

1. Projektovati potpuni binarni 3-bitni asinhroni brojač koristeći ivične T flip-flopove sa EN ulazom i asinhronim ulazima za SET i RESET aktivnim u logičkoj nuli.

a) Obezbediti da za vreme trajanja aktivne vrednosti signala RESET brojača (aktivnog u logičkoj jedinici) brojač bude u stanju nula.

b) Ako je kašnjenje flip-flopa $t_{dff} = 10$ ns, vreme postavljanja ulaza flip-flopa $t_{ds} = 4$ ns i kašnjenje kroz logička kola $t_{dlk} = 5$ ns odrediti maksimalnu učestanost rada ovog brojača i kašnjenje izlaznih signala.

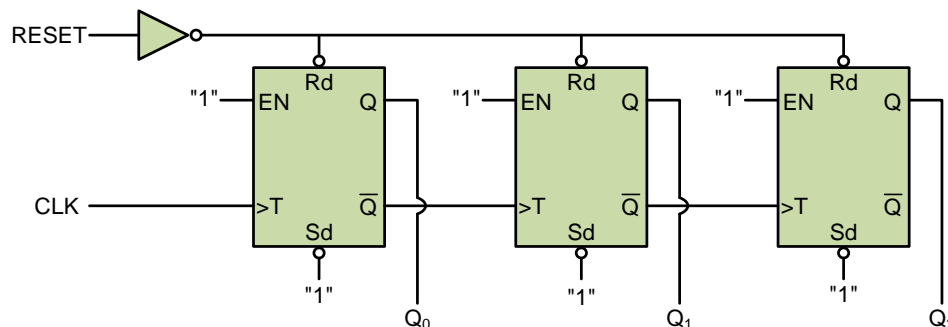
c) Projektovaiti kombinacionu mrežu kojom se dekoduje stanje 2 ovog brojača. Uzimajući u obzir kašnjenje iz prethodne tačke nacrtati vremenske dijagrame stanja brojača i dekodovanog izlaza pri prelasku iz stanja 3 u stanje 4.

d) Da li dolazi do pojave lažnih vrednosti izlaza i ako dolazi predložiti modifikaciju tako da se one otklone.

e) Uraditi sve prethodne tačke na primeru sinhronog 3-bitnog potpunog binarnog brojača realizovanog pomoću ivičnih T flip-flopa.

Rešenje:

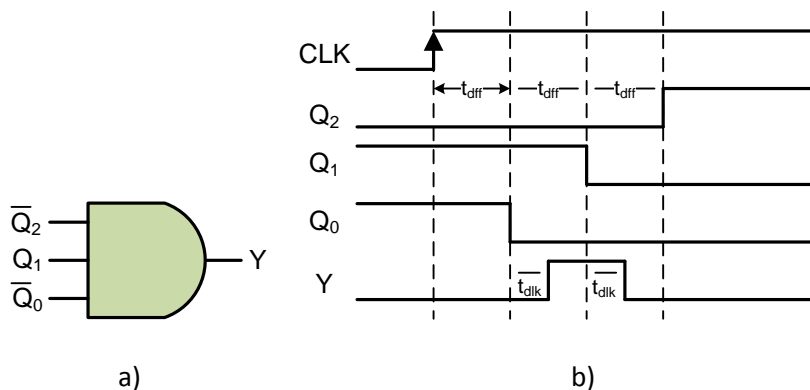
a) Izgled asinhronog brojača sa signalom RESET je prikazan na Slici 1.1.



Slika 1.1. Realizacija potpunog binarnog asinhronog brojača

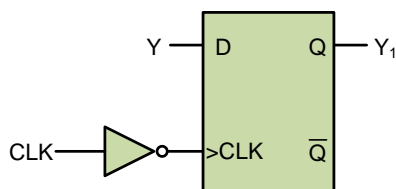
b) Da bi brojač ispravno radio potrebno je da vreme između dve uzlazne ivice signala takta bude dovoljno da se završe svi prelazni procesi. Kako se ovde radi o asinhornom brojaču potrebno je da signal propagira kroz sva tri flip-flopa. Odnosno da bi se postavio izlaz trećeg flip-flopa potrebno je da se prethodno postave izlazi prvog i drugog flip-flopa. Po postavljanju izlaza trećeg flip-flopa imamo validno stanje brojača pošto su završeni svi prelazni procesi. Tako da će kašnjenje ovog izlaza odrediti učestanost rada. Kašnjenje izlaza trećeg flip-flopa je $t_{dQ2} = 3t_{dff}$ odnosno $t_{dQ2} = 30$ ns. Maksimalna učestanost rada ovog brojača je prema tome $f_{max} < 1/t_{dQ2}$ odnosno $f_{max} < 33$ MHz. Kašnjenje izlaza ovog brojača iznosi $t_{dout} = t_{dQ2} = 30$ ns.

c) Kombinaciona mreža koja dekoduje stanje 2 brojača predstavlja trouglasto I kolo prikazana na Slici 1.2 a). Izlaz ove kombinacione mreže je jednak 1 kada je brojač u stanju 2 dok je u suprotnom jednak 0. Vremenski dijagrami stanja brojača i dekodovanog izlaza pri prelasku brojača iz stanja 3 u 4 su prikazani na Slici 1.2 b). Sa Slike 1.2 b) se jasno može uočiti trajanje prelaznih procesa u ovom brojaču računatih u prethodnoj tački.



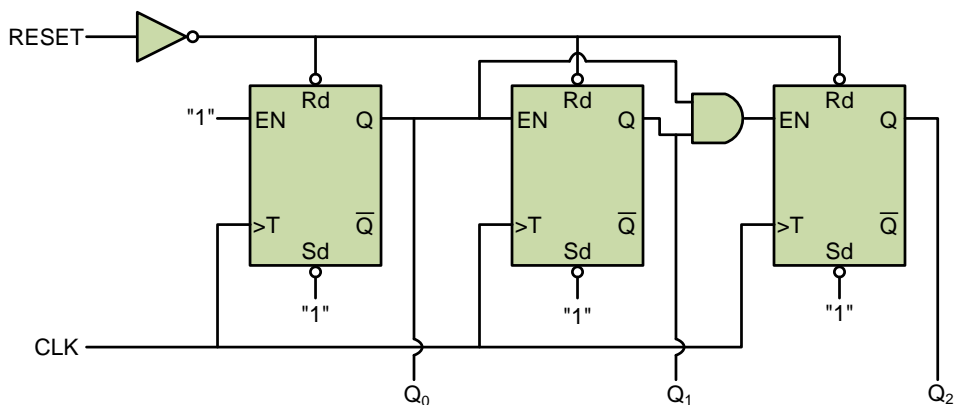
Slika 1.2. Dekodovani izlaz stanja 2 brojača

d) Na osnovu dijagrama iz prethodne tačke se može uočiti da dolazi do pojave gliča na izlazu (lažna jedinica). Ovo je posledica nejednakih kašnjenja izlaza flip-flopova zbog potrebe da signal propagira kroz ceo brojač. Jedno od mogućih rešenja je prikazano na Slici 1.3. Stanje izlaza se osvežava na silaznu ivicu signala takta. Treba voditi računa da u ovom slučaju svi prelazni procesi moraju biti završeni za vreme visokog nivoa signala takta tako da se radna učestanost brojača još više smanjuje.



Slika 1.3. Kolo kojim se otklanjaju gličevi dekodovanog izlaza brojača

e) Na Slici 1.4. je prikazan izgled sinhronog brojača



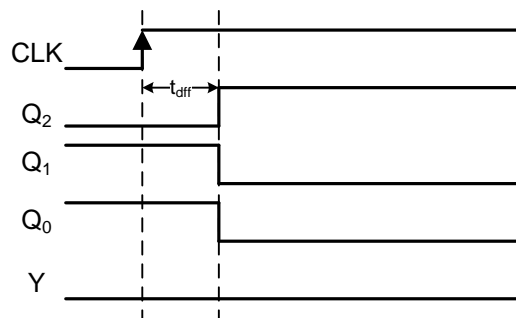
Slika 1.4. Šema sinhronog potpunog binarnog brojača

Kod sinhronog brojača izlazi svih flip-flopova imaju jednako kašnjenje tako da je kašnjenje izlaza brojača $t_{dout} = t_{dff} = 10 \text{ ns}$. Kako bi brojač ispravno radio potrebno je da po dolasku uzlazne ivice signala takta brojač ima validne vrednosti na svojim ulazima. Stoga je potrebno pronaći najdužu putanju od signala takta do ulaza nekog od flip-flopva brojača. Sa Slike 1.4. se vidi da najveće kašnjenje ulaza ima treći flip-flop pošto je potrebno da signal propagira i kroz 1 kolo. Kašnjenje ulaza ovog flip-flopa je $t_{dT2} = t_{dff} + t_{dlk} + t_{ds}$ odnosno $t_{dT2} = 19 \text{ ns}$ pa je $f_{max} < 1/t_{dT2} = 52.6 \text{ MHz}$. Odavde se

vidi da su sinhroni brojači dosta brži od asinhronih, da im je kašnjenje izlaza znatno manje i da ne zavisi od broja bita brojača.

Kombinaciona mreža koja dekoduje stanje 2 ovog brojača ista je kao i kod asinhronne realizacije.

Kako se svi izlazi sinhronog brojača postavljaju sa jednakim kašnjenjem u odnosu na signal takta to ne dolazi do pojave gličeva na dekodovanom izlazu brojača. Vremenski dijagrami stanja sinhronog brojača i dekodovanog izlaza pri prelasku brojača iz stanja 3 u stanje 4 su prikazani na Slici 1.5.



Slika 1.5. Vremenski dijagrami pri prelasku sinhronog brojača iz stanja 3 u stanje

2. a) Projektovati sinhroni brojač koji broji sekvencu $2 \rightarrow 6 \rightarrow 1 \rightarrow 5 \rightarrow 2 \dots$ koristeći ivične JK flip-flopove sa asinhronim ulazima za SET i RESET aktivnim u logičkoj nuli. Obezbediti da pri aktivnoj vrednosti ulaza RESET brojača (aktivan u logičkoj jedinici) brojač bude u stanju 3, a da po deaktiviranju ovog signala na prvu sledeću uzlaznu ivicu signala takta prelazi u stanje 1. Voditi računa da kombinaciona mreža bude minimalne složenosti.

b) Modifikovati prethodno rešenje ako je potrebno da brojač izlazi iz zabranjenih stanja posle najviše jednog perioda signala takta.

Rešenje:

a) Na osnovu zadate sekvence brojača možemo zaključiti da postoje 4 različita stanja kroz koja brojač prolazi i da nam za njegovu realizaciju trebaju samo 2 flip-flopa. Međutim kako je za vreme trajanja signala RESET brojač u stanju koje se razlikuje od svih stanja u kojima se nalazi kad je u modu brojanja ovaj brojač zapravo prolazi kroz 5 stanja pa su za njegovu realizaciju potrebna 3 flip-flopa. Na osnovu ovoga je određena tabela stanja/prelaza ovog brojača prikazana u Tabeli 2.1.

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

Tabela 2.1. Tabela stanja/prelaza brojača

Kako je zahtevano da kombinatorna mreža bude u minimalnoj formi to su prelazi iz zabranjenih stanja ostavljeni nedefinirani kako bi se ostavilo više slobode pri minimizaciji funkcija pobude flip-flopova. Po određivanju funkcija pobude flip-flopova potrebno je proveriti ove prelaze kako bi se izbegla situacija u kojoj se brojač vrti između zabranjenih stanja i nikad ne izlazi iz njih.

Dalje je potrebno odrediti tabelu stanja/pobude brojača. Međutim pre ovoga je potrebno odrediti tabeli stanja/pobude JK flip-flopa. Ona se jednostavno određuje na osnovu karakteristične jednačine rada JK flip-flopa $Q^+ = J\bar{Q} + \bar{K}Q$ i prikazana je u Tabeli 2.2.

Q	Q ⁺	J	K
0	0	0	b
0	1	1	b
1	0	b	1
1	1	b	0

Tabela 2.2. Tabela stanja/pobude JK flip-flopa

Na osnovu Tabele 2.1. i Tabele 2.2., jednostavno se određuje tabela stanja/pobude brojača prikazana u Tabeli 2.3.

Q ₂	Q ₁	Q ₀	Q ₂ ⁺	Q ₁ ⁺	Q ₀ ⁺	J ₂	K ₂	J ₁	K ₁	J ₀	K ₀
0	0	0	b	b	b	b	b	b	b	b	b
0	0	1	1	0	1	1	b	0	b	b	0
0	1	0	1	1	0	1	b	b	0	0	b
0	1	1	0	0	1	0	b	b	1	b	0
1	0	0	b	b	b	b	b	b	b	b	b
1	0	1	0	1	0	b	1	1	b	b	1
1	1	0	0	0	1	b	1	b	1	1	b
1	1	1	b	b	b	b	b	b	b	b	b

Tabela 2.3. Tabela stanja/pobude brojača

		Q ₁ Q ₀				
		J ₂	00	01	11	10
Q ₂	0	b	1	0	1	
	1	b	b	b	b	

		Q ₁ Q ₀				
		K ₂	00	01	11	10
Q ₂	0	b	b	b	b	
	1	b	1	b	1	

		Q ₁ Q ₀				
		J ₁	00	01	11	10
Q ₂	0	b	0	b	b	
	1	b	1	b	b	

		Q ₁ Q ₀				
		K ₁	00	01	11	10
Q ₂	0	b	b	1	0	
	1	b	b	b	1	

		Q ₁ Q ₀				
		J ₀	00	01	11	10
Q ₂	0	b	b	b	0	
	1	b	b	b	1	

		Q ₁ Q ₀				
		K ₀	00	01	11	10
Q ₂	0	b	0	0	b	
	1	b	1	b	b	

Tabela 2.4. Tabele pobude ulaza flip-flopova brojača

Na osnovu Tabele 2.4. dobijaju se pobude ulaza flip-flopova brojača:

$$\begin{array}{lll} J_2 = \overline{Q_1} + \overline{Q_0} & J_1 = Q_2 & J_0 = Q_2 \\ K_2 = 1 & K_1 = Q_2 + Q_0 & K_0 = Q_2 \end{array}$$

Po završetku određivanja funkcija pobude flip-flopova potrebno je proveriti šta se dešava sa prelazima iz zabranjenih stanja.

Ako se brojač nađe u stanju $Q_2Q_1Q_0 = 000$ zamenom ovih vrednosti promenljivih stanja u jednačine pobude koje smo odredili ranije dobijamo:

$$J_2 = 1, K_2 = 1 \Rightarrow Q_2^+ = \overline{Q_2} = 1$$

$$J_1 = 0, K_1 = 0 \Rightarrow Q_1^+ = Q_1 = 0$$

$$J_0 = 0, K_0 = 0 \Rightarrow Q_0^+ = Q_0 = 0$$

Odavde dobijamo da iz stanja 0 brojač prelazi u stanje 4 koje je takođe zabranjeno stanje! Dalje je potrebno odrediti šta se dešava sa brojačem kada se nađe u stanju 4.

Zamenom vrednosti promenljivih stanja za $Q_2Q_1Q_0 = 100$ u jednačine pobude dobijamo:

$$J_2 = 1, K_2 = 1 \Rightarrow Q_2^+ = \overline{Q_2} = 0$$

$$J_1 = 1, K_1 = 1 \Rightarrow Q_1^+ = \overline{Q_1} = 1$$

$$J_0 = 1, K_0 = 1 \Rightarrow Q_0^+ = \overline{Q_0} = 1$$

Iz gornjih jednačina dobijamo da brojač iz stanja 4 prelazi u stanje 3 koje je regularno resetno stanje iz koga nastavlja regularno brojanje.

Potrebno je još odrediti šta se dešava kad se brojač nađe u stanju 7. Na isti način kao u prethodnim primerima dobijamo:

$$J_2 = 0, K_2 = 1 \Rightarrow Q_2^+ = 0$$

$$J_1 = 1, K_1 = 1 \Rightarrow Q_1^+ = \overline{Q_1} = 0$$

$$J_0 = 1, K_0 = 1 \Rightarrow Q_0^+ = \overline{Q_0} = 0$$

Iz stanja 7 brojač prelazi u stanje 0 koje je zabranjeno stanje. Međutim ako pogledamo gornje prelaze iz svih zabranjenih stanja se izlazi posle dovoljno taktnih intervala. Iz stanja 4 se posle samo jednog taktnog intervala prelazi u regularno stanje 3. Za izlazak iz stanja 0 je potrebno 2 taktna intervala zbog prelaza $0 \rightarrow 4 \rightarrow 3$, dok je za izlazak iz stanja 7 potrebno 3 taktna intervala $7 \rightarrow 0 \rightarrow 4 \rightarrow 3$.

Dakle u kom god stanju da se brojač nađe po uključenju napajanja posle najviše 3 taktna intervala se uspostavlja regularan režim rada.

b) Rešenje iz prethodne tačke obezbeđuje ispravan rad brojača posle najviše 3 taktna intervala od uključenja napajanja. Nekada je međutim potrebno obezbediti što ranije ispravan režim rada. Kako bi se obezbedio izlazak iz zabranjenih stanja posle najviše jednog taktnog intervala potrebno je modifikovati funkcije pobude flip-flopova. One više neće biti minimalne ali će obezbediti najbrži izlazak brojača iz zabranjenih stanja.

Jedno od mogućih rešenja predstavlja modifikovanje pobuda J_0 i K_0 . Pobudu za J_0 je potrebno modifikovati tako da se iz stanja 0 prelazi u regularno stanje 5 umesto u zabranjeno stanje 4. Dok je pobudu za K_0 potrebno modifikovati tako da se iz stanja 7 prelazi u regularno stanje 1 za razliku od zabranjenog stanja 0. Ovim modifikacijama smo se odrekli nekih neodređenih vrednosti (označene su bojom) čime se povećava kompleksnost funkcija pobude. Modifikovane funkcije pobude su prikazane u Tabeli 2.5.

		Q_1Q_0			
		00	01	11	10
Q_2	J_0	00	01	11	10
	0	1	b	b	0
1	b	b	b	1	

		Q_1Q_0			
		00	01	11	10
Q_2	K_0	00	01	11	10
	0	b	0	0	b
1	b	1	0	b	

Tabela 2.5. Modifikovane tabele pobude ulaza flip-flopova brojača

Na osnovu Tabele 2.5. dobijaju se modifikovane pobude ulaza flip-flopova brojača:

$$J_0 = Q_2 + \overline{Q_1} \qquad K_0 = Q_2 \overline{Q_1}$$

3. Projektovati sinhroni brojač koji broji sekvencu $2 \rightarrow 6 \rightarrow 1 \rightarrow 5 \rightarrow 2 \dots$ koristeći ivične T flip-flopove sa EN ulazom.

Rešenje:

Iz zadate se sekvence se uočava da brojač prolazi kroz samo 4 različita stanja tako da je za njegovo konstruisanje dovoljno koristiti 2 flip-flopa. Međutim posmatranjem vrednosti izlaza zaključujemo da je za predstavljanje izlaza brojač potrebno najmanje 3 bita (zbog brojeva 5 i 6). Kako je konstrukcija potpunog binarnog 2-bitnog brojača sa T flip-flopovima poznata zadatak se svodi na konstrukciju kombinacione mreže koja će preslikavati stanja brojača u odgovarajuće vrednosti izlaza. Izalze mreže je moguće proizvoljno povezati sa stanjima u cilju da se dobija minimalna kombinaciona mreža. Jedino treba voditi računa o tome da se održi poredak kako bi brojač prolazio kroz zadatu sekvencu. U našem slučaju je usvojeno da stanju 1 brojača odgovara izlaz 1. Na osnovu toga preslikavanje ostalih stanja u izlaze je jednoznačno određeno. Tabela stanja/izlaza je prikazana u Tabeli 3.1.

Q ₁	Q ₀	Y ₂	Y ₁	Y ₀
0	0	1	1	0
0	1	0	0	1
1	0	1	0	1
1	1	0	1	0

Tabela 3.1. Tabela stanja/izlaza brojača

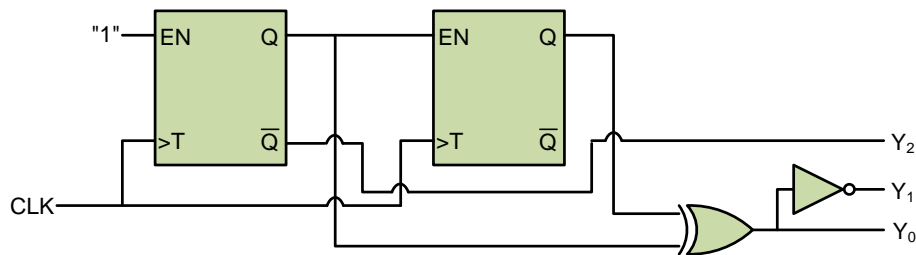
Iz Tabele 3.1. se vrlo jednostavno određuju funkcije kombinacione mreže izlaza:

$$Y_2 = \overline{Q_0}$$

$$Y_1 = \overline{Q_0}Q_1 + Q_0Q_1 = \overline{Q_1} \oplus \overline{Q_0}$$

$$Y_0 = \overline{Q_0}Q_1 + Q_0\overline{Q_1} = Q_1 \oplus Q_0$$

Realizacija traženog brojača je prikazana na Slici 3.1.



Slika 3.1. Realizacija projektovanog brojača

4. Projektovati potpuni binarni 4-bitni sinhroni brojač koristeći ivične JK flip-flopove. Ako je potrebno da:

- Brojač broji unapred.
- Brojač broji unazad.
- Brojač broji unapred ili unazad u zavisnosti od vrednosti ulaznog signala SMER (za SMER=1 broji unapred)

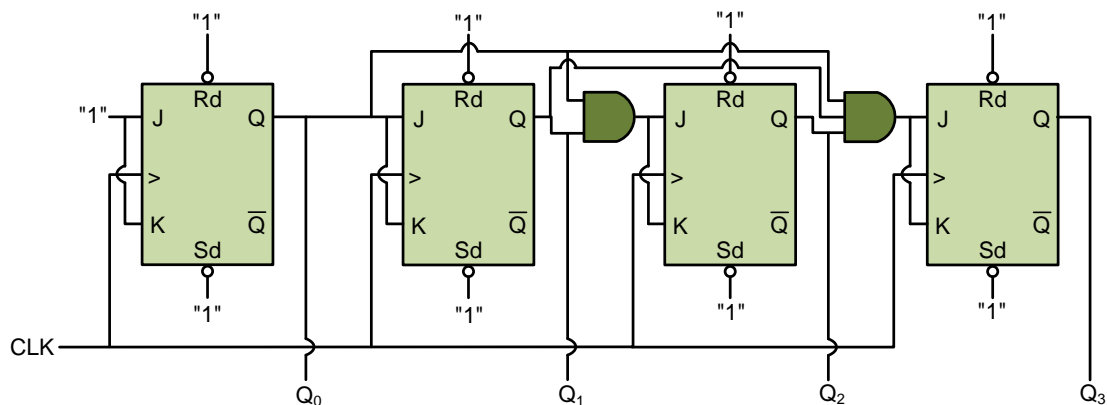
Za sve tri realizacije:

- Ako je kašnjenje flip-flova $t_{dff} = 10$ ns, kašnjenje logičkih kola $t_{dlk} = 5$ ns i vreme postavljanja flip-flova $t_{ds} = 4$ ns izračunati maksimalnu učestanost rada.
- Ako flip-flopovi poseduju ulaze sa asinhroni set i reset, Sd i Rd, aktivne u logičkoj nuli, obezbediti mogućnost asinhronog paralelnog upisa. Upis se kontroliše ulaznim signalom LOAD aktivnim u logičkoj jedinici.
- Obezbediti mogućnost sinhronog paralelnog upisa. Upis se kontroliše ulaznim signalom LOAD aktivnim u logičkoj jedinici.

g) Projektovati brojač ako su pored flip-flopa na raspolaganju samo dvoulazna logička kola.

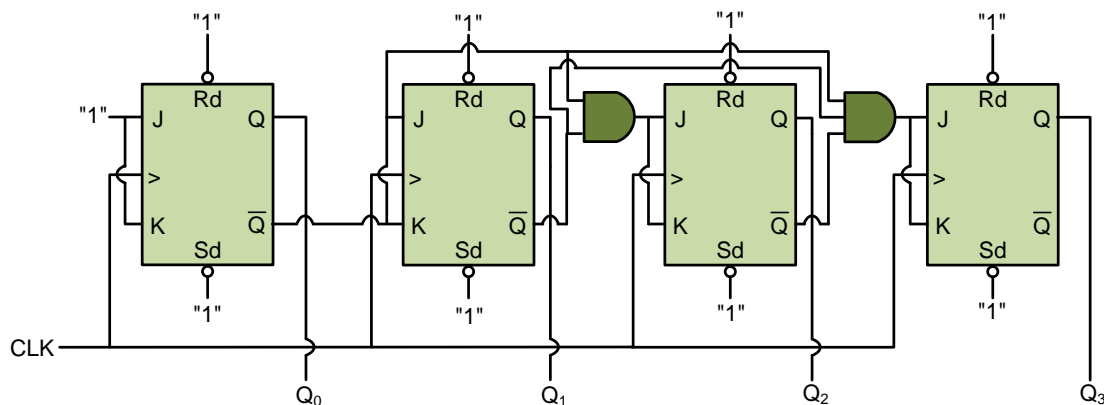
Rešenje:

a) Najpre kako je brojač sinhron to znači da se do svih flip-flopa dovodi isti signal takta. Dalje analizom sekvence brojanja zaključujemo da bit najmanje težine menja stanje sa svakim taktnim intervalom. Ovo se može postići tako što ulaze J i K flip-flopa čiji izlaz predstavlja bit najmanje težine vežemo na logičku jedinicu. Naredni bit u nizu menja stanje na uzlaznu ivicu takta jedino ako je prethodni bit jednak logičkoj jedinici. Dakle potrebno je izlaz Q_0 flip-flopa koji predstavlja bit najmanje težine vezati na ulaze J i K flip-flopa koji generiše bit Q_1 . Na isti način posmatranjem sekvence brojanja dobijamo da su ulazne funkcije preostalih flip-flopa $J_2=K_2=Q_1Q_0$ i $J_3=K_3=Q_2Q_1Q_0$. Realizacija ovog brojača je prikazana na Slici 4.1.



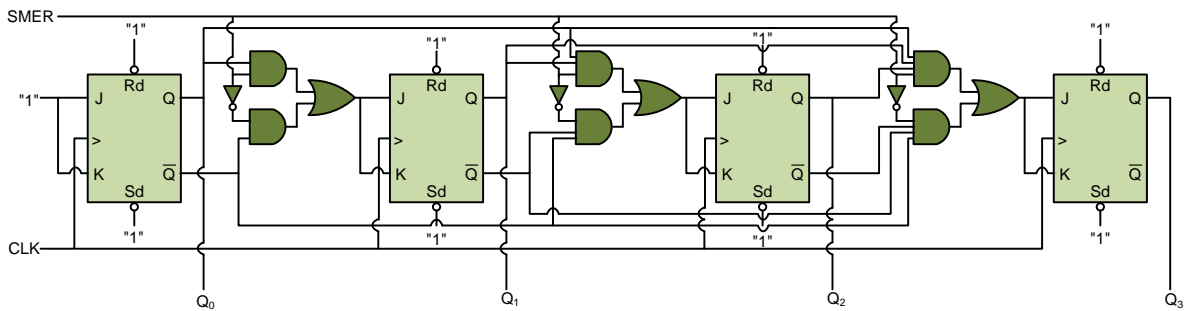
Slika 4.1. Potpuni binarni sinhroni 4-bitni brojač unapred

b) Posmatranjem sekvence brojanja i istim redosledom zaključivanja kao u prethodnoj tački dolazimo do realizacije brojača unazad. Sada se stanje nekog bita menja jedino u slučaju da su svi biti manjih težina jednaki logičkoj nuli. Realizacija brojača unazad je prikazana na Slici 4.2.



Slika 4.2. Potpuni binarni sinhroni 4-bitni brojač unazad

c) Propuštanjem signala sa izlaza Q u slučaju da je $SMER = 1$ odnosno sa \bar{Q} u slučaju da je $SMER = 0$ dobijamo realizaciju obostranog brojača prikazanu na Slici 4.3.



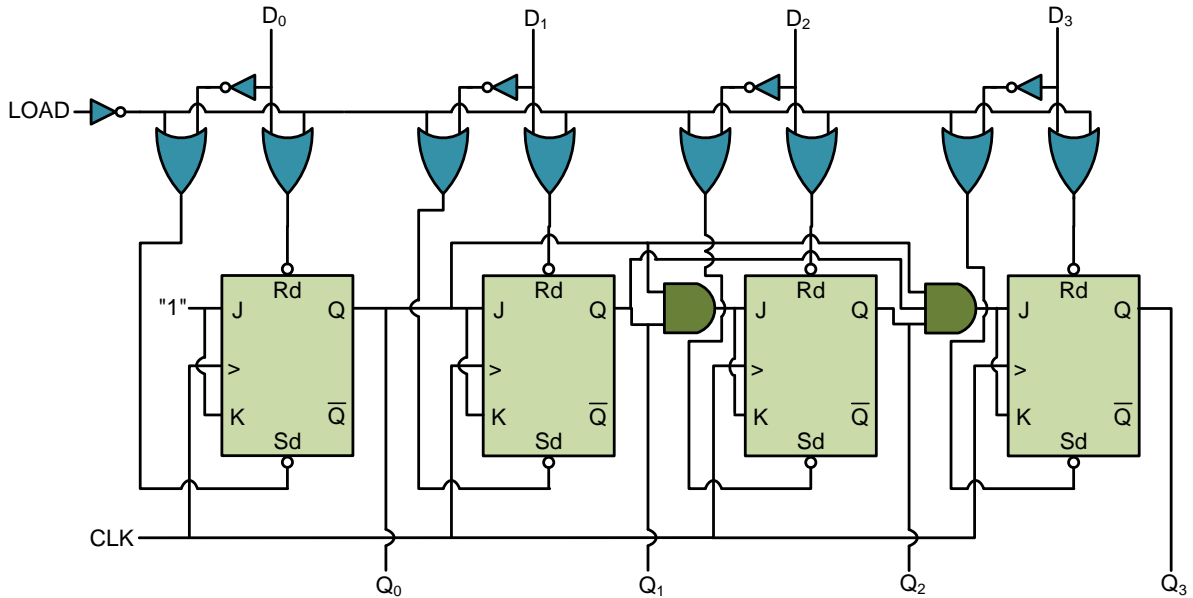
Slika 4.3. Potpuni binarni sinhroni obostrani 4-bitni brojač

d) Da bi se odredila maksimalna učestanost rada potrebno je izračunati maksimalno vreme koje protekne od dolaska uzlazne ivice takta do postavljanja stabilnih vrednosti na ulaze svih flip-flopova. Posmatranjem realizacije brojača pod a) zaključujemo da od bilo kog izlaza flip flopa do ulaza nekog drugog flip-flopa postoji maksimalno jedno logičko kolo. Na osnovu toga minimalno vreme koje je potrebno sačekati do sledeće uzlazne ivice takta je $T = t_{dff} + t_{dlk} + t_{ds}$. Dakle od pojave uzlazne ivice takta prvo se za vreme t_{dff} postavi odgovarajuća vrednost na izlaz flip-flopa, zatim ta nova vrednost propagira u ovom slučaju kroz jedno logičko kolo za vreme t_{dlk} i potrebno je da bude stabilna na ulazima flip-flopa najmanje t_{ds} da bi prelaz bio ispravan. Na osnovu gore navedenog dobijamo da je maksimalna učestanost rada ovog brojača $f_{max} < \frac{1}{T} = 52.6MHz$.

Realizacija pod b) ima istu maksimalnu učestanost rada kao i realizacija pod a)

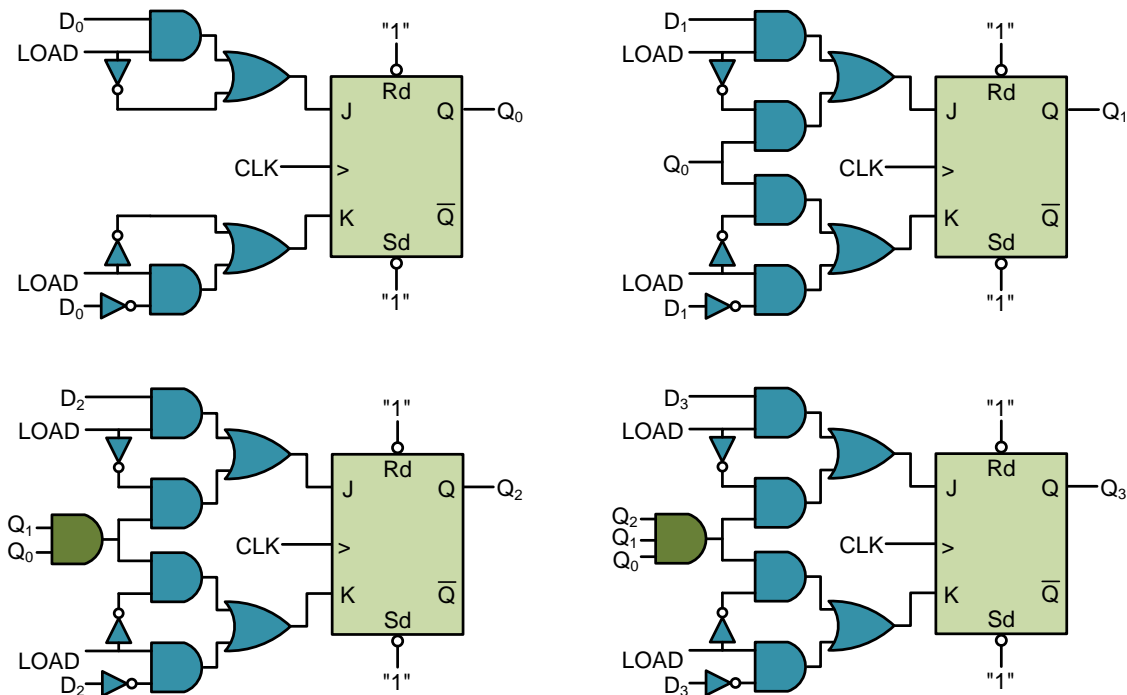
Za realizaciju pod c) zbog dodatih logičkih kola koja omogućavaju izbor smera brojanja povećava se kašnjenje po kritičnoj putanji, tako da sada iznosi $T = t_{dff} + 2t_{dlk} + t_{ds}$. Odnosno maksimalna učestanost ove realizacije je $f_{max} = 41.67MHz$.

e) Potrebno je da za vreme neaktivnog signala $LOAD=0$ na Sd i Rd ulaze flip-flopova dovesti logičku jedinicu čime se omogućava normalan rad brojača, dok je za vreme aktivne vrednosti signala $LOAD=1$ na ulaz Rd potrebno dovesti vrednost sa paralelnog ulaza brojača a na Sd komplementarnu vrednost čime se obezbeđuje asinhron upis odgovarajućeg podatka. Logička šema brojača sa mogućnošću paralelnog asinhronog upisa je prikazana na Slici 4.4.



Slika 4.4. Potpuni binarni sinhroni 4-bitni brojač sa paralelnim asinhronim upisom

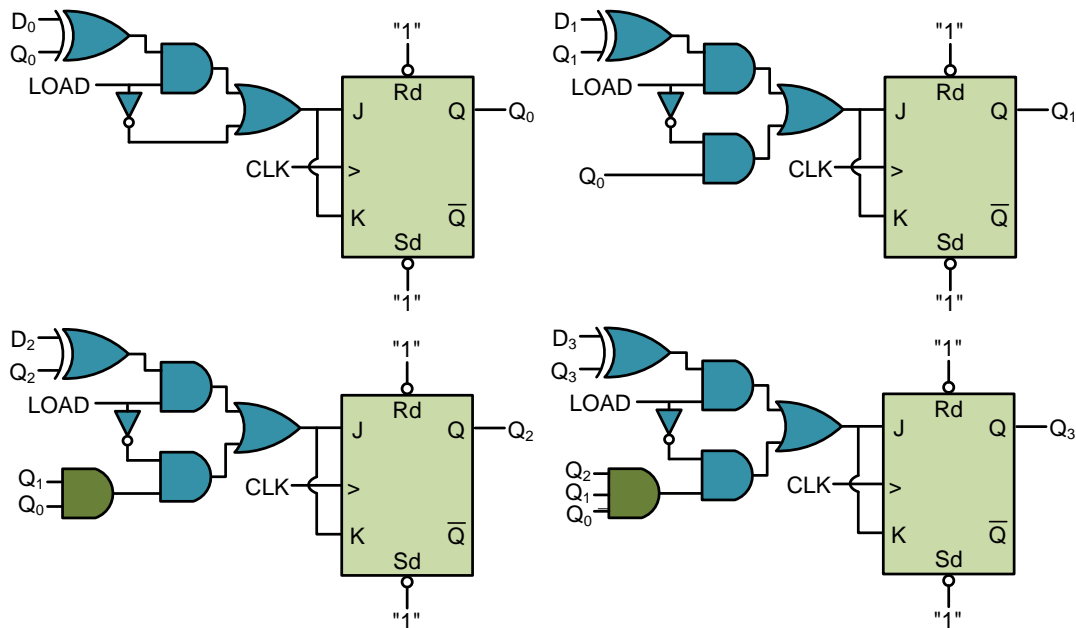
f) Kod sinhronog upisa potrebno je obezbediti da se ulazni podatak pojavi na izalzima brojača na prvu sledeću uzlaznu ivicu signala takta u odnosu na njegovo pojavljivanje na ulazima brojača. Iz ovog razloga korišćenje asinhronih ulaza Sd i Rd ne dolazi u obzir. Kako je brojač realizovan pomoću JK flip flopova to je za vreme upisa (LOAD=1) na ulaz J potrebno postaviti pravu a na ulaz K komplementiranu vrednost ulaznog bita D, dok je za vreme brojanja (LOAD=0) na ulaze brojača potrebno propustiti odgovarajuće signale koji obezbeđuju brojanje kao što je to učinjeno u prethodnim tačkama. Realizacija brojača sa sinhronim upisom prikazana je na Slici 4.4.



Slika 4.4. Potpuni binarni sinhroni 4-bitni brojač sa paralelnim sinhronim upisom

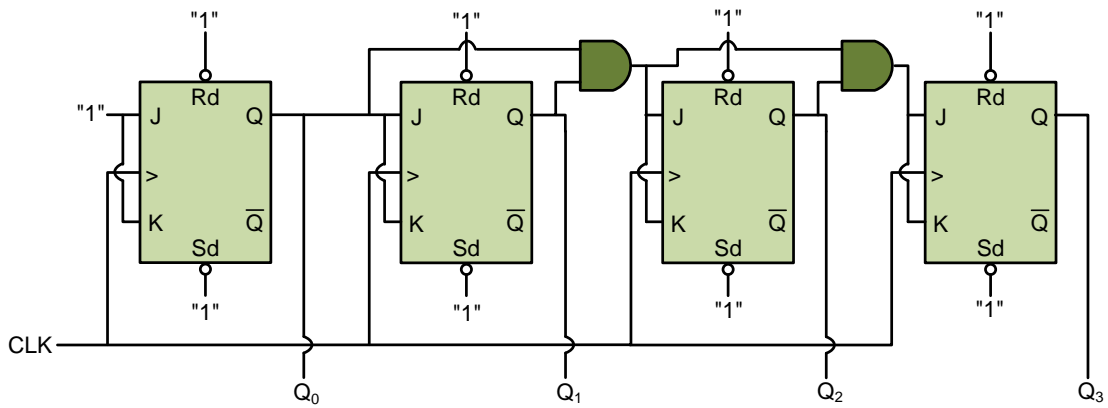
Primerimo da je zbog različite vrednosti koja se dovodi na J i K ulaze u sličaju upisa potrebno generisati posebno logičke funkcije za pobudu ovih ulaza. Ovo se može izbeći ako se koristi sledeća

logika pri upisu. U slučaju da je ulazni bit D isti kao trenutni izlaz flip flopa Q nije potrebno nista uraditi odnosno na ulazima J i K je potrebno postaviti 0. U slučaju da se ulaz D i izlaz Q razlikuju potrebno je promeniti stanje brojača odnosno na ulazima J i K je potrebno postaviti 1. Na ovaj način je postignuto da su u slučaju bilo kog ulaznog podatka D pobude na ulazima J i K identične. Realizacija brojača sa sinhronim upisom realizovana korišćenjem prethodno opisane ideje je prikazana na Slici 4.5. Sa ove slike se vidi da je na ovaj način postignuta značajna ušteda u broju logičkih kola korišćenih pri realizaciji.



Slika 4.5. Potpuni binarni sinhroni 4-bitni brojač sa paralelnim sinhronim upisom, unapređena verzija

g) Ako su na raspolaganju samo dvoulazna logička kola onda moramo pribeći realizaciji prikazanoj na Slici 4.6. Ovakav brojač se naziva i brojač sa serijskim prenosom, dok se realizacija pod a) naziva brojač sa paralelnim prenosom. U ovoj realizaciji kritična putanja je od izlaza Q_0 do ulaza J_3 i K_3 i na njoj se nalaze dva logička kola tako da je sada minimalno vreme koje je potrebno sačekati da bi se stabilizovali ulazi $T = t_{dff} + 2t_{dlk} + t_{ds}$, čime je maksimalna učestanost rada smanjena. Potrebno je primetiti da se sa povećanjem broja bita brojača povećava i kašnjenje odnosno smanjuje maksimalna učestanost rada što nije slučaj kod brojača sa paralelnim prenosom gde je maksimalna učestanost konstantna. Međutim kompleksnost realizacije brojača sa paralelnim prenosom značajno raste sa povećanjem broja bita.



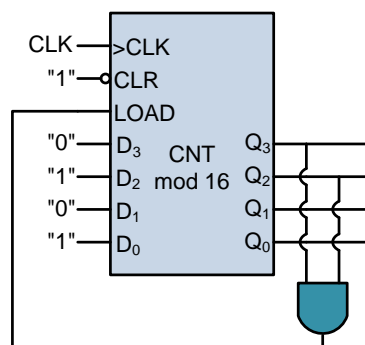
Slika 4.6. Potpuni binarni sinhroni 4-bitni brojač unapred sa dvoulaznim logičkim kolima

5. Ako je na raspolaganju potpuni binarni 4-bitni sinhroni brojač sa mogućnošću asinhronog reseta i sinhronog paralelnog upisa realizovati:

- Brojač koji broji sekvencu između brojeva 5 i 12 (uključujući i njih).
- Brojač koji broji po modulu 10.
- Nacrtati vremenske dijagrame za realizacije pod b).
- Dekodovati stanja datog brojača ako su na raspolaganju dekoderi 3/8.

Rešenje:

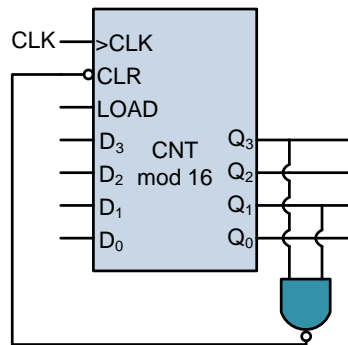
a) Da bi brojač brojao sekvencu između 5 i 12 potrebno je obezbediti da kada stigne do broja 12 brojač pređe u stanje 5. Ovo se može postići dekodovanjem stanja 12 brojača i ovom vrednošću aktiviranjem ulaza LOAD (prethodno je na ulaze za paralelni upis potrebno postaviti odgovarajući podatak u ovom slučaju 5). Tražena realizacija je prikazana na Slici 5.1.



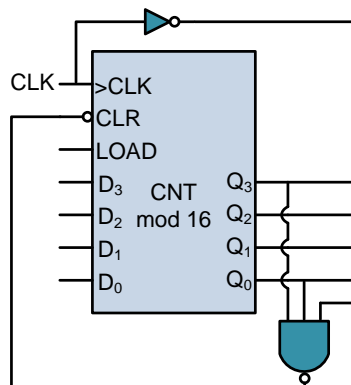
Slika 5.1. Brojač koji broji sekvencu između 5 i 12

b) Da bi realizovali brojač po modulu 10 potrebno je obezbediti da posle stanja 9 pređe u stanje 0. Kako je na raspolaganju asinhroni RESET ne dolazi u obzir da po detekciji stanja 9 aktiviramo RESET brojača pošto bi onda stanje 9 trajalo veoma kratko. Jedno rešenje je da se detektuje stanje 10 i da se njime aktivira RESET brojača. Ova realizacija je prikazana na Slici 5.2. Mana ove realizacije je što se

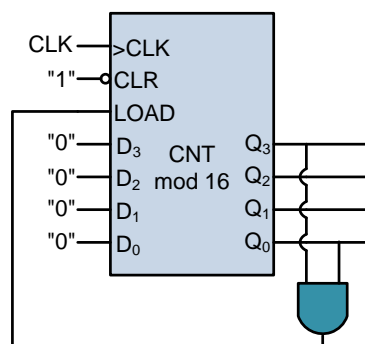
na izlazima brojača kratkotrajno pojavljuje nedozvoljena vrednost dok je on u stanju 10. Ako na izlazima ne sme da se pojavi makar i na kratko nedozvoljena vrednost signala onda je potrebno detektovati vrednost 9 na izlazu brojača. Međutim potrebno je nekako zakasniti signal RESETa što se rešava korišćenjem signala takta kao što je prikazano na Slici 5.3. Moguće je i rešenje korišćenjem sinhronog upisa pri čemu je potrebno u brojač upisati podatak 0 nakon detekcije stanja 9 brojača. Ovo rešenje je prikazano na Slici 5.4.



Slika 5.2. Brojač po modulu 10, korišćenjem asinhronog RESETa, realizacija 1

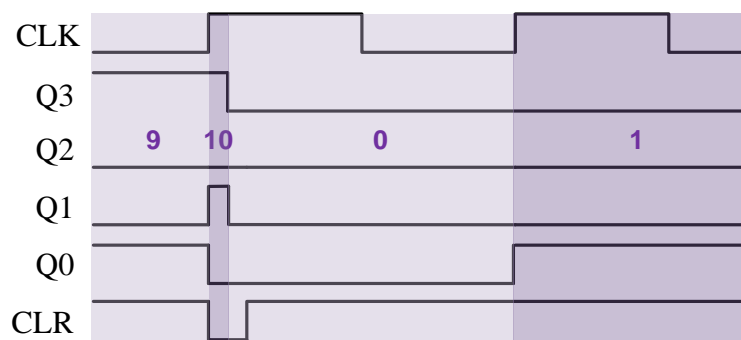


Slika 5.3. Brojač po modulu 10, korišćenjem asinhronog RESETa, realizacija 2

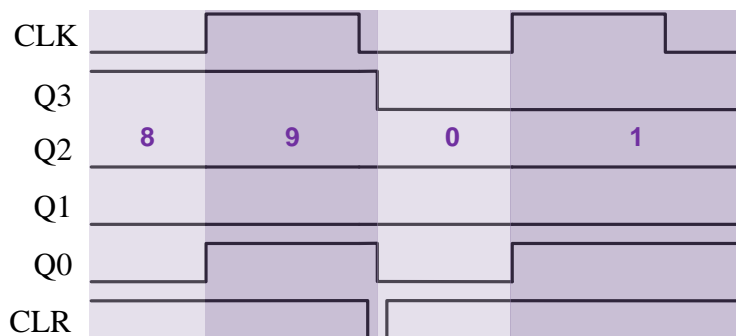


Slika 5.4. Brojač po modulu 10, korišćenjem sinhronog upisa

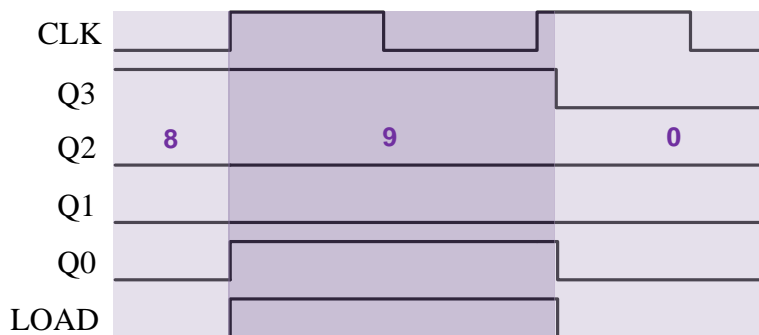
c) Na Slikama 5. 5., 5.6. i 5.7. su prikazani vremenski dijagrami relevantnih signala za sve tri realizacije.



Slika 5.5. Vremenski dijagrami realizacije sa asinhronim RESET-om varijanta 1

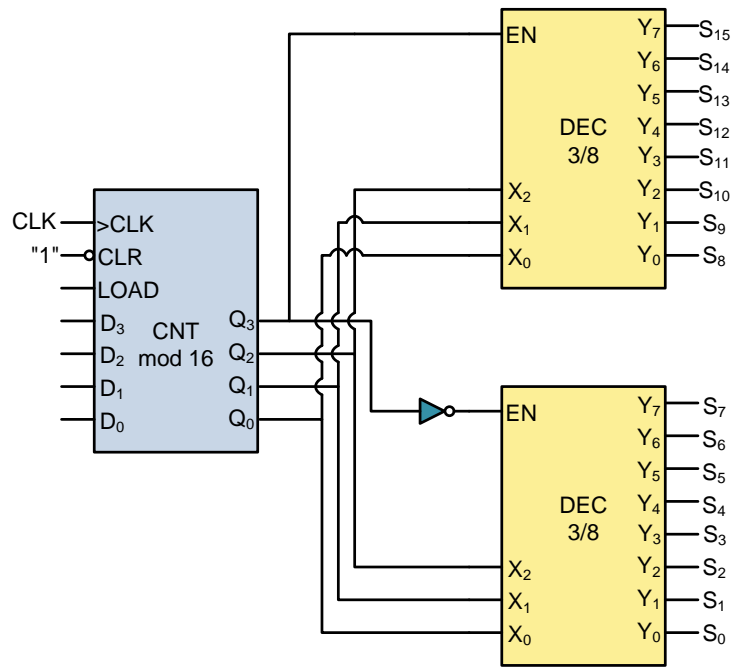


Slika 5.6. Vremenski dijagrami realizacije sa asinhronim RESET-om varijanta 2



Slika 5.7. Vremenski dijagrami realizacije sa sinhronim upisom

d) Kako projektovani brojač ima 4 bita za dekodovanje svih stanja brojača potrebna su nam dva dekodera 3/8 njihovo povezivanje je prikazano na Slici 5.8.



Slika 5.8. Dekodovanje stanja 4-bitnog brojača