

Predmet: OSNOVI DIGITALNE ELEKTRONIKE

OCENA _____

Ispit: 22.08.2008.

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 240 minuta.
2. Ispit se polaže na formularu.
3. Dozvoljeni su kalkulator i hemijska olovka.
4. Ocenjuju se rad kandidata i sposobnost rezonovanja.
5. Traži se koncizan, jasan, čitak odgovor napisan u predviđenom prostoru (linija, boks, crtež).

OCENJIVANJE

R.Br.	1	2	3	4	5	6	7	8	Total
Max	10	10	10	10	10	10	10	10	80
Dobijeno									

1. a) [2] Definisati dimenzije i strukturu tabelarnog zapisa binarne prekidačke funkcije od N n nezvisnopromenljivih.
- b) [5] Ako fukcija iz tačke a) ima vrednost jedan na k slogova i vrednost nula na N $2^n - k$ slogova, definisati zapise funkcije preko sume potpunih proizvoda i preko proizvoda potpunih suma.
- c) [3] Ako je k manje od N $2^n/2$, koji je zapis iz tačke b) prostiji?

Rešenje:

Pred.4, str.3.

2. a) [3] Definisati funkciju binarnog dekodera i pojam potpunog binarnog dekodera.
b) [3] Definisati funkciju binarnog koda i pojam potpunog binarnog koda.
c) [4] Kako se definiše PROM kombinaciona PLD struktura?

Rešenje:

Pred.5, str.3.,5.,6.

- 3. a)** [3] Nacrtati sedmosegmentni LED indikator za jednu cifru sa zajedničkom katodom i napisati njegovu funkcionalnu tabelu paljenja segmenata.
- b)** [7] Projektovati dekodersku mrežu koja pobuđuje indikator iz tačke a., kada je na ulaz dekodera povezan izlaz brojača koji broji 0,1,2,3,4,5,6,7,8,9,0 u prirodnom binarnom kodu.

Rešenje:

Pred.6.,str.14.,15.

- 4. a)** [5] Nacrtati šemu A/D konvertora sa komparatorima (fleš konvertor) za N bita i napisati funkcionalnu tabelu njegovog koda.
- b)** [5] Definisati po konstrukciji i primeni PGA ASIC kolo (programmable gate array application specific integrated circuit).

Rešenje:

- a)** Pred.9, str.3.
- b)** Pred.10. i 4.

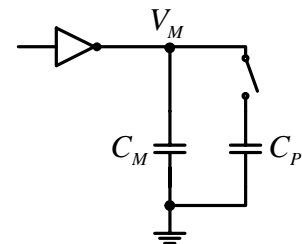
KANDIDAT: Ime _____ Prezime _____ Broj indeksa _____
Dežurni _____

5. Električna linija za prenos podataka može da se ekvivalentno predstavi sa kapacitivnošću $C_M = 10\text{pF}$. Linija podataka se priključuje na izlaz jednog logičkog CMOS kola koje određuje logičko stanje linije (drajver) i na ulaz N logičkih CMOS kola koja se pobuđuju drajverom. Ulazna kapacitivnost jednog logičkog kola je $C_{ul} = 0.1\text{pF}$.

a) [7] Nacrtati vremenski dijagram napona V_M na liniji podataka ako se u trenutku t_0 istovremeno priključe ulazi od 64 logička kola, smatrajući da je na izlazu CMOS drajvera stanje logičke jedinice.

b) [3] Koliki se maksimalni broj ulaza logičkih kola sme istovremeno da uključi na liniju podataka pobuđenu jednim CMOS drajverom, a da ne dođe do promene logičkog stanja na liniji podataka. Prag logičkog odlučivanja CMOS kola je jednak polovini napona napajanja.

NAPOMENA: Raspodela naelektrisanja u čisto kapacitivnoj konturi se menja trenutno, što ne važi za RC konturu.



Slika 1.

Rešenje:

Model koji zadovoljava uslov zadatka prikazan je na slici 1. Kapacitivnost C_P predstavlja ekvivalentnu kapacitivnost N logičkih kola:

$$C_P = NC_{ul}.$$

Kada je prekidač sa slike 1 otvoren (logička kola nisu priključena), napon na liniji podataka je V_M . Količina naelektrisanja na kondenzatoru C_M , kojim je modelovana linija, je:

$$Q = C_M V_M.$$

Kada se zatvori prekidač trenutno dolazi do preraspodele naelektrisanja između kondenzatora C_M i kondenzatora C_P . Prilikom ovog procesa ukupna količina naelektrisanja se ne menja, ali se menja napon na liniji koji sada iznosi V_P . Nakon priključenja logičkih kolâ važi:

$$Q = (C_M + C_P)V_P.$$

Izjednačavanjem izraza za naelektrisanje pre i posle uključenja prekidača dobija se:

$$V_P = \frac{C_M}{C_M + C_P} V_M.$$

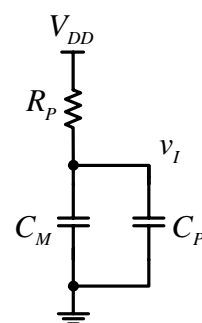
Iz izraza se vidi da ukoliko je na liniji podataka logička nula nema promene napona na izlazu. Stoga je od interesa jedino slučaj kada je na liniji podataka logička jedinica ($V_M = V_{DD}$), kako prikazuje slika 2.

Nakon trenutne preraspodele naelektrisanja dolazi do povratka kola u stacionarno stanje po eksponencijalnom zakonu, sa parametrima

$$v_I(0^+) = \frac{C_M}{C_M + C_P} V_M,$$

$$v_I(\infty) = V_{DD},$$

$$\tau = R_P (C_M + C_P).$$



Slika 2.

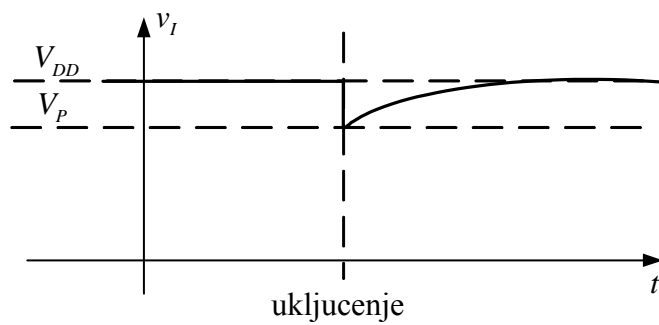
Vremenski oblik napona na liniji podataka prikazan je na slici 3.

Maksimalni broj logičkih kola koja se istovremeno mogu priključiti na liniju određen je minimalno dozvoljenim padom napona na liniji podataka. Ukoliko se pretpostavi da logička kola imaju prag logičkog odlučivanja na polovini napona napajanja, sledi da mora važiti

$$\frac{C_M}{C_M + NC_{ul}} V_{DD} \geq \frac{V_{DD}}{2},$$

odakle se dobija

$$N \leq \frac{C_M}{C_{ul}} = 100.$$



Slika 3.

6. a) [5] Korišćenjem dva D flip-flopa, i potrebnog broja NI, NILI i EXILI logičkih kola, projektovati sekvencijalnu mrežu sa četiri stanja koja se ciklično ponavljaju: $S_3:S_2:S_1:S_0 \rightarrow 1000-1101-0101-0110-1000...$
 b) [5] Ponoviti prethodnu tačku ako nisu na raspolaganju EXILI kola.

Rešenje:

Kako mreža ima 4 stanja dovoljna su 2 flipflopa. Neka je pomoću njih napravljen standardni binarni brojač koji ponavlja sekvencu 00–01–10–11. Na osnovu stanja flip–flopova treba generisati traženu sekvencu, tabela 1.

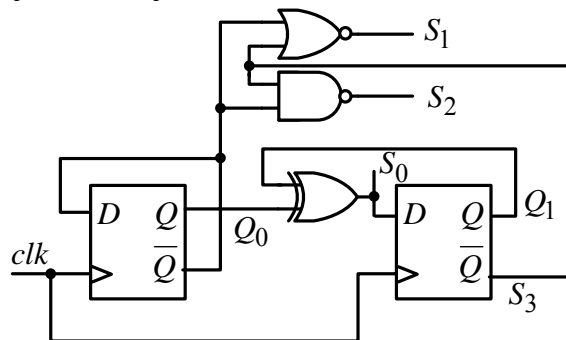
Q_1	Q_0	S_3	S_2	S_1	S_0
0	0	1	0	0	0
0	1	1	1	0	1
1	0	0	1	0	1
1	1	0	1	1	0

Tabela 1.

Očigledno je da važi:

$$S_3 = \overline{Q_1}, \quad S_2 = Q_0 + Q_1 = \overline{Q_0} \cdot \overline{Q_1}, \quad S_1 = Q_0 \cdot Q_1 = \overline{Q_0} + \overline{Q_1}, \quad S_0 = Q_0 \oplus Q_1.$$

Na slici 4. prikazana je realizacija tražene mreže.



Slika 4.

b) Za rešenje u tački a) bi se moglo reći da je „pravolinijsko“. Ono ima, u određenim uslovima, nedostatak što koristi EXILI kolo koje u pojedinim programabilnim tehnologijama, može da bude ekvivalentno korišćenju nekoliko NI kola.

Poznato je da se primenom šift registara mogu efikasno realizovati različiti brojači. Činjenica da oni ne generišu sekvencu uzastopnih binarnih brojeva nije bitna u situacijama kada ta sekvencu nije potrebna.

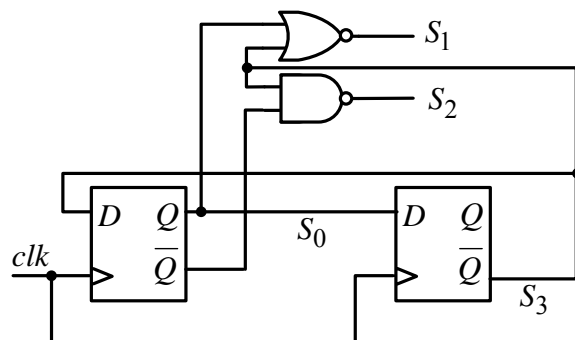
Za rešenje zadatka može efikasno da se iskoristi brojač koji generiše sekvencu 00–01–11–10. Na osnovu stanja flip–flopova treba generisati traženu sekvencu, tabela 2.

Q_1	Q_0	S_3	S_2	S_1	S_0
0	0	1	0	0	0
0	1	1	1	0	1
1	1	0	1	0	1
1	0	0	1	1	0

Tabela 2.

Izlazni kod se u ovom slučaju dobija iz jednačina:

$S_3 = \overline{Q_1}$, $S_2 = Q_0 + Q_1 = \overline{\overline{Q_0} \cdot \overline{Q_1}}$, $S_1 = \overline{Q_0} \cdot Q_1 = \overline{Q_0 + \overline{Q_1}}$, $S_0 = Q_0$.
Na slici 5 prikazana je realizacija tražene mreže.



Slika 5.

7. a) [7] Na slici je prikazana šema četvorobitnog bipolarnog D/A konvertora sa lestvičastom R-2R otpornom mrežom.

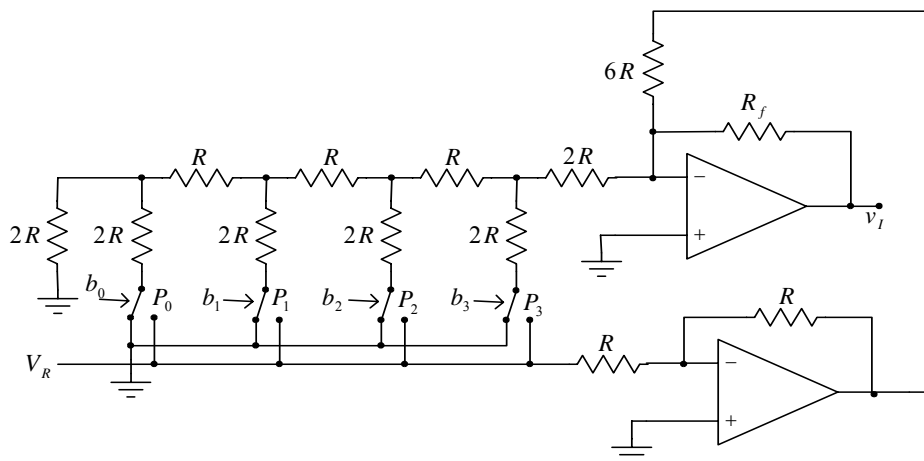
Bipolarni konvertor prikazuje binarnu vrednost 0000 sa po modulu najvećim negativnim naponom, a binarnu vrednost 1111 sa najvećim pozitivnim naponom. Vrednost 1000 se prikazuje na izlazu sa analognim naponom od nula volti.

Na ulaz D/A konvertora se dovodi binarni broj $b_3b_2b_1b_0$. Na izlazu konvertora se dobija analogni napon v_I koji je ekvivalent binarnog broja sa ulaza.

Prekidači P_i ($i=0,1,2,3$) povezuju donji kraj odgovarajućih otpornika $2R$ na napon V_R kada je vrednost odgovarajućeg bita b_i koji kontroliše taj prekidač jednaka 1, odnosno, na masu kada je $b_i=0$. Otpornost prekidača u stanju provođenja je nula.

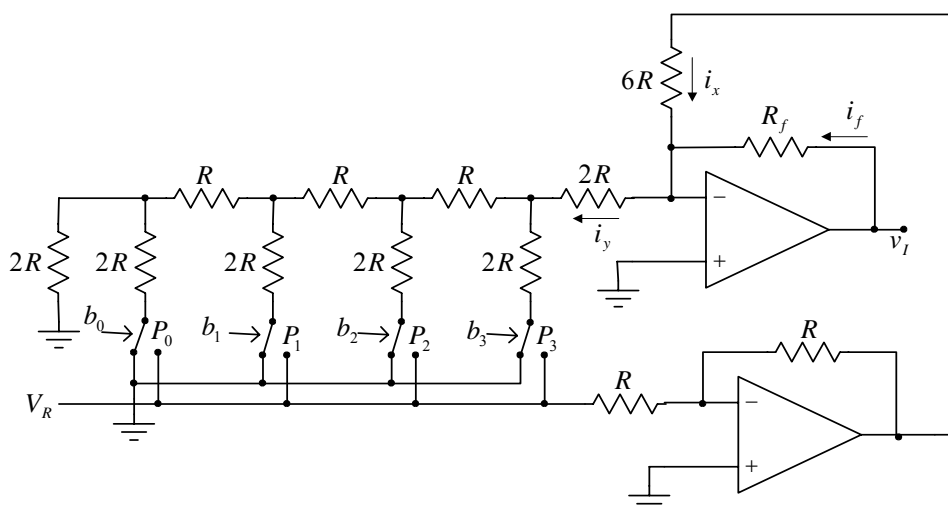
Izvesti izraz za napon v_I u zavisnosti od napona V_R , otpornosti R , R_f i vrednosti binarnih promenljivih b_3, b_2, b_1, b_0 .

b) [3] Ako je poznato da je $V_R = -6V$, odrediti odnos koji treba da zadovoljavaju otpornici R i R_f tako, da za maksimalnu vrednost binarnog broja na ulazu, izlazni napon bude $v_I = 4,375V$.



Rešenje:

a)



$$v_I = R_f \cdot i_f = R_f \cdot (i_y - i_x)$$

$$i_x = \frac{-V_R}{6R}$$

$$i_y = \frac{1}{2} \cdot \frac{(-V_R)}{3R} \cdot b_3 + \frac{1}{4} \cdot \frac{(-V_R)}{3R} \cdot b_2 + \frac{1}{8} \cdot \frac{(-V_R)}{3R} \cdot b_1 + \frac{1}{16} \cdot \frac{(-V_R)}{3R} \cdot b_0$$

Iz prethodne tri jednačine se dobija:

$$v_I = -\frac{1}{16} \cdot \frac{V_R}{3R} \cdot R_f \cdot (8b_3 + 4b_2 + 2b_1 + b_0) + \frac{V_R}{6R} \cdot R_f = \frac{V_R}{6R} \cdot R_f - \frac{V_R}{48R} \cdot R_f \cdot (8b_3 + 4b_2 + 2b_1 + b_0)$$

b) Po uslovu zadatka je $v_I = 4,375 \text{ V}$ za $b_3 b_2 b_1 b_0 = 1111$. Na osnovu izraza izvedenog u tački a) dobija se:

$$v_I = \frac{-6V}{6R} \cdot R_f - \frac{-6V}{48R} \cdot R_f \cdot 15 = 4,375V$$

Dalje sledi:

$$-\frac{R_f}{R} + \frac{15R_f}{8R} = 4,375 \text{ odnosno:}$$

$$\frac{7R_f}{8R} = 4,375 \text{ odakle se konačno dobija da je traženi odnos:}$$

$$\frac{R_f}{R} = 5$$

8. [10] Koristeći metod dizajniranja na bazi opisa ponašanja sistema, napraviti VHDL model osmobitnog registra prema sledećim specifikacijama.

U pomenuti registar treba da se paralelno upisuje osmobitni sadržaj doveden na ulaz d (koji predstavlja osmobitni vektor) registra, sinhrono sa uzlaznom ivicom signala takta clk samo ukoliko je signal sinhronog reseta rst na niskom logičkom nivou i ukoliko je ulazni kontrolni signal rot na niskom logičkom nivou. Ukoliko je signal sinhronog reseta rst na visokom logičkom nivou, sadržaj registra treba da se resetuje sinhrono sa uzlaznom ivicom signala takta clk , bez obzira na trenutnu vrednost signala d i rot . I konačno, ukoliko je signal sinhronog reseta rst na niskom logičkom nivou, a ulazni kontrolni signal rot na visokom logičkom nivou, sadržaj registra treba da se rotira za jedno mesto udesno sinhrono sa uzlaznom ivicom signala takta clk , bez obzira na trenutnu vrednost signala d . Trenutni sadržaj registra je uvek raspoloživ u vidu izlaznog osmobitnog vektora q .

Biti ulaznog i izlaznog vektora (d i q) su poređani od MSB ka LSB u smeru sa leva na desno.

Napomena: Ocenjuje se precizna upotreba sintakse.

Rešenje:

```

LIBRARY ieee;
USE ieee.std_logic_1164.all;

ENTITY eight_bit_register IS PORT
(
    clk,rst,rot : IN STD_LOGIC;
    d : IN STD_LOGIC_VECTOR(7 DOWNTO 0);
    q : OUT STD_LOGIC_VECTOR(7 DOWNTO 0)
);
END eight_bit_register;

ARCHITECTURE behav OF eight_bit_register IS
    SIGNAL tmp : STD_LOGIC_VECTOR(7 DOWNTO 0);
BEGIN
    PROCESS (clk) BEGIN
        IF rising_edge(clk) THEN
            IF rst='1' THEN
                tmp <= (OTHERS => '0');
            ELSIF rot='1' THEN
                tmp <= tmp(0) & tmp(7 DOWNTO 1);
            ELSE
                tmp <= d;
            END IF;
        END IF;
    END PROCESS;
    q <= tmp;
END behav;

```