

ETF U BEOGRADU, KATEDRA ZA ELEKTRONIKU

Predmet: OSNOVI DIGITALNE ELEKTRONIKE

OCENA _____

Ispit: 14.09.2012.

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

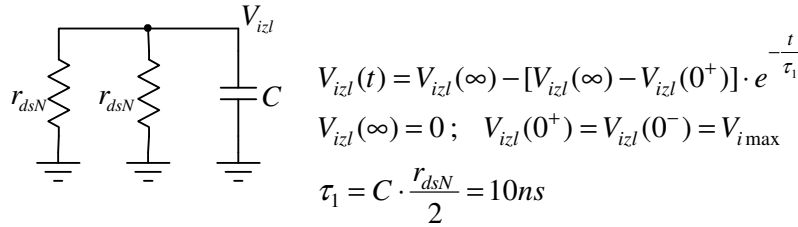
R.Br.	1	2	3	4	Total
Max	20	20	20	20	80
Dobijeno					

1. [20] Ulazi dvoulaznog NILI CMOS logičkog kola su kratko spojeni i povezani na naponski generator povorke unipolarnih pravougaonih impulsa amplitude 5V, frekvencije 50MHz i jednakog trajanja impulsa i pauze. Izlaz CMOS logičkog kola je povezan na magistralu podataka čija je ukupna kapacitivnost prema masi 50pF. Napon napajanja CMOS logičkog kola iznosi 5V. PMOS tranzistori u provodnom režimu se mogu ekvivalentirati sa otpornostima $r_{dsP} = 100 \Omega$, a u neprovodnom režimu sa $r_{dsP} \rightarrow \infty$, dok se NMOS tranzistori u provodnom režimu mogu ekvivalentirati sa otpornostima $r_{dsN} = 400 \Omega$, a u neprovodnom režimu sa $r_{dsN} \rightarrow \infty$.

Izračunati i nacrtati vremenski oblik napona na izlazu CMOS logičkog kola u ustaljenom stanju.

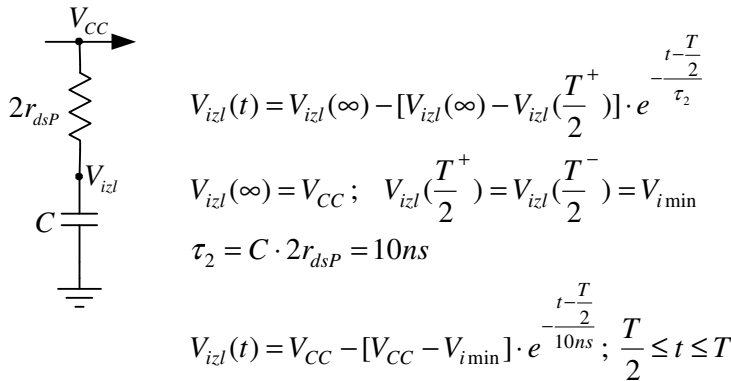
Rešenje:

Uspostavljanje logičke nule na izlazu se može predstaviti sledećom ekvivalentnom šemom:



$$V_{izl}(t) = V_{i\max} \cdot e^{-\frac{t}{10ns}}; \quad 0 \leq t \leq \frac{T}{2}; \quad T = \frac{1}{f} = 20ns$$

Uspostavljanje logičke jedinice na izlazu se može predstaviti sledećom ekvivalentnom šemom:



Iz uslova:

$$V_{izl}(\frac{T}{2}) = V_{i\min} \quad \text{i} \quad V_{izl}(T) = V_{i\max}$$

sledi:

$$V_{i\min} = V_{i\max} \cdot e^{-\frac{T}{2 \cdot 10ns}} \quad \text{i} \quad V_{i\max} = V_{CC} - [V_{CC} - V_{i\min}] \cdot e^{-\frac{T}{10ns}}$$

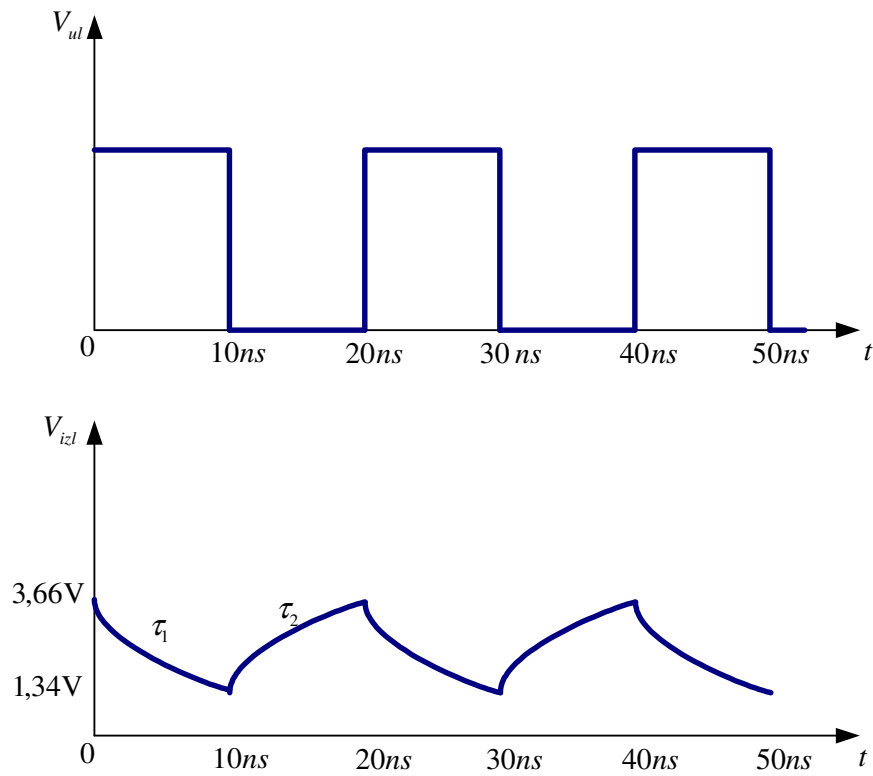
Rešavanjem dobijenih jednačina po $V_{i\min}$ i $V_{i\max}$ se dobija:

$$V_{i\max} = \frac{V_{CC}}{1 + e^{-1}} \approx 3,66V \quad \text{i} \quad V_{i\min} = \frac{e^{-1} \cdot V_{CC}}{1 + e^{-1}} \approx 1,34V$$

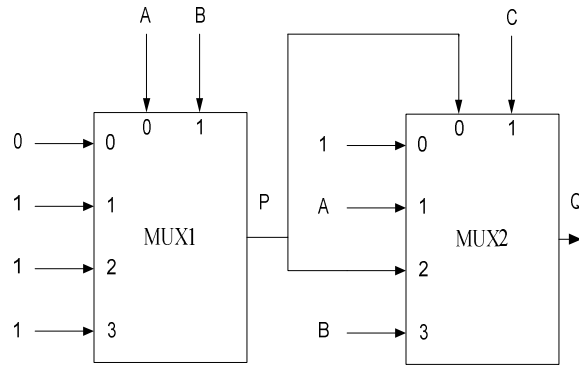
Dakle $V_{izl}(t)$ je periodičan signal sa periodom 20ns, a unutar intervala trajanja jedne periode važi:

$$V_{izl}(t) = 3,66V \cdot e^{-\frac{t}{10ns}} \quad \text{za} \quad 0 \leq t \leq 10ns \quad \text{i} \quad V_{izl}(t) = 5V - 3,66V \cdot e^{-\frac{t-10ns}{10ns}} \quad \text{za} \quad 10ns \leq t \leq 20ns.$$

Vremenski oblici signala na ulazu i izlazu datog CMOS logičkog kola su prikazani na sledećoj slici:



2. [20] Na slici je prikazano digitalno kolo sastavljeno od dva multipleksera.



- Kreirati tabelu isititosti za prikazano kolo (izlazi su P i Q)
- Koristeći se Karnoovim kartama kreirati Bulove funkcije P i Q. Uz pomoć minimalnog broja osnovnih logičkih kola izvršiti sintezu kombinacione mreže koja obavlja istu funkciju kao mreža prikazana na slici.
- Uz pomoć odgovarajućeg dekodera i minimalnog broja dvoulazinih ILI logičkih kola realizovati digitalno kolo sa slike.

Rešenje:

- Tablica istinitosti je sledeća

A	B	C	P	Q
0	0	0	0	1
0	0	1	0	0
0	1	0	1	0
0	1	1	1	1
1	0	0	1	1
1	0	1	1	0
1	1	0	1	1
1	1	1	1	1

- Minimizacijom uz pomoć Karnoovim karti se dobijaju sledeći izrazi

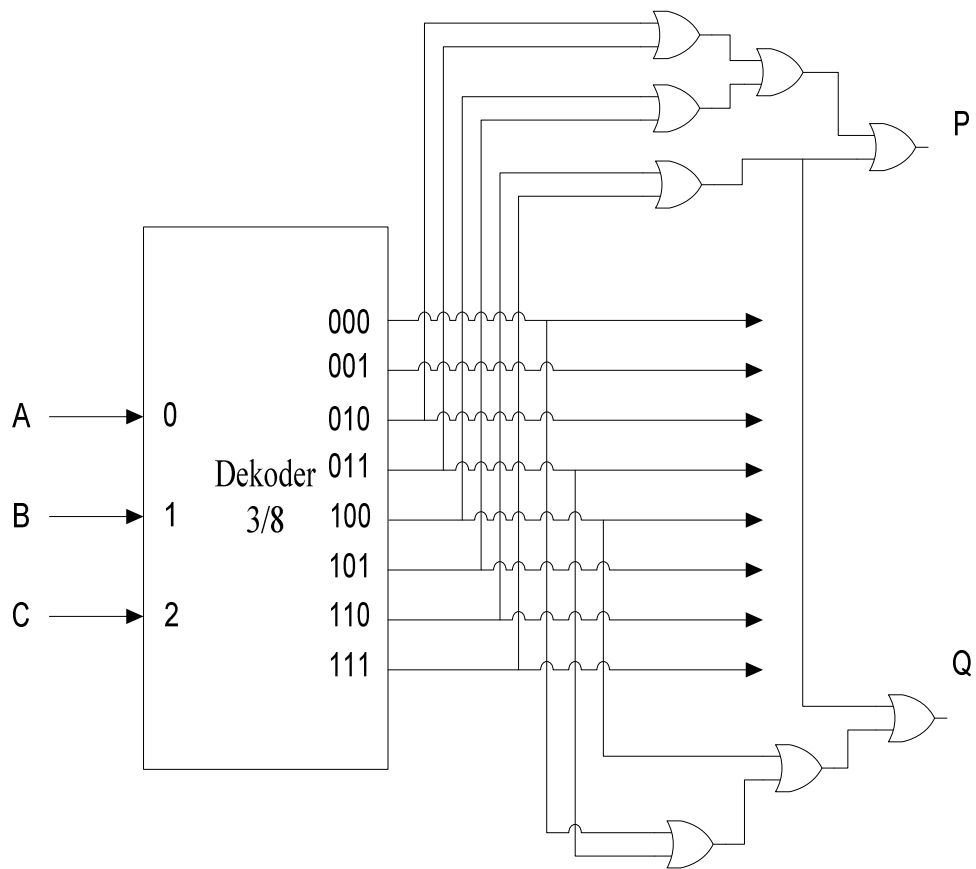
$$P = B + A$$

$$Q = BC + \overline{A}\overline{C} + \overline{B}\overline{C}$$

- Kako digitalno kolo ima tri ulaza potrebno je koristiti dekodera 3-8. Izrazi za P i Q se mogu napisati u složenijoj formi tako da figurišu sve tri promenljive (ili se vrednosti ulaza za koje su izlazi na logičkoj jedinici mogu očitati direktno iz tabele).

$$P = \overline{A}\overline{B}\overline{C} + \overline{A}B\overline{C} + A\overline{B}\overline{C} + A\overline{B}C + A\overline{B}C + ABC$$

$$Q = \overline{A}\overline{B}\overline{C} + \overline{A}B\overline{C} + \overline{A}\overline{B}C + A\overline{B}\overline{C} + ABC$$



3. [20] a) Korišćenjem potrebnog broja D flip-flopa i po jednog dvoulaznog NI, NILI, EXILI logičkog kola projektovati sekvencijalnu mrežu koja na svojim izlazima ponavlja sekvencu $S_3:S_2:S_1:S_0 \rightarrow 1000-1101-0101-0110-1000....$ Kao osnovu za projektovanje ove sekvencijalne mreže iskoristiti binarni brojač unapred. Traženo kolo realizovati isključivo primenom raspoloživih elemenata.

b) Napraviti realizaciju funkcije tražene u tački a) koristeći potreban broj flipflopova i po jedno NI i NILI kolo. Kao osnovu za projektovanje koristiti brojač u Grejovom kodu.

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.

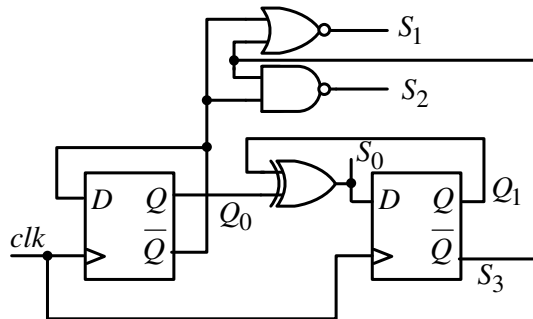
Q_1	Q_0	S_3	S_2	S_1	S_0
0	0	1	0	0	0
0	1	1	1	0	1
1	0	0	1	0	1
1	1	0	1	1	0

Tabela 1.

Očigledno je da važi:

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

Na slici je prikazana realizacija tražene mreže.



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.

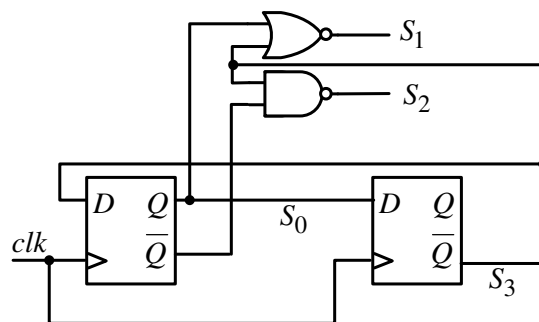
Q_1	Q_0	S_3	S_2	S_1	S_0
0	0	1	0	0	0
0	1	1	1	0	1
1	1	0	1	0	1
1	0	0	1	1	0

Tabela

Izlazni kod se u ovom slučaju dobija iz jednačina:

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

Na slici je prikazana je realizacija tražene mreže.



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;

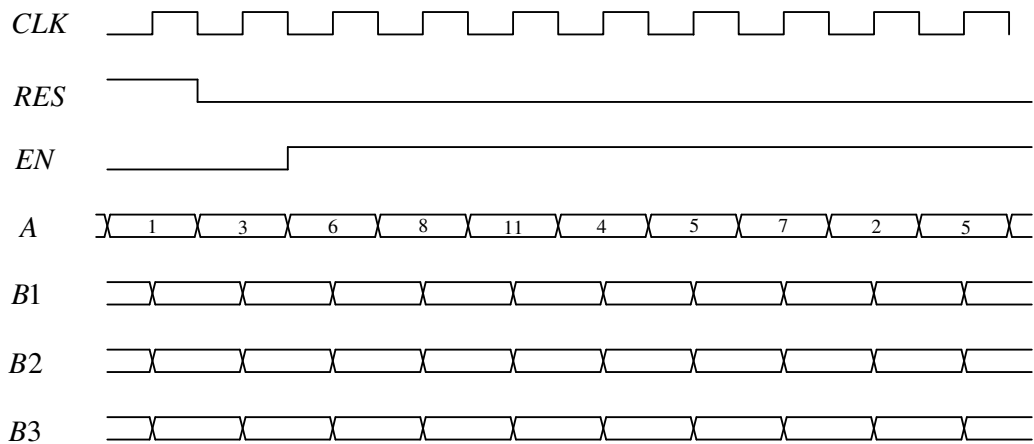
ENTITY seq IS PORT
(
  clk,res,en          : IN STD_LOGIC;
  a                   : IN STD_LOGIC_VECTOR(5 DOWNTO 0);
  b1                  : OUT STD_LOGIC_VECTOR(5 DOWNTO 0);
  b2                  : OUT STD_LOGIC_VECTOR(5 DOWNTO 0);
  b3                  : OUT STD_LOGIC_VECTOR(5 DOWNTO 0)
);
END seq;

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 (res='1') THEN
        current_state <= s0;
        b1 <= (OTHERS => '0');
        b2 <= (OTHERS => '0');
        b3 <= (OTHERS => '0');
        temp <= "000001";
      ELSIF (en='1') THEN
        CASE current_state IS
          WHEN s0 => b1 <= (OTHERS => '0');
                     b2 <= (OTHERS => '0');
                     b3 <= (OTHERS => '0');
                     temp <= "000010";
                     current_state <= s1;
          WHEN s1 => b1 <= STD_LOGIC_VECTOR(temp);
                     b2 <= a(4 DOWNTO 0) & a(0);
                     b3 <= a(5) & a(5 DOWNTO 1);
                     temp <= temp + UNSIGNED(a);
                     current_state <= s2;
          WHEN s2 => b1 <= a(3 DOWNTO 0) & a(1) & a(1);
                     b2 <= a;
                     b3 <= a(5) & a(4 DOWNTO 1) & a(2);
                     temp <= temp + 1;
                     IF (a<"000100") THEN
                       current_state <= s1;
                     ELSE
                       current_state <= s3;
                     END IF;
          WHEN s3 => b1 <= STD_LOGIC_VECTOR(temp);
                     b2 <= a(3 DOWNTO 0) & a(2 DOWNTO 1);
                     b3 <= a;
                     temp <= temp + UNSIGNED(a);
                     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 upisati ih na dati vremenski dijagram u **decimalnom** formatu.



Rešenje:

