

ETF U BEOGRADU, KATEDRA ZA ELEKTRONIKU

Predmet: OSNOVI DIGITALNE ELEKTRONIKE

OCENA _____

Ispit: 21.01.2015.

Odgovorni nastavnik i asistenti: Dragan Vasiljević, Goran Savić i Lazar Karbunar

DEŽURNI:

KANDIDAT:

Sala _____
Vreme početka _____
Vreme završetka _____
Potpis _____

Ime _____
Prezime _____
Broj indeksa _____
Potpis _____

USLOVI ISPITA

1. Trajanje ispita 180 minuta.
2. Ispit se polaže na formularu.
3. Dozvoljeni su kalkulator i olovka.
4. Ocenjuju se rad kandidata i razumevanje gradiva.
5. Traži se koncizan, jasan, čitak odgovor napisan u predviđenom prostoru (linija, boks, crtež).

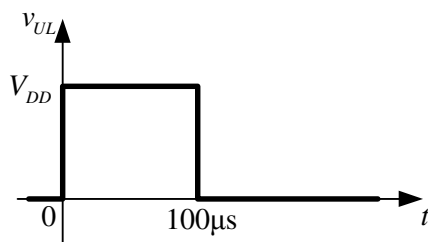
OCENJIVANJE

R.Br.	1	2	3	4	Ukupno
Max	20	20	20	20	80
Dobijeno					

1. Ulazi trouglažnog CMOS NI logičkog kola su kratko spojeni, a izlaz je opterećen kondenzatorom kapacitivnosti $C = 20\text{pF}$. Na ulaz logičkog kola se dovodi naponski impuls v_{UL} trajanja $100\mu\text{s}$, prikazan na slici. Svaki od MOS tranzistora koji čine logičko kolo, u neprovodnom režimu ima beskonačnu otpornost između drejna i sorsa. Svaki od NMOS tranzistora koji čine logičko kolo, u provodnom režimu ima otpornost od 20Ω između drejna i sorsa. Logičko kolo se napaja sa $V_{DD} = 5\text{V}$.

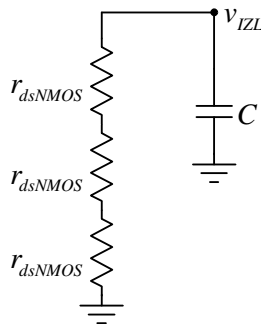
a) [10] Ako su otpornosti između drejna i sorsa PMOS tranzistora u provodnom režimu međusobno jednake, odrediti njihovu vrednost tako da vremena trajanja usponske i silazne ivice u odzivu na pobudni impuls budu međusobno jednaka. Pod vremenom trajanja usponske (silazne) ivice signala na izlazu se podrazumeva vreme koje protekne od trenutka kada naponski nivo dostigne vrednost 10% (90%) od ukupne promene nivoa, do trenutka kada naponski nivo dostigne 90% (10%) od ukupne promene nivoa.

b) [10] Pod uslovom iz tačke a) izračunati i nacrtati vremenski oblik izlaznog napona logičkog kola v_{IZL} .

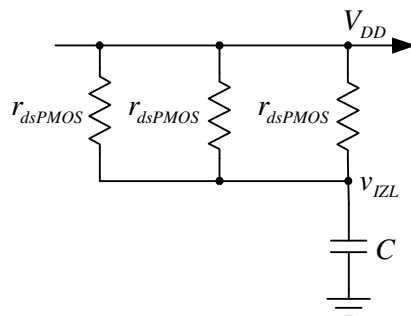


Rešenje:

a) Kada se nivo ulaznog signala promeni sa 0V na 5V, ekvivalentna šema kola je:



Kada se nivo ulaznog signala promeni sa 5V na 0V, ekvivalentna šema kola je:



Vremenska konstanta pražnjenja kondenzatora C (kada se naponski nivo na izlazu menja sa logičke jedinice na logičku nulu) je:

$\tau_{SIL} = C \cdot 3r_{dsNMOS} = 1,2ns$ (što je mnogo kraće od vremena trajanja pobudnog impulsa).

Vremenska konstanta punjenja kondenzatora C (kada se naponski nivo na izlazu menja sa logičke nule na logičku jedinicu) je:

$$\tau_{UZZ} = C \cdot \frac{r_{dsPMOS}}{3}.$$

Da bi vremena trajanja usponske i silazne ivice u odzivu na pobudni impuls bila međusobno jednaka neophodno je da vremenske konstante τ_{SIL} i τ_{UZZ} budu međusobno jednake. To će biti zadovoljeno ako je:

$$3 \cdot r_{dsNMOS} = \frac{r_{dsPMOS}}{3} \Rightarrow r_{dsPMOS} = 9 \cdot r_{dsNMOS} \Rightarrow r_{dsPMOS} = 180\Omega.$$

b) U ustaljenom stanju pre pojave pobudnog impulsa napon na izlazu logičkog kola je bio na nivou logičke jedinice (5V). Kada se nivo ulaznog signala promeni sa 0V na 5V, napon na izlazu logičkog kola se menja u skladu sa jednačinom:

$$v_{IZL}(t) = v_{IZL}(\infty) - [v_{IZL}(\infty) - v_{IZL}(0^+)] \cdot e^{-\frac{t}{\tau_{SIL}}}$$

$$v_{IZL}(\infty) = 0; \quad v_{IZL}(0^+) = v_{IZL}(0^-) = V_{DD}$$

$$v_{IZL}(t) = V_{DD} \cdot e^{-\frac{t}{1,2ns}} = V_{DD} \cdot e^{-8,333 \cdot 10^8 \cdot t}; \quad 0 \leq t \leq 100\mu s;$$

Zbog toga što je $\tau_{SIL} \ll 100\mu s$, signal na izlazu logičkog kola će praktično dostići novu stacionarnu vrednost (0V) pre pojave silazne ivice pobudnog impulsa.

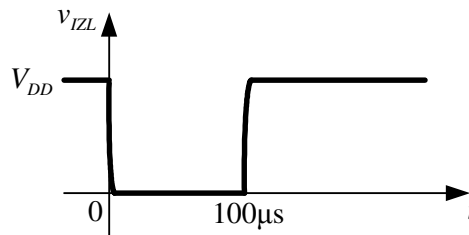
Kada se nivo ulaznog signala promeni sa 5V na 0V, napon na izlazu logičkog kola se menja u skladu sa jednačinom:

$$v_{IZL}(t) = v_{IZL}(\infty) - [v_{IZL}(\infty) - v_{IZL}(t_0)] \cdot e^{-\frac{t-t_0}{\tau_{UZZ}}}; \quad t_0 = 100\mu s;$$

$$v_{IZL}(\infty) = V_{DD}; \quad v_{IZL}(t_0^-) = v_{IZL}(t_0^+) = 0;$$

$$v_{IZL}(t) = V_{DD} - [V_{DD} - 0] \cdot e^{-\frac{t-t_0}{1,2ns}} = V_{DD}(1 - e^{-8,333 \cdot 10^8 \cdot (t-t_0)}); \quad t \geq 100\mu s.$$

Vremenski oblik izlaznog napona je prikazan na sledećoj slici:



(NAPOMENA: silazna i uzlazna ivica signala v_{IZL} su eksponencijalnog karaktera, opisane gornjim jednačinama, iako se to na dijagramu manje jasno vidi).

2. a) [10] Projektovati kombinacionu mrežu, u minimalnoj formi, koja na izlazu generiše kvadrat ulaznog trobitnog broja (binarno kodovani pozitivni celi brojevi). Nacrtati traženu mrežu sa minimalnim brojem logičkih kola ako su na raspolaganju: dvoulazna I kola, dvoulazna ILI kola i invertori.

b) [10] Kombinacionu mrežu iz prethodne tačke realizovati korišćenjem minimalnog broja jednobitnih memorijskih PROM elemenata kapaciteta 16 reči sa dvodimenzionim adresiranjem. Za svaki memorijski element je potrebno simbolički nacrtati dekodera (2/4) i multiplexer (4/1) sa odgovarajućim upravljačkim signalima a funkciju koja se realizuje tim elementom predstaviti postavljanjem tačaka koje označavaju prisustvo prekidačkog elementa na preseku odgovarajućih linija dekodera i multipleksera (ucrtati takođe i pasivne komponente koje učestvuju u realizaciji odgovarajuće funkcije).

Rešenje:

a)

Kombinaciona tabela 1. opisuje traženu mrežu:

X dec.	X_4	X_3	X_2	X_1	X^2 dec.	Y_5	Y_4	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	1	1	0	0	0	0	0	1
2	0	0	1	0	4	0	0	0	1	0	0
3	0	0	1	1	9	0	0	1	0	0	1
4	0	1	0	0	16	0	1	0	0	0	0
5	0	1	0	1	25	0	1	1	0	0	1
6	0	1	1	0	36	1	0	0	1	0	0
7	0	1	1	1	49	1	1	0	0	0	1

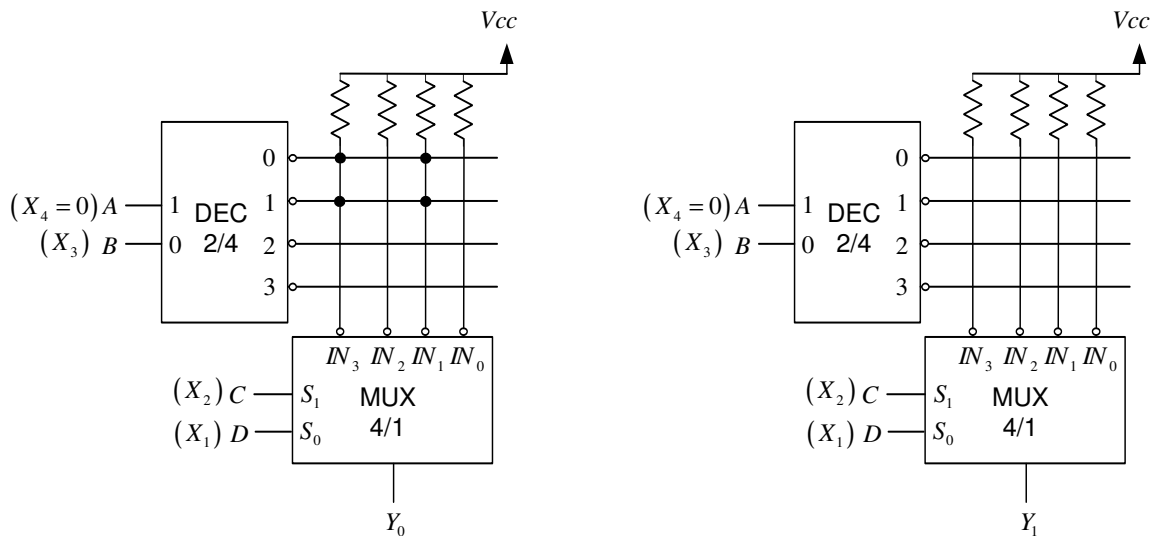
Tabela 1.

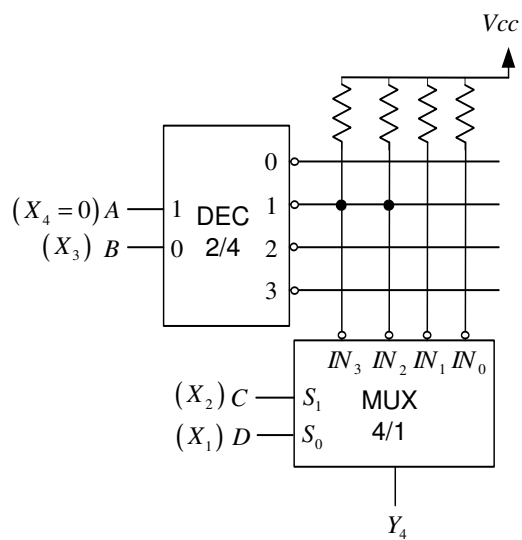
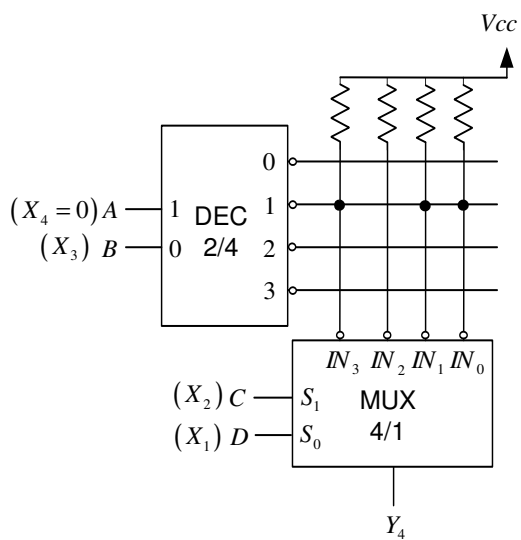
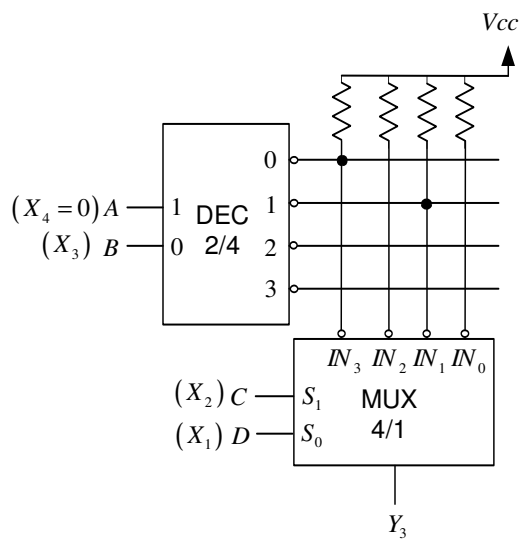
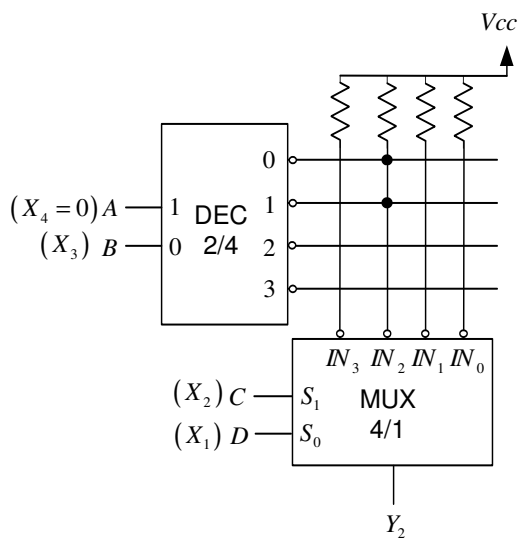
Minimizacijom pomoću Karnoovih mapa, dobija se:

$$Y_0 = X_1, \quad Y_1 = 0, \quad Y_2 = X_2 \bar{X}_1, \quad Y_3 = X_1(X_3 \oplus X_2),$$

$$Y_4 = X_3(\bar{X}_2 + X_1), \quad Y_5 = X_3 X_2.$$

b)



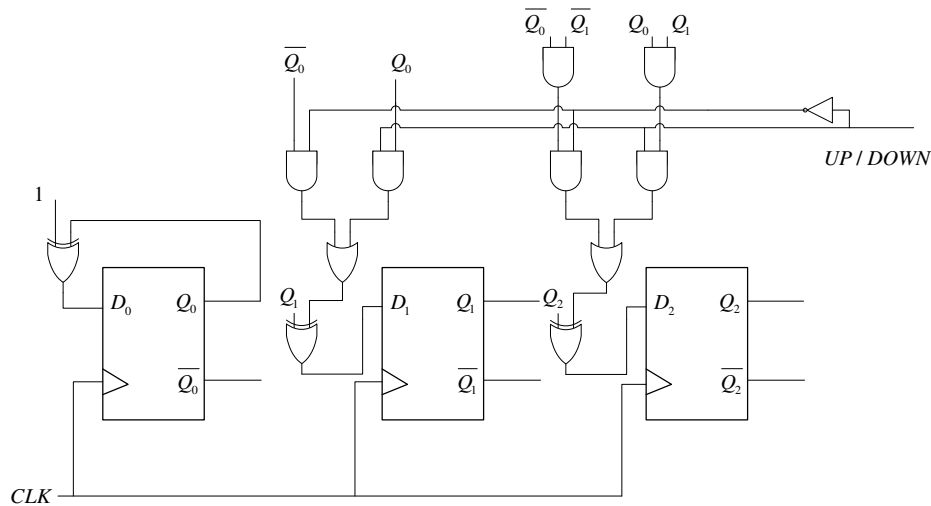


Napomena!! Memorijski modul koji realizuje Y_1 je nepotreban, pošto je $Y_1 = 0$ za sve vrednosti ulaznih promenljivih.

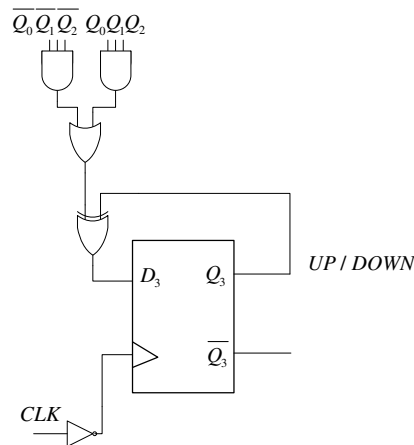
3. [20] Konstruisati trobitni brojač sa D flip flopovima i potrebnim logičkim kolima koji broji u sekvenci 0,1,2,3,4,5,6,7. Kada se brojač nađe u stanju 7 kreće da broji unazad u sekvenci 7,6,5,4,3,2,1,0 da bi po dolasku u stanje 0 ponovo počeo da broji unapred. Stanje 7 i 0 traju samo jedan taktni interval. Koristiti minimalan broj logičkih kola. Nacrtati električnu šemu povezivanja logičkih kola i flip flopova.

Rešenje:

Na slici je prikazana realizacija obostranog trobitnog brojača (gore-dole) u zavisnosti od kontrolnog signala UP/DOWN.



Da bi brojač menjao režim rada, potrebno je uvesti dodatni memorijski element koji će na stanje 7 odnosno 0 promeniti sopstveno stanje. Izlaz tog dodatnog memorijskog elementa se može koristiti kao upravljačku signal UP/DOWN.



Da bi se krajnja stanja (0 i 7) zadržala samo jedan taktni interval potrebno je upravljački element taktovati na silaznu ivicu signala takta tako da se promena smera brojanja obavlja između pojave dve susedne uzlazne ivice signala takta (u suprotnom kolo neće ispravno funkcionisati).

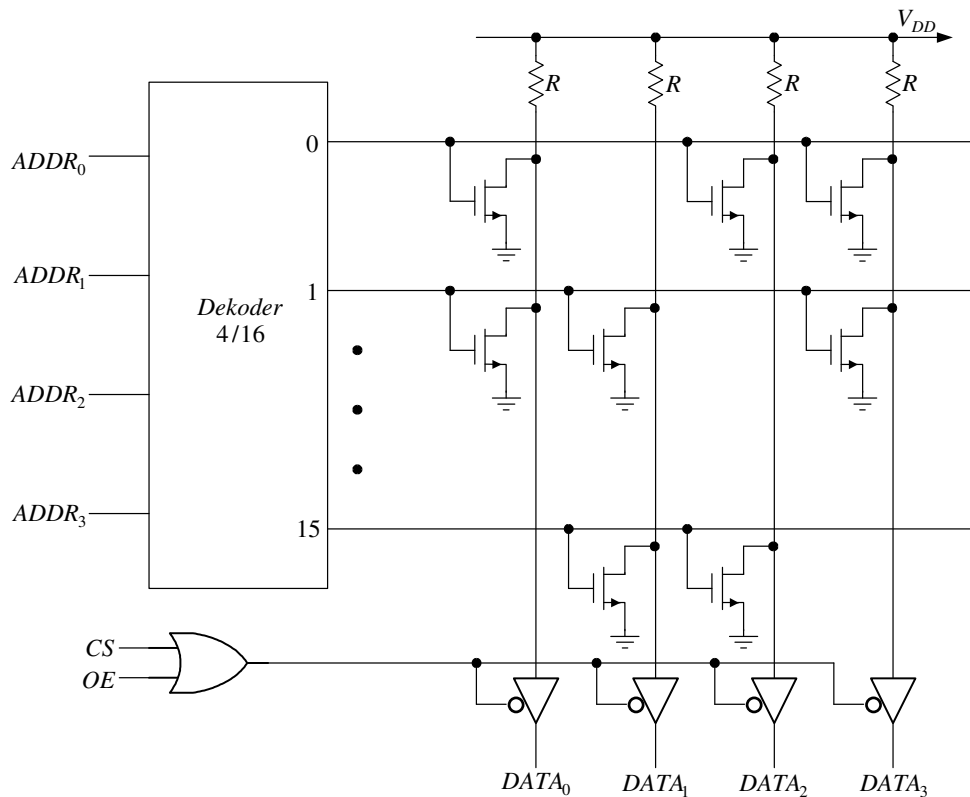
4. a) [5] Nacrtati realizaciju ROM (*Read only*) memorije kapaciteta 16 četvorobitnih reči, sa MOSFET tranzistorima kao memorijskim elementima i jednodimenzionim adresiranjem.

b) [15] Koristeći metod dizajniranja na bazi opisa ponašanja sistema, napraviti VHDL model dekodera vrsta koji je deo realizacije ROM memorije iz tačke a).

Napomena: Ocenjuje se precizna upotreba sintakse.

Rešenje:

a) Tražena realizacija ROM memorije je prikazana na sledećoj slici:



b) Traženi VHDL kod dekodera vrsta je:

```
library IEEE;
use IEEE.std_logic_1164.all;

entity decoder is
    port (ADDR: in STD_LOGIC_VECTOR (3 downto 0);
          Y: out STD_LOGIC_VECTOR (15 downto 0));
end decoder;

architecture behav of decoder is
begin
    process (ADDR)
    begin
        case ADDR is
            when "0000" => Y <= "0000000000000001";
            when "0001" => Y <= "0000000000000010";
            when "0010" => Y <= "0000000000000100";
            when "0011" => Y <= "0000000000001000";
            when "0100" => Y <= "0000000000010000";
            when "0101" => Y <= "0000000000100000";
```

```
when "0110" => Y <= "0000000001000000";
when "0111" => Y <= "0000000010000000";
when "1000" => Y <= "0000000100000000";
when "1001" => Y <= "0000001000000000";
when "1010" => Y <= "0000010000000000";
when "1011" => Y <= "0000100000000000";
when "1100" => Y <= "0001000000000000";
when "1101" => Y <= "0010000000000000";
when "1110" => Y <= "0100000000000000";
when "1111" => Y <= "1000000000000000";
when others => Y <= "0000000000000000";
end case;
end process;
end behav;
```