

Osnovi digitalne elektronike (13E042OD)

- VEŽBE -

Sekvencijalna kola

- 1 Potrebno je projektovati potpuni 3-bitni brojač koristeći ivične T flip-flopove sa EN ($ENable$) ulazom i asinhronim ulazima za SET i $RESET$ aktivnim u logičkoj nuli i potrebna osnovna logička kola. Takođe potrebno je obezbediti da za vreme trajanja aktivne vrednosti signala $RESET$ (aktivnog u logičkoj jedinici) brojač bude u stanju nula ($Q_2Q_1Q_0 = "000"$).
- Projektovati asinhroni brojač koji broji unapred.
 - Projektovati asinhroni brojač koji broji unazad.
 - Projektovati asinhroni brojač koji broji unapred ili unazad u zavisnosti od signala $SMER$ (ako je $SMER = "1"$, onda brojač broji unapred, u suprotnom broji unazad).
 - Ako je kašnjenje flip-flopova $t_{dff} = 10$ ns, vreme postavljanja ulaza flip-flopova $t_{setup} = 6$ ns, vreme držanja ulaza flip-flopova $t_{hold} = 4$ ns i kašnjenje logičkih kola $t_{dlk} = 5$ ns, odrediti maksimalnu učestanost rada brojača iz c).
 - Projektovati sinhroni brojač koji broji unapred i za njega ponoviti tačku d).

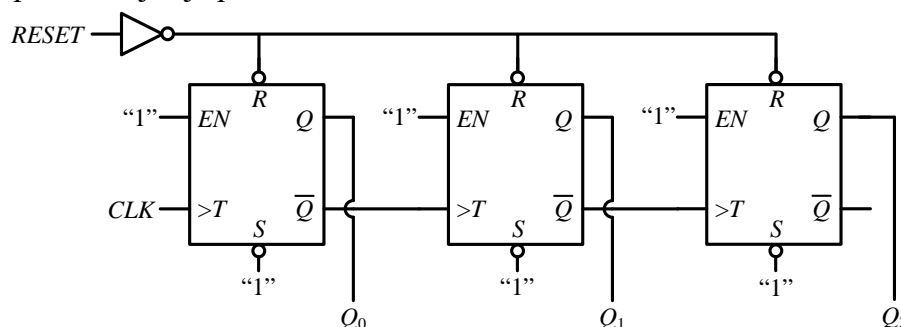
Rešenje:

a) Potpuni trobitni brojač se može naći u 8 različitih stanja datih u tabeli stanja/prelaza u tabeli 1.1. Ako broji unapred, iz stanja "000" prelazi u stanje "001", zatim na sledeću uzlaznu ivicu signala takta prelazi iz stanja "001" u stanje "010" itd. Iz tabele se može primetiti da signal Q_0 menja vrednost na svaku uzlaznu ivicu signala takta, što znači da će taj signal biti signal na izlazu prvog T flip-flopa na čiji je T ulaz doveden signal takta. Dalje, vidi se da signal Q_1 menja vrednost samo ako signal Q_0 prelazi sa "1" na "0", tj. pri njegovoj silaznoj ivici, što je ekvivalentno uzlaznoj ivici signala $\overline{Q_0}$, pa se na T ulaz drugog T flip-flopa dovodi signal $\overline{Q_0}$. Slično, signal Q_2 menja vrednost pri silaznoj ivici signala Q_1 , što je ekvivalentno uzlaznoj ivici signala $\overline{Q_1}$, pa se na T ulaz drugog T flip-flopa dovodi signal $\overline{Q_1}$. Da bi svi flip-

Q_2	Q_1	Q_0	Q_2^+	Q_1^+	Q_0^+
0	0	0	0	0	1
0	0	1	0	1	0
0	1	0	0	1	1
0	1	1	1	0	0
1	0	0	1	0	1
1	0	1	1	1	0
1	1	0	1	1	1
1	1	1	0	0	0

Tabela 1.1 – Tabela stanja/prelaza brojača unapred

flopovi radili, potrebno je na sve EN ulaze dovesti logičku jedinicu. Na kraju, kako je signal $RESET$ aktivan u logičkoj jedinici, a ulaz za asinhroni reset flip-flopova aktivan u logičkoj nuli, potrebno je invertovati signal $RESET$ i njegovu invertovanu vrednost dovesti na asinhronne ulaze za reset svih flip-flopova. Opisani brojač je prikazan na slici 1.1.

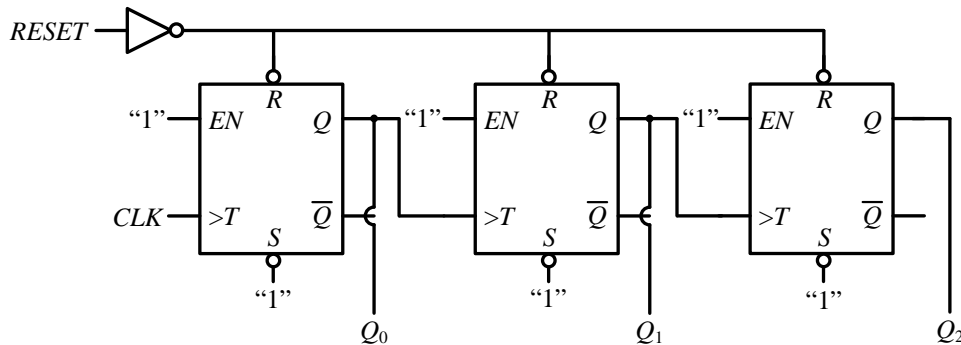


Slika 1.1 – Asinhroni 3-bitni potpuni brojač unapred

b) Potpuni trobitni brojač koji broji unazad se može naći u 8 različitih stanja datih u tabeli stanja/prelaza u tabeli 1.2. Sličnom analizom kao u tački a) dolazimo do zaključka da signal Q_1 menja vrednost pri uzlaznoj ivici signala Q_0 , a da signal Q_2 menja vrednost pri ulaznoj ivici signala Q_1 . Šema brojača koji broji unazad je data na slici 1.2.

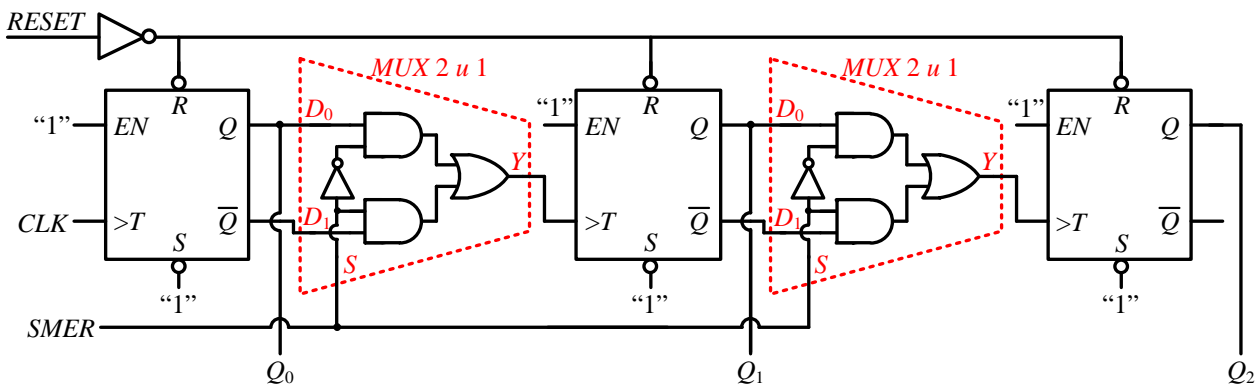
Q_2	Q_1	Q_0	Q_2^+	Q_1^+	Q_0^+
1	1	1	1	1	0
1	1	0	1	0	1
1	0	1	1	0	0
1	0	0	0	1	1
0	1	1	0	1	0
0	1	0	0	0	1
0	0	1	0	0	0
0	0	0	1	1	1

Tabela 1.2 – Tabela stanja/prelaza brojača koji broji unazad



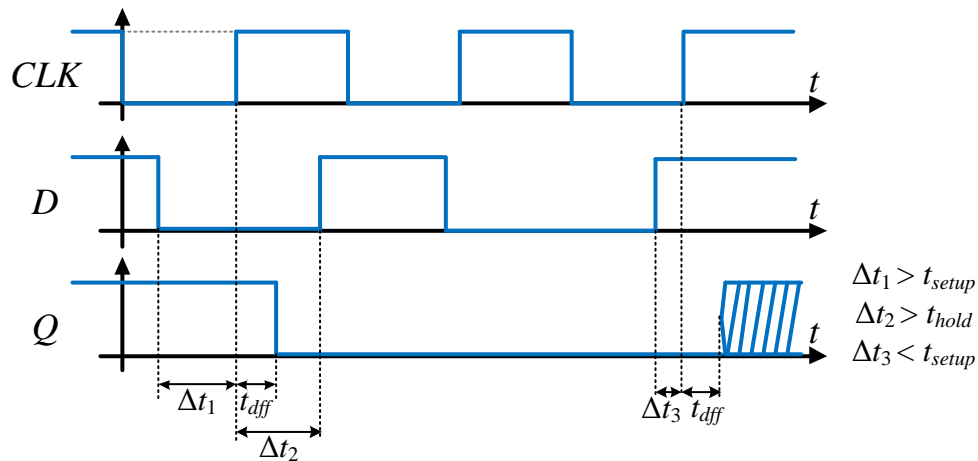
Slika 1.2 – Asinhroni 3-bitni potpuni brojač unazad

c) Na osnovu tačke a) i tačke b), možemo zaključiti da je, i kod brojača koji broji unapred i kod brojača koji broji unazad, ulaz T flip-flopa čiji je izlaz Q_0 povezan na signal takta CLK . Na T ulaze flip-flopa 1 i 2 se dovode signali Q_0 ili \bar{Q}_0 , odnosno Q_1 ili \bar{Q}_1 , u zavisnosti od toga da li brojač broji unapred ili unazad, pa je zbog toga najzgodnije u ovoj tački iskoristiti multipleksere 2 u 1 kojima se na selekcionu ulazu dovodi signal $SMER$ koji definiše smer brojanja. Na slici 1.3 je prikazan traženi brojač, pri čemu su multiplekseri realizovani korišćenjem isključivo osnovnih logičkih kola.

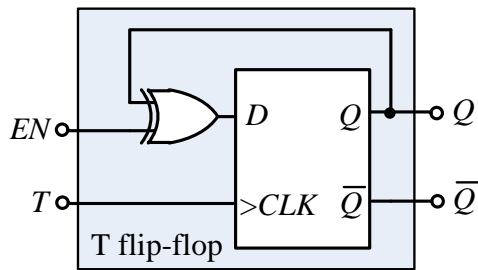


Slika 1.3 – Asinhroni 3-bitni potpuni brojač sa mogućnošću odabira smera brojanja

d) Najpre, objasnimo šta su vreme postavljanja i vreme držanja ulaznog signala. Neka imamo D flip-flop. Vreme postavljanja (t_{setup}) je minimalno vreme koje je potrebno da signal D bude stabilan pre ulazne ivice signala takta CLK , da bi se na izlazu Q nakon kašnjenja flip-flopa t_{diff} pojavila vrednost signala D . Vreme držanja (t_{hold}) je minimalno vreme koje je potrebno da signal D bude stabilan nakon ulazne ivice signala takta CLK , da bi se na izlazu Q pojavila vrednost signala D . Ako ova vremena nisu zadovoljena, onda se ne može predvideti šta će se pojaviti na izlazu Q (slika 1.4).

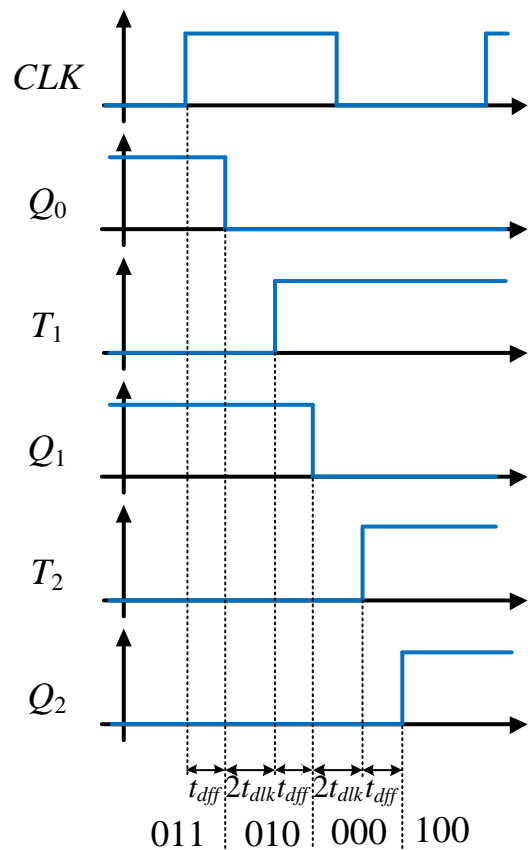


Slika 1.4 – Vremenski dijagrami signala D flip-flopa kada su zadovoljena vremena postavljanja i držanja (levo) i kada ona nisu zadovoljena



Slika 1.5 – Realizacija T flip-flopa sa ENABLE ulazom

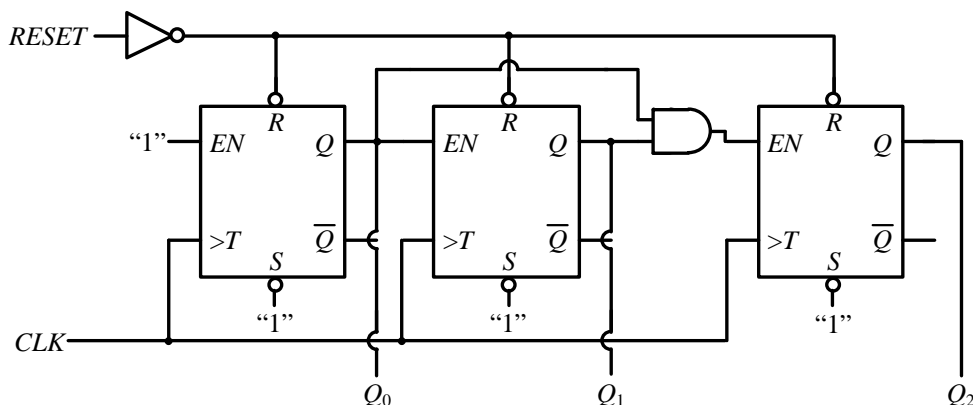
Da bi brojač ispravno radio, potrebno je da vreme između dve uzlazne ivice signala takta bude dovoljno da se završe svi prelazni procesi. Na slici 1.5 je prikazana najčešća realizacija T flip-flopa sa *EN* ulazom. Sa slike se vidi da se vremena postavljanja i držanja odnose samo na *ENABLE* ulaz, pa za asinhroni brojač sa slike 1.3. nije potrebno voditi računa o ovim vremenima jer je *EN* uvek na “1”. Da bi se završili svi prelazni procesi kod navedenog asinhronog brojača potrebno je da signal ispropagira kroz sva tri flip-flopa, tj. da bi se postavio izlaz trećeg flip flopa, potrebno je da se prethodno postave izlazi prvog i drugog flip-flopa i da signal ispropagira kroz sva logička kola do T ulaza trećeg flip-flopa. Kašnjenje izlaza trećeg flip-flopa će odrediti učestanost rada i ono iznosi $t_{dQ2} = t_{dff} + 2t_{dlk} + t_{dff} + 2t_{dlk} + t_{dff} = 3t_{dff} + 4t_{dlk} = 50$ ns, pa je maksimalna učestanost rada $f_{max} = 1/t_{dQ2} = 20$ MHz. Primer koji ilustruje ove prelaze (sa $Q_2Q_1Q_0 = “011”$ na $Q_2Q_1Q_0 = “100”$) je dat na slici 1.6.



Slika 1.6 – Vremenski dijagrami signala brojača pri prelasku iz stanja “011” u stanje “100”

e) Kod sinhronih brojača svi flip-flopovi rade na jedan signal takta i postavljaju izlaze istovremeno. S obzirom na to da je jedini ivični ulaz kod T flip-flopa ulaz *T*, signal takta moramo dovesti na *T* ulaze svih flip-flopova, a za kontrolu koji od flip-flopova menja vrednost koriste se *EN* ulazi.

Posmatrajući tabelu 1.1, može se zaključiti da flip-flop 1 menja stanje kada je prethodno signal Q_0 bio na "1". Takođe, flip-flop 2 menja stanje kada su prethodno signali Q_0 i Q_1 bili na "1". Šema sinhronog brojača koji broji unapred data je na slici 1.7.



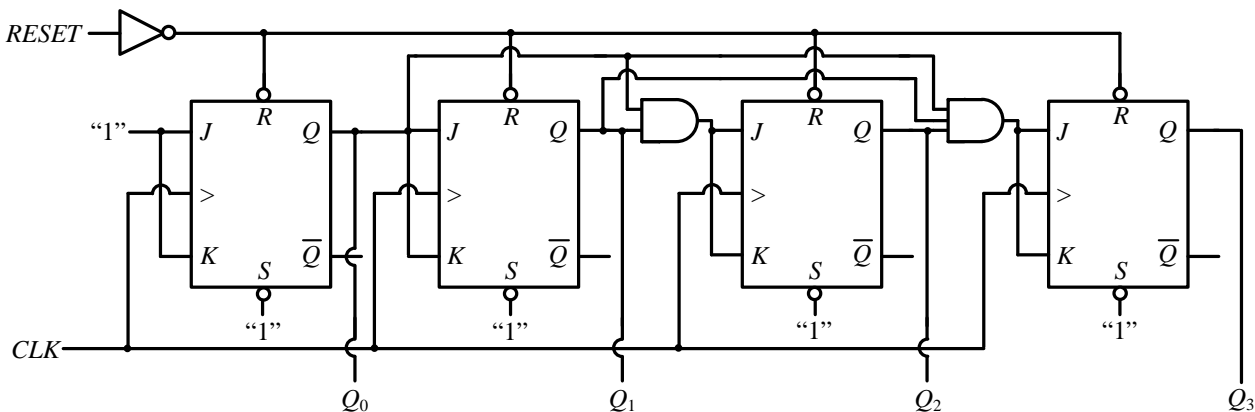
Slika 1.7 – Sinhroni 3-bitni potpuni brojač unapred

Izlazi svih flip-flopora imaju jednako kašnjenje u odnosu na uzlaznu ivicu signala takta i ono iznosi t_{dff} . Da bi brojač ispravno radio, potrebno je da u trenutku sledeće uzlazne ivice signala takta, svi flip-flopori imaju validne vrednosti na svojim EN ulazima. Stoga je potrebno pronaći najdužu putanju od signala takta do EN ulaza nekog od flip-flopora brojača. Takođe, potrebno je i uračunati vreme postavljanja, pa je minimalno vreme koje je potrebno da prođe između dve uzlazne ivice signala takta $\Delta t = t_{dff} + t_{dlk} + t_{setup} = 21$ ns. Maksimalna učestanost rada je $f_{max} = 1/\Delta t = 47,6$ MHz.

- Projektovati potpuni 4-bitni brojač unapred koristeći ivične JK flip-floporve sa asinhronim ulazima za SET i $RESET$ aktivnim u logičkoj nuli i potrebna osnovna logička kola. Takođe potrebno je obezbediti da za vreme trajanja aktivne vrednosti signala $RESET$ (aktivnog u logičkoj jedinici) brojač bude u stanju nula ($Q_3Q_2Q_1Q_0 = "0000"$). Ako je kašnjenje flip-flopora $t_{dff} = 10$ ns, vreme postavljanja ulaza flip-flopora $t_{setup} = 6$ ns, vreme držanja ulaza flip-flopora $t_{hold} = 4$ ns i kašnjenje logičkih kola $t_{dlk} = 5$ ns, odrediti maksimalnu učestanost rada brojača.

Rešenje:

JK flip-flop se ponaša kao T flip-flop sa $ENABLE$ ulazom ako mu se povežu J i K ulazi. Na osnovu prethodnog zadatka, lako se dolazi do šeme potpunog 4-bitnog brojača sa JK flip-floporima (slika 2.1).



Slika 2.1 – Sinhroni 4-bitni potpuni brojač unapred sa JK flip-floporima

Maksimalna učestanost rada je, na osnovu prethodnog zadatka $f_{max} = 1/\Delta t = 47,6$ MHz, gde je $\Delta t = t_{dff} + t_{dlk} + t_{setup} = 21$ ns.

Domaći:

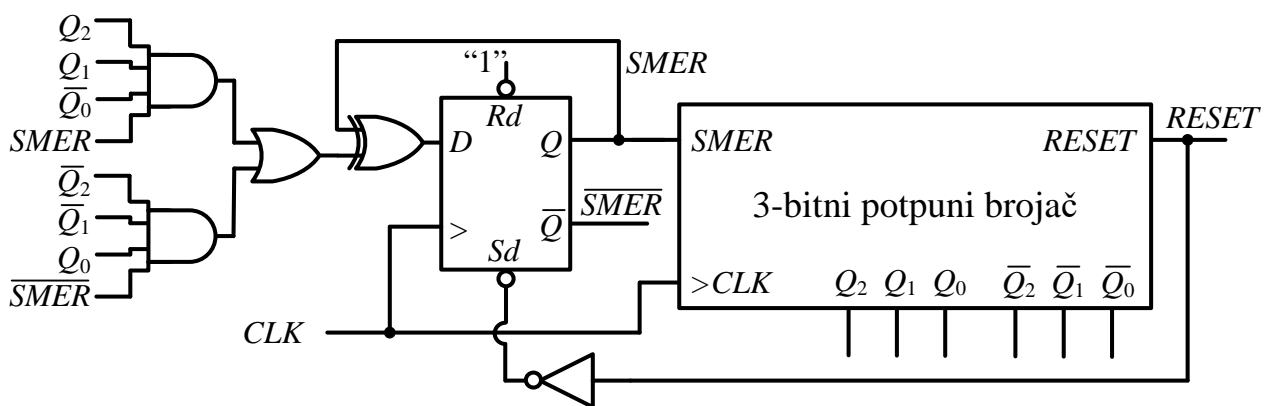
- Ponoviti zadatak ako je potrebno projektovati sinhroni brojač koji ima ulazni signal *SMER* kojim se definiše smer brojanja, kao u zadatku 1.

- 3** Ako je na raspolaganju 3-bitni potpuni brojač sa mogućnošću odabira smera brojanja iz zadatka 1, uz pomoć jednog dodatnog *D* flip-flopa koji ima asinhronu ulazu za *SET* i *RESET* aktivne na logičkoj nuli i potrebnih osnovnih logičkih kola, projektovati brojač koji broji u sekvenci 0-1-2-3-4-5-6-7-6-5-4-3-2-1-0-1-2-... Potrebno je obezbediti da za vreme trajanja aktivne vrednosti signala *RESET* (aktivnog u logičkoj jedinici) brojač bude u stanju nula ($Q_2Q_1Q_0 = "000"$), a neposredno nakon ukidanja signala *RESET*, brojač treba da broji unapred.

Rešenje:

Očigledno je potreban još jedan memorijski element koji će da pamti smer brojanja sve dok brojač ne dođe do vrednosti 7 ili 0 kada treba promeniti smer. Za ovu namenu je pogodan T flip-flop sa *EN* ulazom, a koji se realizuje kao na slici 1.5 iz zadatka 1. Na T ulaz flip-flopa treba dovesti isti signal takta kao I na potpuni brojač, a aktivirati signal *EN* samo u situacijama kada treba promeniti smer. To su slučajevi kada 3-bitni potpuni brojač treba da pređe iz stanja 6 u stanje 7 i kada treba da pređe iz stanja 1 u stanje 0. Realizacija ovog brojača je data na slici 3.1.

Važno: Obratiti pažnju da se ovo rešenje razlikuje od rešenja sa časa. U rešenju sa časa može da nastane problem u situacijama kada se signal reseta ukine pre silazne ivice signala takta, pa se tada dešava da brojač ima samo dva stanja i da broji u sekvenci 0-7-0-7-...

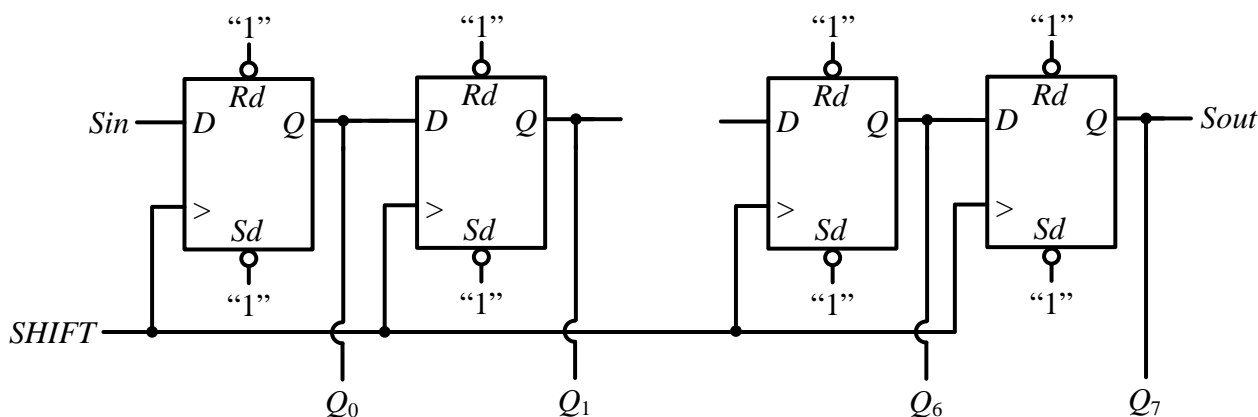


Slika 3.1 – Sinhroni brojač koji broji u sekvenci 0-1-2-3-4-5-6-7-6-5-4-3-2-1-0-1-2-...

- 4 Ako su na raspolaganju D flip-flopovi koji imaju asinhronne ulaze za SET i $RESET$ aktivne na logičkoj nuli i proizvoljan broj osnovnih logičkih kola, projektovati 8-bitni pomerački registar koji ima signale $SHIFT$ (takt), S_{in} ($Serial In$), S_{out} ($Serial Out$), $D_{in}[7:0]$ (ulazi za asinhroni paralelni upis) i $Q[7:0]$ (trenutno stanje registra), kao i signal $LOAD$ koji kada je aktivan omogućava asinhroni upis podatka $D_{in}[7:0]$ u pomerački registar.

Rešenje:

Šema pomeračkog registra bez mogućnosti paralelnog upisa data je na slici 4.1.



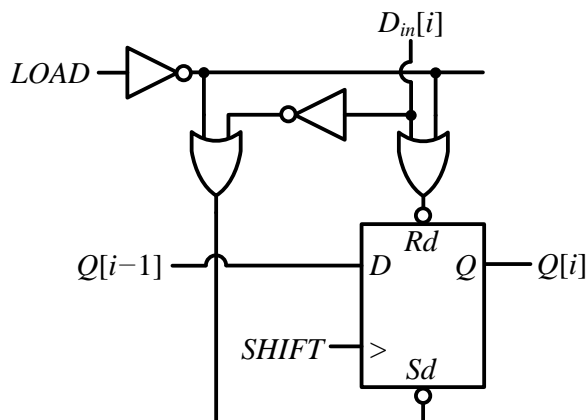
Slika 4.1 – 8-bitni pomerački registar bez paralelnog upisa

S obzirom na to da su na raspolaganju flip-flopovi sa asinhronim ulazima sa set i reset, pogodno je za asinhroni upis iskoristiti ove ulaze. Ako je signal koji treba učitati u i -ti flip-flop $D_{in}[i] = "1"$, tada je potrebno setovati flip flop, tj. na ulaz $\overline{Sd}[i]$ treba dovesti signal "0", a na ulaz $\overline{Rd}[i]$ treba dovesti signal "1", pod uslovom da je signal $LOAD$ na "1". Slično, ako je signal koji treba učitati u i -ti flip-flop $D_{in}[i] = "0"$, tada je potrebno resetovati flip flop, tj. na ulaz $\overline{Sd}[i]$ treba dovesti signal "1", a na ulaz $\overline{Rd}[i]$ treba dovesti signal "0", a u suprotnom su oba signala $\overline{Sd}[i]$ i $\overline{Rd}[i]$ postavljeni na "1" (tabela 4.1). Na osnovu ovoga, zaključujemo da su

Tabela 4.1 – Signali potrebni za asinhroni upis

$LOAD$	$D_{in}[i]$	$\overline{Sd}[i]$	$\overline{Rd}[i]$
0	0	1	1
0	1	1	1
1	0	1	0
1	1	0	1

$\overline{Sd}[i] = \overline{LOAD} \cdot \overline{D_{in}[i]} = \overline{LOAD} + D_{in}[i]$ i $\overline{Rd}[i] = \overline{LOAD} \cdot D_{in}[i] = \overline{LOAD} + \overline{D_{in}[i]}$. Opisani paralelni upis je isti za sve flip-flopove u pomeračkom registru (slika 4.2).



Slika 4.2 – Asinhroni paralelni upis u ćeliju pomeračkog registra

- 5 Koristeći minimalan broj ivičnih JK flip-floпова i minimalan broj proizvoljnih osnovnih logičkih kola projektovati sinhroni brojač koji broji u sekvenci 0-2-10-4-8-11-7-9-12-0-... U slučaju da se brojač nađe u nekom nedozvoljenom stanju, potrebno je da nailaskom prve sledeće uzlazne ivice signala takta pređe u stanje "0000". Na ulazu brojača, pored signala takta, dovodi se i kontrolni signal *LOAD* (aktivan na logičkoj jedinici) kojim se omogućava sinhroni upis u brojač podatka $D[3:0]$.

Rešenje:

Brojač ima 9 stanja, pa je za njegovu realizaciju neophodno iskoristiti 4 flip-flopa. Tabela stanja/prelaza u kojoj se vidi koje je sledeće stanje brojača $Q_3^+Q_2^+Q_1^+Q_0^+$ nakon stanja $Q_3Q_2Q_1Q_0$ data je u tabeli 5.1. Sledeće stanje brojača, očigledno je određeno trenutnim stanjem brojača. Tako

Tabela 5.1 – Tabela stanja/prelaza brojača

Q_3	Q_2	Q_1	Q_0	Q_3^+	Q_2^+	Q_1^+	Q_0^+
0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0
0	0	1	0	1	0	1	0
0	0	1	1	0	0	0	0
0	1	0	0	1	0	0	0
0	1	0	1	0	0	0	0
0	1	1	0	0	0	0	0
0	1	1	1	1	0	0	1
1	0	0	0	1	0	1	1
1	0	0	1	1	1	0	0
1	0	1	0	0	1	0	0
1	0	1	1	0	1	1	1
1	1	0	0	0	0	0	0
1	1	0	1	0	0	0	0
1	1	1	0	0	0	0	0
1	1	1	1	0	0	0	0

na primer, ako je trenutno stanje brojača 0010 potrebno je na prvu sledeću uzlaznu ivicu signala takta setovati flip-floповe 3 i 1, a resetovati flip-floповe 2 i 0. Tabela stanja/pobude JK flip flopa je data u tabeli 5.2. Oznaka *b* znači da je nevažno da li će odgovarajući signal imati vrednost 0 ili 1. Npr. ako je trenutno stanje $Q=0$, a naredno $Q^+=1$, potrebno je vrednost signala J postaviti na 1, dok je za signal K nebitno, u oba slučaja će se setovati flip-flop.

Tabela 5.2 – Tabela stanja/pobude JK flip-flopa

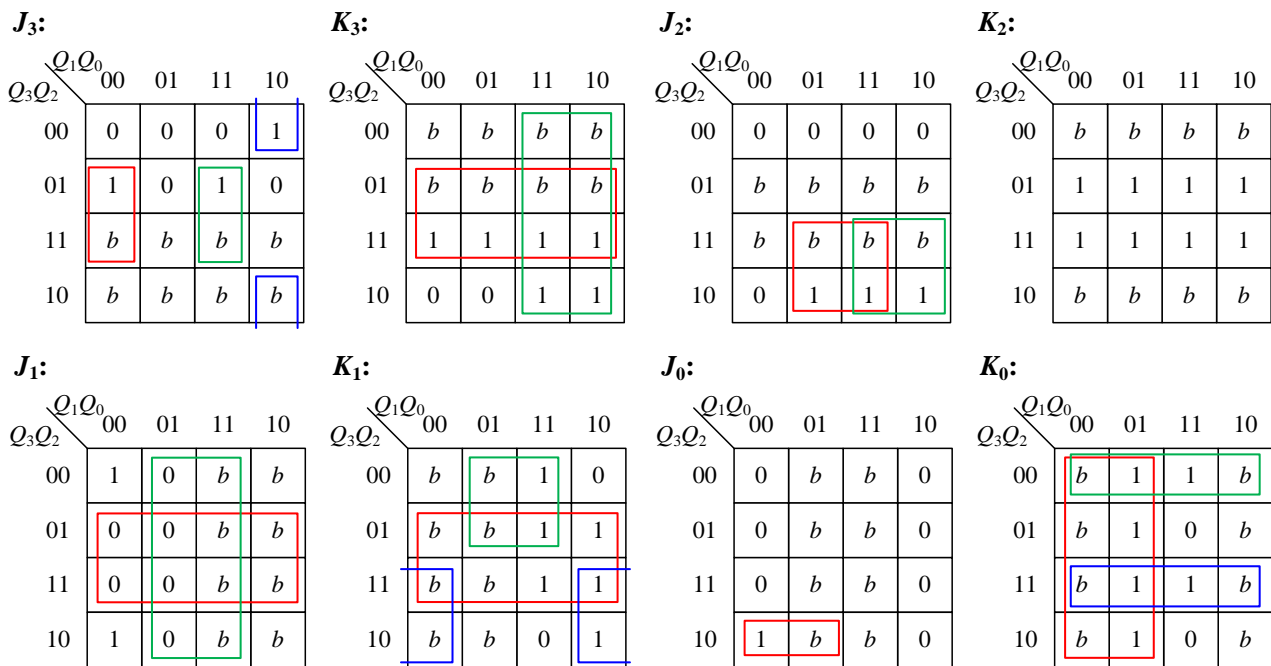
Q	Q^+	J	K
0	0	0	<i>b</i>
0	1	1	<i>b</i>
1	0	<i>b</i>	1
1	1	<i>b</i>	0

Na osnovu toga se može napraviti tabela stanja/pobude celog brojača, data u tabeli 5.3.

Tabela 5.3 – Tabela stanja/pobude brojača

Q_3	Q_2	Q_1	Q_0	Q_3^+	Q_2^+	Q_1^+	Q_0^+	J_3	K_3	J_2	K_2	J_1	K_1	J_0	K_0
0	0	0	0	0	0	1	0	0	<i>b</i>	0	<i>b</i>	1	<i>b</i>	0	<i>b</i>
0	0	0	1	0	0	0	0	0	<i>b</i>	0	<i>b</i>	0	<i>b</i>	<i>b</i>	1
0	0	1	0	1	0	1	0	1	<i>b</i>	0	<i>b</i>	<i>b</i>	0	0	<i>b</i>
0	0	1	1	0	0	0	0	0	<i>b</i>	0	<i>b</i>	<i>b</i>	1	<i>b</i>	1
0	1	0	0	1	0	0	0	1	<i>b</i>	<i>b</i>	1	0	<i>b</i>	0	<i>b</i>
0	1	0	1	0	0	0	0	0	<i>b</i>	<i>b</i>	1	0	<i>b</i>	<i>b</i>	1
0	1	1	0	0	0	0	0	0	<i>b</i>	<i>b</i>	1	<i>b</i>	1	0	<i>b</i>
0	1	1	1	1	0	0	1	1	<i>b</i>	<i>b</i>	1	<i>b</i>	1	<i>b</i>	0
1	0	0	0	1	0	1	1	<i>b</i>	0	0	<i>b</i>	1	<i>b</i>	1	<i>b</i>
1	0	0	1	1	1	0	0	<i>b</i>	0	1	<i>b</i>	0	<i>b</i>	<i>b</i>	1
1	0	1	0	0	1	0	0	<i>b</i>	1	1	<i>b</i>	<i>b</i>	1	0	<i>b</i>
1	0	1	1	0	1	1	1	<i>b</i>	1	1	<i>b</i>	<i>b</i>	0	<i>b</i>	0
1	1	0	0	0	0	0	0	<i>b</i>	1	<i>b</i>	1	0	<i>b</i>	0	<i>b</i>
1	1	0	1	0	0	0	0	<i>b</i>	1	<i>b</i>	1	0	<i>b</i>	<i>b</i>	1
1	1	1	0	0	0	0	0	<i>b</i>	1	<i>b</i>	1	<i>b</i>	1	0	<i>b</i>
1	1	1	1	0	0	0	0	<i>b</i>	1	<i>b</i>	1	<i>b</i>	1	<i>b</i>	1

Na osnovu tabele 5.3 mogu se kreirati Karnoove mape za svaki od ulaza J_i i K_i , prikazane na slici 5.1.



Slika 5.1 – Karnoove mape logičkih funkcija za pobude flip-floпова бројача

Logičke funkcije ulaza su na osnovu slike 5.1:

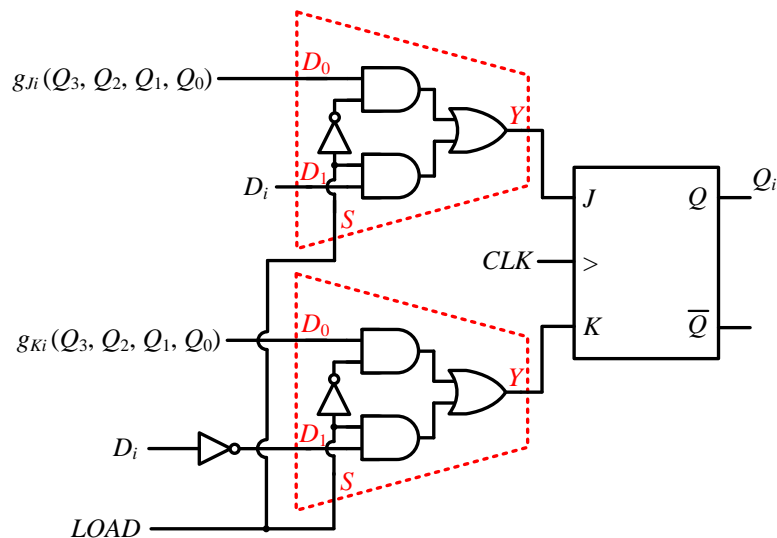
$$g_{J3}(Q_3, Q_2, Q_1, Q_0) = Q_2 \overline{Q_1} \overline{Q_0} + Q_2 Q_1 Q_0 + \overline{Q_2} Q_1 \overline{Q_0}, \quad g_{K3}(Q_3, Q_2, Q_1, Q_0) = Q_2 + Q_1$$

$$g_{J2}(Q_3, Q_2, Q_1, Q_0) = Q_3 Q_0 + Q_3 Q_1 = Q_3(Q_0 + Q_1), \quad g_{K2}(Q_3, Q_2, Q_1, Q_0) = "1"$$

$$g_{J1}(Q_3, Q_2, Q_1, Q_0) = \overline{Q_2} \cdot \overline{Q_0} = \overline{Q_2 + Q_0}, \quad g_{K1}(Q_3, Q_2, Q_1, Q_0) = Q_2 + Q_3 \overline{Q_0} + \overline{Q_3} Q_0 = Q_2 + Q_3 \oplus Q_0$$

$$g_{J0}(Q_3, Q_2, Q_1, Q_0) = Q_3 \overline{Q_2} \overline{Q_1}, \quad g_{K0}(Q_3, Q_2, Q_1, Q_0) = \overline{Q_1} + \overline{Q_3} \overline{Q_2} + Q_3 Q_2 = \overline{Q_1} + Q_3 \oplus Q_2$$

Sinhroni paralelni upis se najlakše obezbeđuje postavljanjem multipleksera na ulaze J i K svih flip-floпова. Primer i -tog flip-floпа traženog бројача prikazan je na slici 5.2.



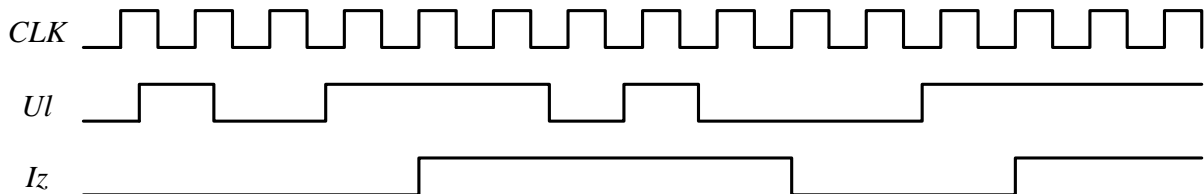
Slika 5.2 – i -ta ćelija бројача

6 Projektovati sinhronu sekvencijalnu mrežu kojom se vrši debaunsiranje ulaznog signala. Vrednost ulaznog signala se preslikava na izlaz ako su obuhvaćene najmanje 2 uzlazne ivice taktnog signala. Vremenski dijagram ulaznih i izlaznih signala je prikazan na slici. Odrediti:

a) Tabelu stanja/izlaza, tabelu prelaza/izlaza, i tabelu pobude/izlaza ako su za realizaciju na raspolaganju ivični D flip-flovi i proizvoljan broj osnovnih logičkih kola.

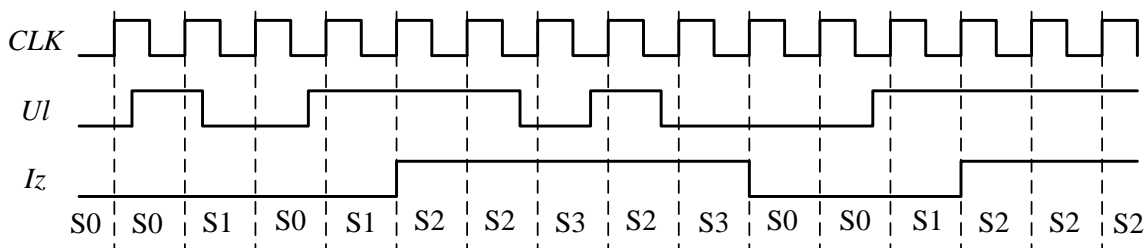
b) Nacrtati dijagram stanja ove sekvencijalne mreže

c) Realizovati mrežu korišćenjem ivičnih D flip-flova



Rešenje:

a) Na osnovu vremenskih dijagrama i opisa rada mreže mogu se identifikovati različita stanja u kojima se sistem može naći, što je prikazano na slici 6.1.



Slika 6.1. Vremenski dijagrami i stanja sekvencijalne mreže

Najpre posmatranjem oblika izlaznog signala može se zaključiti da se njegova vrednost menja sinhrono sa uzlaznom ivicom signala takta, tako da je pogodno da i projektovana sekvencijalna mreža menja stanje sa uzlaznom ivicom signala takta.

U početnom trenutku vrednost izlaznog signala je 0 i mreža se nalazi u stanju S0. Ako je na sledeću uzlaznu ivicu taktnog signala vrednost ulaza 0 mreža ostaje u stanju S0. Ako je pak na sledeću uzlaznu ivicu vrednost ulaznog signala jednaka 1 prelazi se u novo stanje S1. Ovo stanje i dalje održava vrednost izlaza na 0 ali označava da je obuhvaćena jedna uzlazna ivica taktnog signala aktivnom vrenošću ulaznog signala. Ako je na sledeću uzlaznu ivicu signala takta vrednost ulaznog signala 1 mreža prelazi u stanje S2 u kojem je vrednost izlaznog signala 1. Dakle odlaganje preslikavanja ulaznog signala na izlaz je dobijeno prelaskom mreže iz stanja S0 u S2 preko stanja S1. Ako pak za vreme dok je mreža u stanju S1 na sledeću uzlaznu ivicu signala takta vrednost ulaznog signala bude 0 mreža se vraća u stanje S0. Na ovaj način je dobijeno ignorisanje ulaznih impulsa čije je trajanje kraće od periode taktnog signala. Za promenu vrednosti izlaznog signala sa 1 na 0 na sličan način dobijamo da mreža prelazi iz stanja S2 u stanje S0 kroz jedno međustanje, u ovom slučaju S3.

Određena stanja i prelazi mreže su sumirani u tabeli stanja/izlaza. Tabela stanja/izlaza je prikazana u tabeli 6.1.

Tabela 6.1. Tabela stanja/izlaza

		Ul		Iz
		0	1	
S	S0	S0	S1	0
	S1	S0	S2	0
	S2	S3	S2	1
	S3	S0	S2	1

Dalje je potrebno kodovati svako stanje odogovarajućim promenljivama stanja. Kako projektovana sekvencijalna mreža prolazi kroz 4 stanja to su dovoljne 2 promenljive stanja (2 flip-flopa). Princip kodovanja stanja se zasniva na tri pravila:

- 1) inicijalno stanje mreže kodovati svim nulama ili svim jedinicama
- 2) obezbediti da se izborom kodovanja stanja, prilikom najvećeg broja prelaza, menja minimalan broj bita (stanja mreže)
- 3) obezbediti da funkcija izlaza bude minimalne kompleksnosti

Vodeći se prethodnim pravilima dobija se tabela prelaza/izlaza prikazana u tabeli 6.2.

Tabela 6.2. Tabela prelaza/izlaza

		Ul		Iz
		0	1	
Q_1Q_0	00	00	01	0
	01	00	11	0
	11	10	11	1
	10	00	11	1

$Q_1^+Q_0^+$

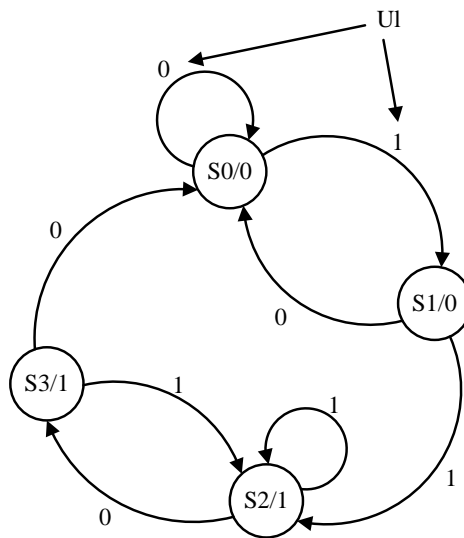
Koristeći tabelu 6.2. i karakterističnu jednačinu D flip-flopa $Q^+ = D$ dolazi se do tabele pobude/izlaza prikazane u tabeli 6.3.

Tabela 6.3. Tabela pobude/izlaza

		Ul		Iz
		0	1	
Q_1Q_0	00	00	01	0
	01	00	11	0
	11	10	11	1
	10	00	11	1

D_1D_0

b) Na osnovu tabele 6.1. dobija se dijagram stanja projektovane sekvencijalne mreže koji ustvari predstavlja grafički prikaza tabele stanja/izlaza.



Slika 6.2. Dijagram stanja sekvencijalne mreže

c) Na osnovu tabele pobude/izlaza date u tabeli 6.3. minimizacijom pomoću Karnoovih karata dobijaju se funkcije pobude flip-flova D_1 i D_0 .

Tabela 6.4. Tabele pobude ulaza flip-flova

		Q_1Q_0			
		00	01	11	10
UI	D_1	0	0	1	0
		0	1	1	1

		Q_1Q_0			
		00	01	11	10
UI	D_0	0	0	0	0
		1	1	1	1

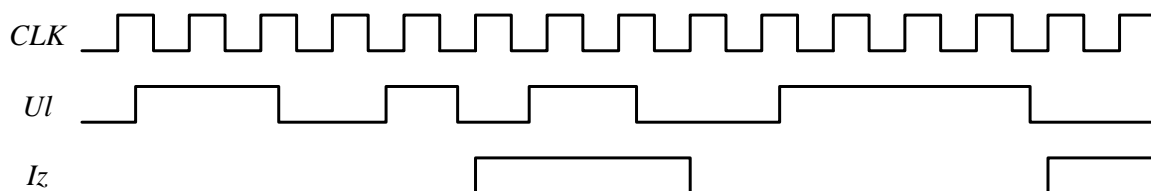
Na osnovu tabele 6.4. dobijaju se funkcije pobude ulaza flip-flova:

$$D_1 = UI(Q_1 + Q_0) + Q_1Q_0$$

$$D_0 = UI$$

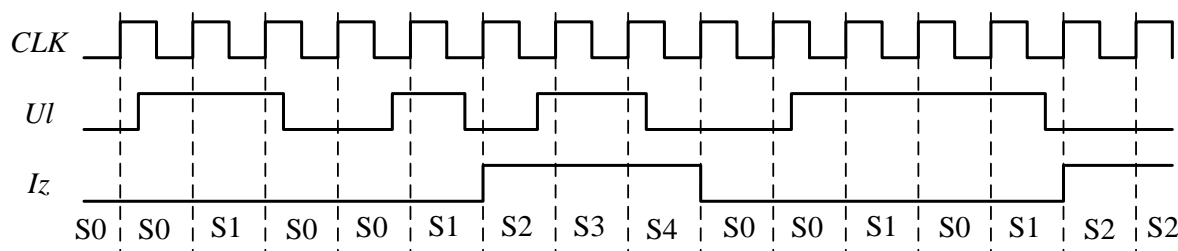
7 Projektovati sinhronu sekvencijalnu mrežu koja je zadata vremenskim dijagramima prikazanim na slici. Izlazni signal se setuje (postavlja na logičku 1) ako je aktivnom vrednošću ulaznog signala obuhvaćen neparni broj uzlaznih ivica signala takta a resetuje ako je obuhvaćen paran broj uzlaznih ivica. Odrediti:

- Tabelu stanja/izlaza, tabelu prelaza/izlaza, i tabelu pobude/izlaza ako su za realizaciju na raspolaganju ivični D flip-flovi i proizvoljan broj osnovnih logičkih kola.
- Nacrtati dijagram stanja ove sekvencijalne mreže
- Realizovati mrežu korišćenjem ivičnih D flip-flova



Rešenje:

a) Na osnovu vremenskih dijagrama i opisa rada mreže mogu se identifikovati različita stanja u kojima se sistem može naći, što je prikazano na slici 7.1.



Slika 7.1. Vremenski dijagrami i stanja sekvencijalne mreže

U početnom trenutku mreža je u stanju S0 i na izlazu je logička 0. Sve dok je ulazni signal na nivou logičke 0 sekvencijalna mreža ostaje u stanju S0. Ako se, dok je u stanju S0, na sledeću uzlaznu ivicu pojavi logička 1 ulaznog signala, mreža prelazi u stanje S1. Stanje S1 označava da je trenutnim impulsom ulaznog signala obuhvaćen neparan broj uzlaznih ivica takta. Ako se, dok je mreža u stanju S1, na sledeću uzlaznu ivicu takta pojavi logička 0 ulaznog signala, znači da je pređašnji impuls ulaznog signala završen i da je njime bio obuhvaćen neparan broj uzlaznih ivica pa je prema tome potrebno setovati izlazni signal tj. mreža prelazi u stanje S2. Mreža ostaje u stanju S2 sve dok se ponovo ne registruje logička 1 na ulazu. Tada mreža prelazi u stanje S3 koje označava da je trenutno u toku impuls ulaznog signala i da je obuhvaćen neparan broj uzlaznih ivica. Iz stanja S3 mreža može preći u stanje S4 koje označava da je impuls ulaznog signala još uvek u toku i da je do sada obuhvaćen paran broj uzlaznih ivica taktnog signala ili u stanje S2 u slučaju logičke nule na ulazu što označava da je prethodnim impulsom ulaznog signala obuhvaćen neparan broj uzlaznih ivica. Ako se impuls ulaznog signala nastavi dok je mreža u stanju S4 na sledeću uzlaznu ivicu prelazi se u stanje S3 pošto se menja parnost obuhvaćenih ivica dok u slučaju da se impuls završi dok je mreža u stanju S4 znači da je prethodnim impulsom ulaznog signala obuhvaćen paran broj uzlaznih ivica takta i izlaz je potrebno resetovati odnosno mreža prelazi u stanje S0.

Određena stanja i prelazi mreže su sumirani u tabeli stanja/izlaza. Tabela stanja/izlaza je prikazana u tabeli 7.1.

Tabela 7.1. Tabela stanja/izlaza

		Ul		Iz
		0	1	
S	S0	S0	S1	0
	S1	S2	S0	0
	S2	S2	S3	1
	S3	S2	S4	1
	S4	S0	S3	1

Dalje je potrebno kodovati svako stanje odgovarajućim promenljivama stanja. Kako projektovana sekvencijalna mreža prolazi kroz 5 stanja to su potrebne 3 promenljive stanja (3 flip-flopa). Vodeći se principima kodovanja stanja opisanim u prethodnom zadatku dobija se tabela prelaza/izlaza prikazana u tabeli 7.2.

Tabela 7.2. Tabela prelaza/izlaza

		Ul		Iz
		0	1	
$Q_2Q_1Q_0$	000	000	001	0
	001	011	000	0
	011	011	010	1
	010	011	110	1
	110	000	010	1
	100	bbb	bbb	b
	101	bbb	bbb	b
	111	bbb	bbb	b

$Q_2^+Q_1^+Q_0^+$

Tabela 7.3. Tabela pobude/izlaza

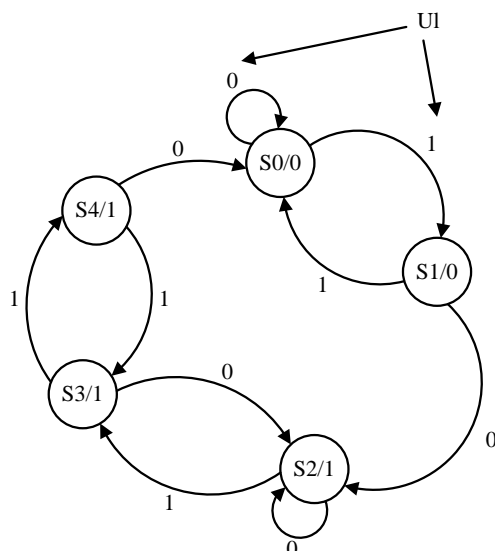
		Ul		Iz
		0	1	
$Q_2Q_1Q_0$	000	000	001	0
	001	011	000	0
	011	011	010	1
	010	011	110	1
	110	000	010	1
	100	bbb	bbb	b
	101	bbb	bbb	b
	111	bbb	bbb	b

$D_2D_1D_0$

Kako projektovana mašina stanja u regularnom radu prolazi kroz 5 od ukupno 8 stanja potrebno je voditi računa šta se dešava ako se nađe u nekom od 3 zabranjena stanja. Jedna mogućnost je da se mašina stanja projektuje tako da se iz zabranjenih stanja prelazi u neko regularno stanje, čime se dobija mašina maksimalne pouzdanosti. Druga mogućnost, kojom se dobija mreža minimalne kompleksnosti je da se prelazi iz zabranjenih stanja ostave nedefinisanim i koriste u minimizaciji. Po završenoj minimizaciji je potrebno proveriti šta se dešava sa ovim prelazima i da li mašina izlazi iz zabranjenih stanja. Ako se mašina u nekom slučaju vrti unutar zabranjenih stanja potrebno je izvršiti korekciju pobudnih funkcija tako da se ovo izbegne. Kako se u ovom zadatku zahteva da mreža bude minimalne kompleksnosti korišćen je drugi pristup.

Koristeći tabelu 7.2. i karakterističnu jednačinu D flip-flopa $Q^+ = D$ dolazi se do table pobude/izlaza prikazane u tabeli 7.3.

b) Na osnovu table 7.1. dobija se dijagram stanja projektovane sekvencijalne mreže koji ustvari predstavlja grafički prikaza table stanja/izlaza.



Slika 7.2. Dijagram stanja sekvencijalne mreže

c) Na osnovu tabele pobude/izlaza date u tabeli 7.3. minimizacijom pomoću Karnoovih karata dobijaju se funkcije pobude flip-flopova D_2 , D_1 i D_0 .

Tabela 7.4. Tabele pobude ulaza flip-flopova

		Q_1Q_0			
		00	01	11	10
UIQ_2	D_2	0	0	0	0
	01	<i>b</i>	<i>b</i>	<i>b</i>	0
	11	<i>b</i>	<i>b</i>	<i>b</i>	0
	10	0	0	0	1
	00	0	0	0	0

		Q_1Q_0			
		00	01	11	10
UIQ_2	D_1	0	1	1	1
	01	<i>b</i>	<i>b</i>	<i>b</i>	0
	11	<i>b</i>	<i>b</i>	<i>b</i>	1
	10	0	0	1	1
	00	0	0	0	0

		Q_1Q_0			
		00	01	11	10
UIQ_2	D_0	0	1	1	1
	01	<i>b</i>	<i>b</i>	<i>b</i>	0
	11	<i>b</i>	<i>b</i>	<i>b</i>	0
	10	1	0	0	0
	00	0	0	0	0

Na osnovu tabele 7.4. dobijaju se funkcije pobude ulaza flip-flopova:

$$D_2 = UI\overline{Q_2}Q_1\overline{Q_0} \quad D_1 = \overline{UI}Q_0 + \overline{Q_2}Q_1 + UIQ_1$$

$$D_0 = \overline{UI}Q_0 + \overline{UI}\overline{Q_2}Q_1 + UI\overline{Q_1}\overline{Q_0}$$

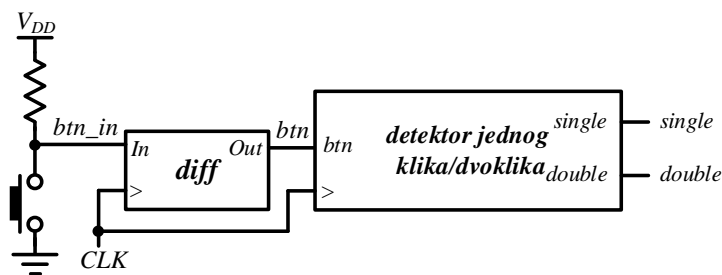
Ako se iz nekog razloga (npr. usled elektromagnetske smetnje) sistem nađe u nedozvoljenom stanju (stanja 100, 101, 111), neophodno je proveriti da li mašina izlazi iz tih zabranjenih stanja. U tabeli 7.5 dat je prikaz prelaza iz zabranjenih stanja u neko sledeće stanje na osnovu funkcija pobude dobijenih za ulaze flip-flopova D_2 , D_1 i D_0 . U tabeli se vidi da mašina uvek prelazi u neko od dozvoljenih stanja i da se ne vrti u zabranjenim stanjima, pa stoga nije neophodno modifikovati funkcije pobude.

Tabela 7.5. Tabela prelaza/izlaza zabranjenih stanja

		UI		
		0	1	I_z
$Q_2Q_1Q_0$	100	000	000	0
	101	011	000	0
	111	011	010	1

$Q_2^+Q_1^+Q_0^+$

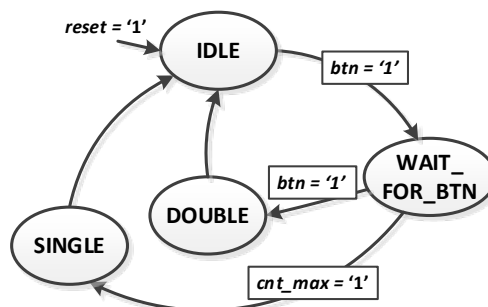
8 Projektovati sinhronu sekvencijalnu mrežu koja detektuje jedan pritisak tastera i brzi dvoklik na taster. Ako je detektovan samo jedan pritisak na taster, mreža treba da postavi signal *single* na logičku jedinicu u trajanju samo jedne periode takta. Ako je detektovan brzi dvoklik, mreža treba da postavi signal *double* na logičku jedinicu u trajanju samo jedne periode takta. Šema sistema je prikazana na slici. Najpre je potrebno projektovati mašinu stanja koja detektuje pritisak tastera, tj. silaznu ivicu signala *btn_in*, a koja na svom izlazu generiše signal *btn* aktivan na logičkoj jedinici trajanja tačno jedne periode signala takta. Za projektovanje detektora jednog klika odnosno detektora dvoklika smatrati da su na raspolaganju potpuni 6-bitni sinhron brojač sa sinhronim ulazom za *RESET* i signalom dozvole *EN*, D flip-flopovi i potreban broj logičkih kola. Dvoklik je detektovan ako se dva pritiska tastera pojave u intervalu koji je manji od periode ili jednak periodu 6-bitnog brojača. Ako je vreme između pojave dve logičke jedinice signala *btn* veće od navedenog perioda potrebno je generisati signal *single* za svaki od pojedinačnih pritisaka tastera.



Rešenje:

Projektovanje mašine stanja za diferenciranje silazne ivice se ostavlja studentima za vežbu. Mašina stanja ima samo 3 stanja i lako se realizuje uz pomoć 2 flip flopa.

Detektor dvoklika bi trebalo da ima neko stanje u kome se ništa ne dešava i u kome se čeka prvi pritisak tastera, tj. prva pojava logičke jedinice na signalu *btn*. To stanje ćemo zvati *IDLE*. Nakon što se pojavi prvi pritisak tastera potrebno je startovati brojač i preći u stanje u kome se čeka naredni pritisak tastera. Ovo stanje ćemo nazvati *WAIT_FOR_BTN*. Ako se pritisak tastera pojavi pre nego što brojač dostigne vrednost $2^6 - 1$ prelazi se u stanje *DOUBLE* u kome se ostaje samo jedan period signala takta nakon čega se mašina stanja vraća u *IDLE*. U stanju *DOUBLE* treba generisati signal *double*. Ako se detektuje maksimalna vrednost na izlazu brojača, a signal *btn* je i dalje na logičkoj nuli prelazi se u stanje *SINGLE* u kome se ostaje samo jedan period signala takta nakon čega se mašina stanja vraća u *IDLE*. U stanju *SINGLE* treba generisati signal *single*. Ulaz u ovu mašinu stanja je očigledno signal *btn*, ali i signal koji govori da li je vrednost brojača $2^6 - 1$. Taj signal ćemo nazvati *cnt_max* i lako ga generisati kao izlaz 6-ulaznog I kola na čije ulaze se dovode svi biti izlaza brojača *B[5:0]*. Dijagram stanja je prikazan na slici 8.1.



Slika 8.1 – Dijagram stanja detektora jednog illi duplog pritiska na taster

Primititi da je mreža u stanju *IDLE* ako je signal *reset* na logičkoj jedinici. U rešenju neće biti eksplicitno naglašeno, ali se ta funkcionalnost može realizovati bilo kao asinhroni bilo kao sinhroni reset kako je već rađeno u prethodnim zadacima.

Brojač treba da broji isključivo u stanju *WAIT_FOR_BTN*, pa se signal dozvole generiše jedino u tom stanju, dok je u svim ostalima na logičkoj nuli. Zvaćemo ga *cnt_en*. Takođe, u *IDLE* stanju je neophodno resetovati brojač ako je on stigao do neke vrednosti koja nije 0, pa se signal reseta brojača generiše jedino u *IDLE* stanju i zvaćemo ga *cnt_reset*. Izlazni signal *single* je aktivan jedino u stanju *SINGLE*, a izlazni signal *double* je aktivan jedino u stanju *DOUBLE*.

Na osnovu svega prethodnog, možemo dobiti tabelu stanja/izlaza koja je prikazana u tabeli 8.1. Primititi da se u tabeli nalaze i kontrolni signali za brojač koji meri vreme između dva pritiska tastera.

Tabela 8.1 – Tabela stanja/izlaza glavne mašine stanja

Trenutno stanje	<i>btn, cnt_max</i>				<i>cnt_en</i>	<i>cnt_reset</i>	<i>single</i>	<i>double</i>
	00	01	11	10				
<i>IDLE</i>	<i>IDLE</i>	<i>IDLE</i>	<i>WAIT_FOR_BTN</i>	<i>WAIT_FOR_BTN</i>	0	1	0	0
<i>WAIT_FOR_BTN</i>	<i>WAIT_FOR_BTN</i>	<i>SINGLE</i>	<i>DOUBLE</i>	<i>DOUBLE</i>	1	0	0	0
<i>SINGLE</i>	<i>IDLE</i>	<i>IDLE</i>	<i>IDLE</i>	<i>IDLE</i>	0	0	1	0
<i>DOUBLE</i>	<i>IDLE</i>	<i>IDLE</i>	<i>IDLE</i>	<i>IDLE</i>	0	0	0	1

Nakon kodovanja stanja dobija se tabela prelaza/izlaza koja je ista kao i tabela pobude/izlaza jer su na raspolaganju D flip flopovi (tabela 8.2). Iz ove tabele se dobijaju Karnoove mape za logičke funkcije ulaza flip flopova prikazane u tabeli 8.3.

Tabela 8.2 – Tabela prelaza/izlaza glavne mašine stanja

Q_1Q_0	<i>btn, cnt_max</i>				<i>cnt_en</i>	<i>cnt_reset</i>	<i>single</i>	<i>double</i>
	00	01	11	10				
00	00	00	01	01	0	1	0	0
01	01	11	10	10	1	0	0	0
11	00	00	00	00	0	0	1	0
10	00	00	00	00	0	0	0	1

$Q_1+Q_0+ = D_1D_0$

Tabela 8.3 – Tabele pobude/ulaza

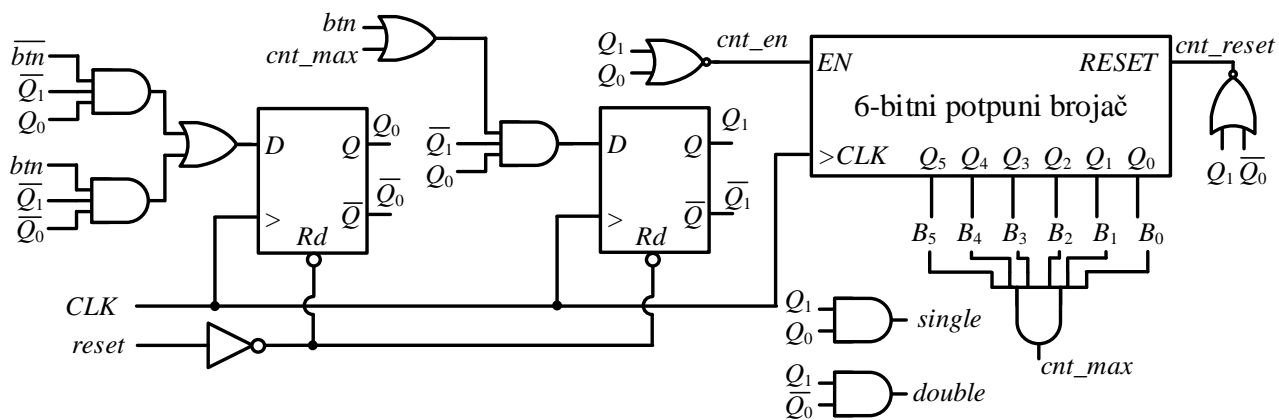
D_1 :

Q_1Q_0	<i>btn, cnt_max</i>			
	00	01	11	10
00	0	0	0	0
01	0	1	1	1
11	0	0	0	0
10	0	0	0	0

D_0 :

Q_1Q_0	<i>btn, cnt_max</i>			
	00	01	11	10
00	0	0	1	1
01	1	1	0	0
11	0	0	0	0
10	0	0	0	0

Kompletna realizacija sekvencijalne mreže iz ovog zadatka, ne računajući diferencijator silazne ivice, prikazana je na slici 8.2.



Slika 8.2 – Realizacija detektora koji detektuje jedan odnosno dupli pritisak tastera

9 BONUS ZADATAK ZA ZAINTERESOVANE

Projektovati sinhronu sekvencijalnu mrežu koja na ulazu uzima dva N -tobitna broja, $A_{N-1..0}$ i $B_{N-1..0}$, serijski, bit po bit pri čemu prvo dolazi bit najveće težine, a na izlazu generiše veći od ova dva broja $C_{N-1..0} = \max\{A_{N-1..0}, B_{N-1..0}\}$, takođe serijski, bit po bit.