

ETF U BEOGRADU, KATEDRA ZA ELEKTRONIKU

Predmet: OSNOVI DIGITALNE ELEKTRONIKE

OCENA _____

Ispit: 19.01.2011.

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

DEŽURNI:

KANDIDAT:

Sala _____
Vreme početka _____
Vreme završetka _____
Potpis _____

Ime _____
Prezime _____
Broj indeksa _____
Potpis _____

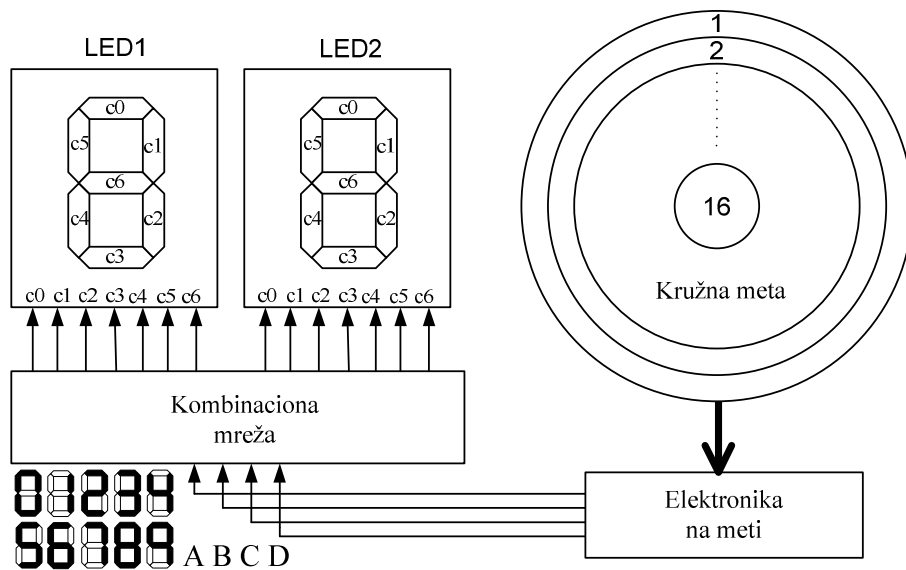
USLOVI ISPITA

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	Total
Max	20	20	20	20	80
Dobijeno					

1. [20] Elektronska meta se sastoji od šesnaest koncentričnih krugova. Prilikom pogađanja pikada nekog od krugova, elektronika na meti detektuje oblast u koju je pikado udario i generiše na izlazu odgovarajući četvorobitni broj ABCD (A cifra najveće težine) koji odgovara toj oblasti (kružnom prstenu). Svakom kružnom prstenu je dodeljen jedan četvorobitni broj, a radi minimizovanja greške kada pikado udari blizu granice dva kružna prstena, susednim prstenima su dodeljeni susedni brojevi iz Grejovog koda. Prsten najveće površine je kodiran sa 0000 i ukoliko pikado udari u njega takmičar dobija 1 poen, centralni prsten (krug) je kodiran sa 1000 i ukoliko pikado udari u njega takmičar dobija 16 poena. Prsteni između periferijskog i centralnog nose poene obrnuto srazmerno svojoj veličini (kako je to prikazano na slici), ukoliko takmičar promaši celu metu takođe dobija jedan (utešni) poen. Četvorobitni brojevi koji predstavljaju kodove kružnih oblasti dolaze na digitalno kolo koje treba da izvrši konverziju koda i prikaže rezultat na dva sedmosegmentna LED displeja. Koristeći osnovna logička kola izvršiti sintezu kombinacione mreže (sintezu vršiti minimizacijom). LED1 displej prikazuje cifru veće težine, a ukoliko ona ne postoji, LED1 displej treba da prikaže nulu. Segmenti na LED displejima su upaljeni ukoliko je odgovarajući ulaz c_i na logičkoj jedinici.



Rešenje:

Tabela istinitosti za LED displeje su prikazane sledećim slikama:

Tabela za LED1

ABCD	C0	C1	C2	C3	C4	C5	C6	Poeni
0000	1	1	1	1	1	1	0	1
0001	1	1	1	1	1	1	0	2
0011	1	1	1	1	1	1	0	3
0010	1	1	1	1	1	1	0	4
0110	1	1	1	1	1	1	0	5
0111	1	1	1	1	1	1	0	6
0101	1	1	1	1	1	1	0	7
0100	1	1	1	1	1	1	0	8
1100	1	1	1	1	1	1	0	9
1101	0	1	1	0	0	0	0	10
1111	0	1	1	0	0	0	0	11
1110	0	1	1	0	0	0	0	12
1010	0	1	1	0	0	0	0	13

1011	0 1 1 0 0 0 0	14
1001	0 1 1 0 0 0 0	15
1000	0 1 1 0 0 0 0	16

Tabela za LED2

ABCD	C0 C1 C2 C3 C4 C5 C6	Poeni
0000	0 1 1 0 0 0 0	1
0001	1 1 0 1 1 0 1	2
0011	1 1 1 1 0 0 1	3
0010	0 1 1 0 0 1 1	4
0110	1 0 1 1 0 1 1	5
0111	1 0 1 1 1 1 1	6
0101	1 1 1 0 0 0 0	7
0100	1 1 1 1 1 1 1	8
1100	1 1 1 0 0 1 1	9
1101	1 1 1 1 1 1 0	10
1111	0 1 1 0 0 0 0	11
1110	1 1 0 1 1 0 1	12
1010	1 1 1 1 0 0 1	13
1011	0 1 1 0 0 1 1	14
1001	1 0 1 1 0 1 1	15
1000	1 0 1 1 1 1 1	16

Minimizacijom uz pomoć Karnoovih mapa se dobijaju izrazi za upravljačke promenjive LED displeja.

Za LED1:

$$C0 = |A+B*|C*|D$$

$$C1 = 1$$

$$C2 = 1$$

$$C3 = C0 = |A+B*|C*|D$$

$$C4 = C0 = |A+B*|C*|D$$

$$C5 = C0 = |A+B*|C*|D$$

$$C6 = 0$$

Za LED2:

$$C0 = |A*D + A*|D + A*|C + B*|D$$

$$C1 = |B*C + A*B + B*|C + |A*|B$$

$$C2 = |A*C + B*D + A*|B + |C*|D$$

$$C3 = B*C*|D + |A*C*D + A*|C*D + |A*|B*D + A*|B*|D + |A*B*|D$$

$$C4 = |A*B*|C*|D + A*|B*|C*|D + |A*|B*|C*D + A*B*|C*D + |A*B*C*D + A*B*C*|D$$

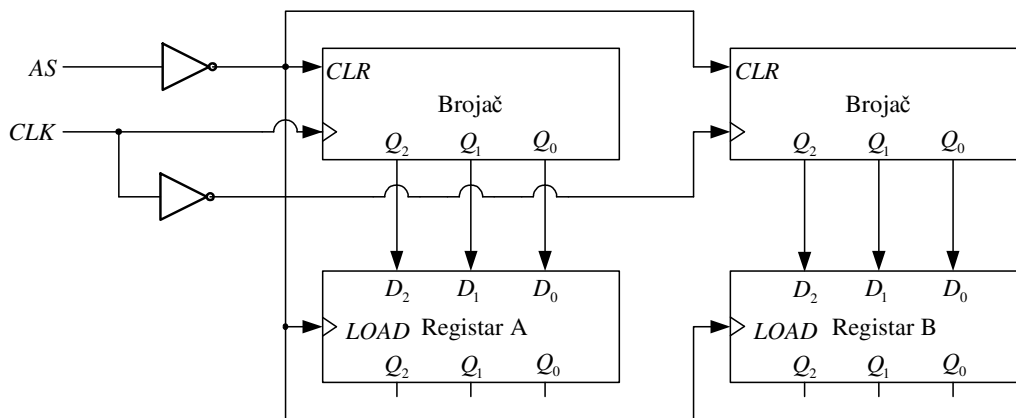
$$C5 = A*|C + |A*C*|D + |A*B*C + A*|B*D + B*|C*|D$$

Napomena $|X_i = \overline{X_i}$

2. [20] Realizovati sekvencijalnu mrežu koja u registar A upisuje broj uzlaznih ivica signala takta CLK koje je obuhvatio asinhroni signal AS (za vreme dok je bio na nivou logičke jedinice), a u registar B broj silaznih ivica signala takta CLK koje je obuhvatio asinhroni signal AS (za vreme dok je bio na nivou logičke jedinice). Trajanje signala AS je $0,5T < t_{AS} < 7T$, a razmak između susednih vremenskih intervala u kojima je AS na nivou logičke jedinice je veći od T , gde je T perioda signala takta CLK. Na raspolaganju je minimalan potreban broj registara i brojača, sa minimalnim potrebnim brojem bita, kao i potreban broj invertora. Brojači broje unapred sinhrono sa uzlaznom ivicom signala koji se dovodi na odgovarajući ulaz. Brojači poseduju i ulaz za sinhroni reset CLR (koji je aktivan na visokom logičkom nivou). Podaci u registre se upisuju sinhrono sa uzlaznom ivicom signala koji se dovodi na odgovarajući ulaz LOAD.

Rešenje:

Tražena sekvencijalna mreža je prikazana na sledećoj slici:



Kada je asinhroni signal AS na niskom logičkom nivou, sadržaj oba brojača se resetuje na svaku uzlaznu ivicu signala takta CLK (za levi brojač), odnosno silaznu ivicu takta CLK (za desni brojač). Kada je asinhroni signal AS na visokom logičkom nivou, sadržaj oba brojača se inkrementira na svaku uzlaznu ivicu signala takta (za levi brojač), odnosno silaznu ivicu takta (za desni brojač) i na taj način levi brojač broji uzlazne ivice signala takta CLK, dok desni brojač broji silazne ivice signala takta CLK. Po okončanju vremenskog intervala u kome je asinhroni signal AS bio na visokom logičkom nivou, sadržaj brojača se upisuje u odgovarajuće registre (naizmenično silazne ivice signala AS). Time se broj uzlaznih ivica signala takta CLK koje je obuhvatio asinhroni signal AS za vreme dok je bio na nivou logičke jedinice upisuje u registar A, a broj silaznih ivica signala takta CLK koje je obuhvatio asinhroni signal AS za vreme dok je bio na nivou logičke jedinice upisuje u registar B.

U realizaciji su korišćeni trobitni brojači i registri jer je po uslovu zadatka trajanje signala AS $0,5T < t_{AS} < 7T$, što znači da je minimalan broj bita potrebnih za reprezentovanje broja ivica obuhvaćenih signalom AS tri.

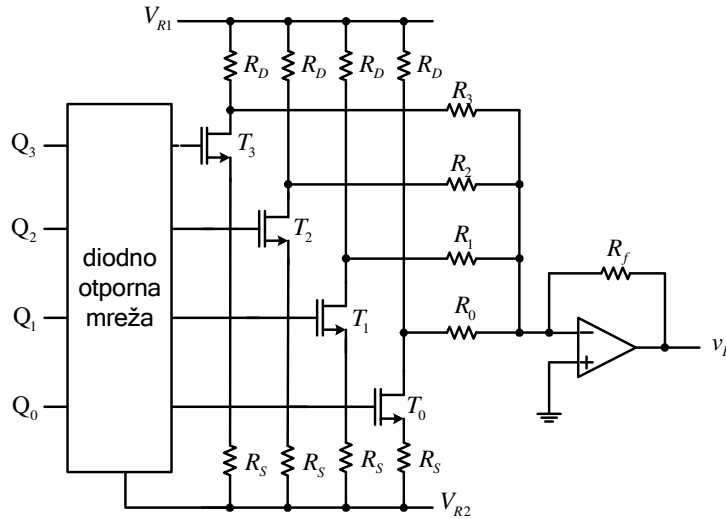
3. [20] NMOS tranzistori u kolu D/A konvertora sa slike rade kao prekidači tako da je na drejnu provodnog tranzistora potencijal mase.

a) Odrediti vrednosti otpornika R_D , R_0 , R_1 , R_2 i R_3 ako se zahteva da maksimalna apsolutna vrednost izlaznog napona bude $|V_{\max}| = 7V$ a minimalna $|V_{\min}| = 0V$. Poznato je: $V_{R1} = 10V$,

$V_{R2} = -1V$, $R_S = 900\Omega$, otpornost NMOS tranzistora kada provode $r_{ON} = 100\Omega$, $R_f = 4k\Omega$. Sve ostale karakteristike svih komponenti su idealne.

b) Nacrtati vremenski dijagram izlaznog napona v_I ako je na ulaz D/A konvertora priključen binarni brojač koji broji unapred od stanja $Q_3Q_2Q_1Q_0 = 0000$ do

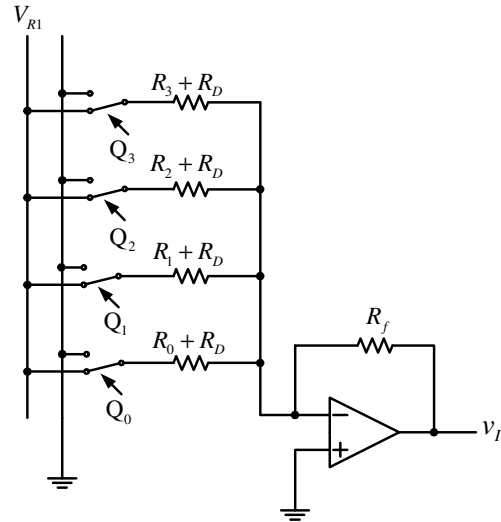
$Q_3Q_2Q_1Q_0 = 1111$, a zatim promeni smer i broji do stanja $Q_3Q_2Q_1Q_0 = 0000$. Sva kašnjenja kroz kolo sa slike su zanemarljiva.



Rešenje:

a) Na slici je prikazana ekvivalentna šema posmatranog D/A konvertora. U pitanju je unipolarni D/A konvertor u kome ulaznoj kombinaciji 1111 odgovara napon na izlazu 0V, a ulaznoj kombinaciji 0000 odgovara najmanji negativan napon na izlazu.

Vrednost otpornosti R_D se određuje iz uslova da kada tranzistor vodi nema struje kroz odgovarajući otpornik koji povezuje prekidač i izlaz kola. Kako su ovi otpornici jednim krajem vezani za masu (negativni ulaz operacionog pojačavača), potrebno je obezbediti 0V na drejnu tranzistora kada tranzistor vodi.



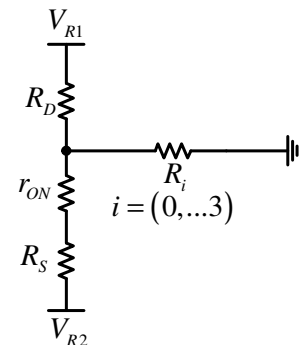
Prema slici ekvivalentnih otpornosti, važi sledeća jednačina (I Kirhofov zakon za struje za drejn tranzistora):

$$\frac{V_{R1}}{R_D} = \frac{|V_{R2}|}{R_S + r_{ON}}$$

$$R_D = \frac{V_{R1}}{|V_{R2}|} (R_S + r_{ON}) = 10k\Omega$$

Ako je na ulazu kombinacija 0000 svi prekidači su isključeni, prema izlazu teče maksimalna struja, tako da je izlazni napon najmanji mogući negativan

$$v_I = -15R_f \frac{V_{R1}}{R_0 + R_d} = -|V_{\max}| = -7V$$



$$R_0 + R_D = 15R_f \frac{V_{R1}}{7V} = 86k\Omega$$

$$R_0 = 76k\Omega$$

Na osnovu osobina otporne mreže određujemo ostale otpornosti

$$R_1 + R_d = \frac{R_0 + R_d}{2} = 43k\Omega$$

$$R_1 = 33k\Omega$$

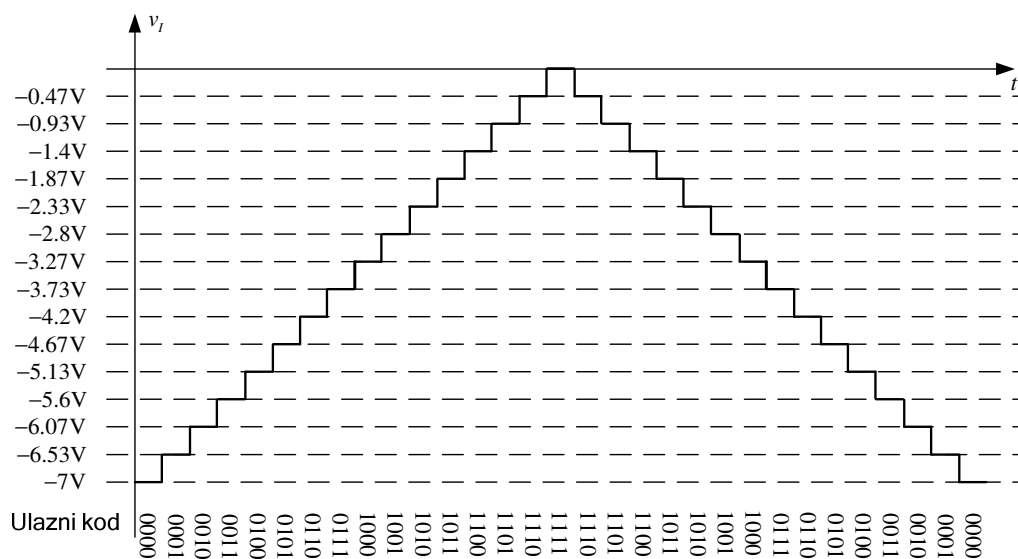
$$R_2 + R_d = \frac{R_1 + R_d}{2} = 21.5k\Omega$$

$$R_2 = 10.5k\Omega$$

$$R_3 + R_d = \frac{R_2 + R_d}{2} = 10.75k\Omega$$

$$R_3 = 750\Omega$$

b) Traženi vremenski dijagram napona prikazan je na slici



KANDIDAT: Ime _____ Prezime _____ Broj indeksa _____
 Dežurni _____

4. [20] Dat je VHDL kod kojim je opisana jedna sekvencijalna mreža.

```

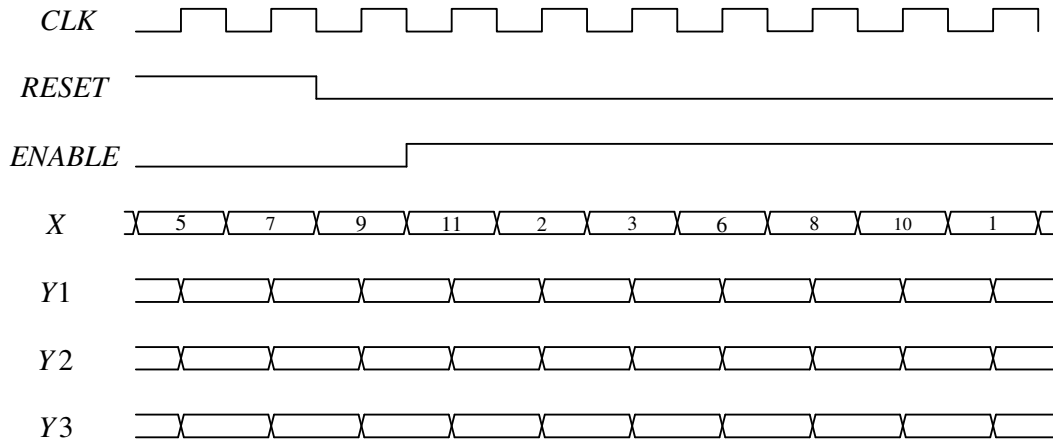
LIBRARY ieee;
USE ieee.std_logic_1164.all;
USE ieee.std_logic_arith.all;

ENTITY seq IS PORT
(
  clk,reset,enable      : IN STD_LOGIC;
  x                      : IN STD_LOGIC_VECTOR(5 DOWNTO 0);
  y1                     : OUT STD_LOGIC_VECTOR(5 DOWNTO 0);
  y2                     : OUT STD_LOGIC_VECTOR(5 DOWNTO 0);
  y3                     : OUT STD_LOGIC_VECTOR(5 DOWNTO 0)
);
END seq;

ARCHITECTURE behav OF seq IS
  TYPE states IS (s0,s1,s2,s3);
  SIGNAL current_state   : states;
  SIGNAL temp            : UNSIGNED(5 DOWNTO 0);
BEGIN
  PROCESS(clk) BEGIN
    IF rising_edge(clk) THEN
      IF (reset='1') THEN
        current_state <= s0;
        y1            <= (OTHERS => '0');
        y2            <= (OTHERS => '0');
        y3            <= (OTHERS => '0');
        temp          <= "000001";
      ELSIF (enable='1') THEN
        CASE current_state IS
          WHEN s0 => y1      <= (OTHERS => '0');
                   y2      <= (OTHERS => '0');
                   y3      <= (OTHERS => '0');
                   temp    <= "000001";
                   current_state <= s1;
          WHEN s1 => y1      <= STD_LOGIC_VECTOR(temp);
                   y2      <= x(4 DOWNTO 0) & x(0);
                   y3      <= x(5) & x(5 DOWNTO 1);
                   temp    <= temp + UNSIGNED(x);
                   current_state <= s2;
          WHEN s2 => y1      <= x(3 DOWNTO 0) & x(0) & x(0);
                   y2      <= x;
                   y3      <= x(5) & x(4 DOWNTO 1) & x(1);
                   temp    <= temp + 2;
                   IF (x<"001000") THEN
                     current_state <= s3;
                   ELSE
                     current_state <= s1;
                   END IF;
          WHEN s3 => y1      <= STD_LOGIC_VECTOR(temp);
                   y2      <= x(3 DOWNTO 0) & x(1 DOWNTO 0);
                   y3      <= x;
                   temp    <= temp + UNSIGNED(x);
                   current_state <= s0;
        END CASE;
      END IF;
    END IF;
  END PROCESS;
END behav;

```

Na osnovu vremenskih dijagrama ulaznih signala i vektora, odrediti vrednosti izlaznih vektora i u decimalnom formatu ih upisati na dati vremenski dijagram.



Rešenje:

