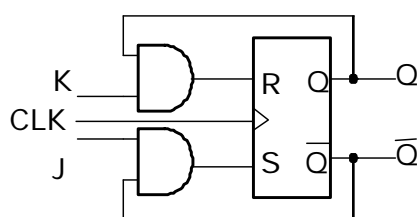
Dobljena diagrama minimiziramo in zapišemo enačbi vhodov J in K:

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

ter

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

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: <https://estudent.fri.uni-lj.si>

## Rešitev 2. naloge:

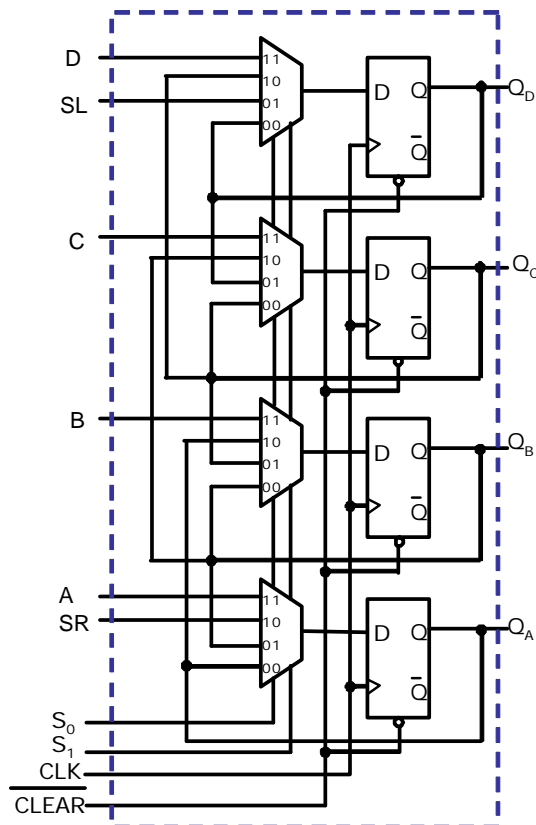
Vsako od operacij izpišemo v pravilnostno tabelo v kateri združimo funkcijska bita  $S_1$  in  $S_0$  in trenutno stanje na  $i$ -tem mestu registra  $Q_i(t)$ . Mesta registra od leve proti desni so  $Q_i = (Q_A, Q_B, Q_C, Q_D)$ . Realizacija z D flip-flopi nam analizo močno poenostavi, zaradi enačbe 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		
$Q_2$	$Q_1$	$Q_0$	$Q_2$	$Q_1$	$Q_0$	$T_2$	$T_1$	$T_0$
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

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 števec določimo enačbe T-FF:

Iz stolpca  $T_0$  se vidi:

$$T_0 = 1$$

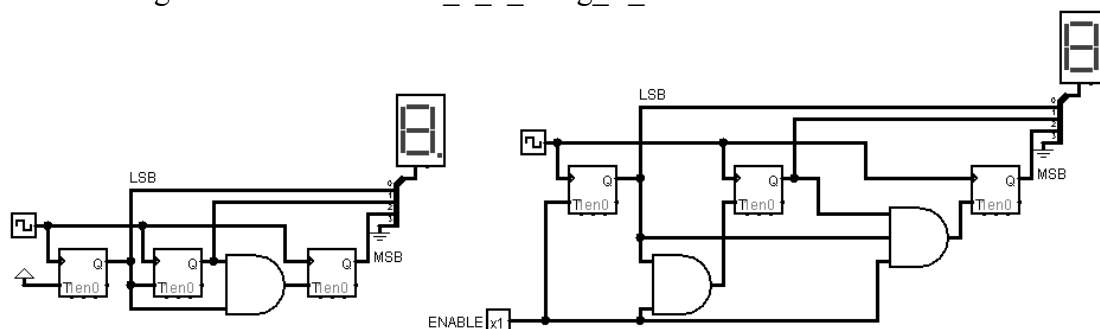
Z opazovanjem stolpcev trenutnega stanja določimo  $T_1$ :

$$T_1 = Q_0$$

Podobno lahko določimo  $T_2$ :

$$T_2 = Q_0 \cdot Q_1$$

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



Funkcijo omogočanja štetja bi lahko realizirali s pisanjem tabele prehajanje stanj števec za štiri spremenljivke (ENABLE,  $Q_0$ ,  $Q_1$ ,  $Q_2$ ). Z malo razmišljanja se izognemo dolgotrajnemu pisanju: Števec bo obstal (ENABLE='0'), če so vsi vhodi T-FF enaki '0'. Števec bo štel (ENABLE='1'), ko bodo vhodi T-FF lahko spreminjali stanje, torej se bo izhod prejšnje stopnje nespremenjen pojavil na vhodu naslednje stopnje  $T_{i+1}$ . Povedano strnemo v tabelo za vhod vsakega T-FF:

ENABLE	Izhod prejšnje stopnje	$T_{i+1}$
0	0	0
0	1	0
1	0	0
1	1	1

Funkcijo ENABLE realiziramo z dvovhodnimi AND vrati pred vhodom posameznega T-FF. Na vhoda AND vrat vodimo ENABLE in izhod iz prejšnje stopnje štetja, kot kaže desni del slike. Na LSB mestu štetja AND vrat ne potrebujemo, ampak ENABLE priključimo neposredno na vhod T-FF.

Čas pisanja je 60 minut. Vsaka naloga je vredna 10 točk.

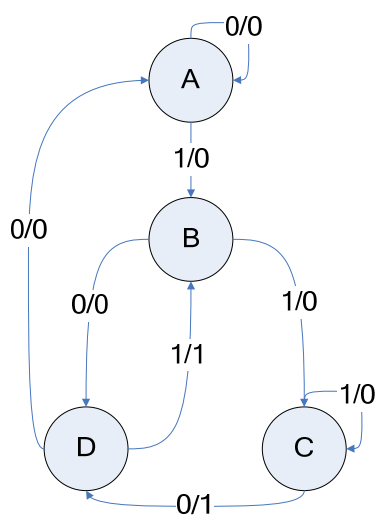
Na list z rešitvami se podpišite in napišite še vpisno številko ter kateri predmet pišete.

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

#### Rešitev 4. naloge:

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'.



Delovanje avtomata preizkusimo na testnem zaporedju:

stanje	A	B	D	B	D	A	B	C	D	B	D	A	B	C	D	B	D	B	C	D	A	B	C	D
w	0	1	0	1	0	0	1	1	0	1	0	0	1	1	0	1	0	1	1	0	0	1	1	0
z	0	0	0	1	0	0	0	0	1	1	0	0	0	0	1	1	0	1	0	1	0	0	0	1

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>