

ETF U BEOGRADU, KATEDRA ZA ELEKTRONIKU

Predmet: OSNOVI DIGITALNE ELEKTRONIKE

OCENA _____

Ispit: 22.08.2014.

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. Na ulaze kombinacione mreže dolaze podaci D_1 i D_0 u digitalnoj formi. Osim ovih ulaza mreža ima i kontrolne digitalne ulaze C_1 , C_0 i izlaz Y. Izlaz Y zavisi od kontrolnih ulaza i od vrednosti podataka D_1 i D_0 na način kako je to prikazano u tabeli.

C_1	C_0	Y
0	0	$D_1 + D_0$
0	1	1
1	0	0
1	1	$D_0 \cdot D_1$

a) Ako su podaci D_1 i D_0 jednobitni, a operacije u tabeli pripadaju skupu operacija Bulove algebre odrediti:

a1) [2] Veličinu izlazne promenjive Y u bitima.

a2) [3] Korišćenjem minimalnog broja osnovnih logičkih kola realizovati kombinacionu mrežu koja ostvaruje funkciju iz tabele.

b) Ako su podaci D_1 i D_0 dvobitni, a operacije u tabeli predstavljaju standardne aritmetičke operacije odrediti:

b1) [5] Veličinu izlazne promenjive Y u bitima.

b2) [5] Korišćenjem multipleksera, sabirača i množača, nacrtati blok šemu koja realizuje opisanu višebitnu funkciju Y.

b3) [5] Upotrebom osnovnih logičkih kola realizovati multiplekser, sabirač i množač sa potrebnim brojem bita.

Rešenje:

a) U slučaju da su podaci jednobitni izlaz Y je takođe jednobitan i tablica istinitosti koja opisuje rad mreže je prikazana.

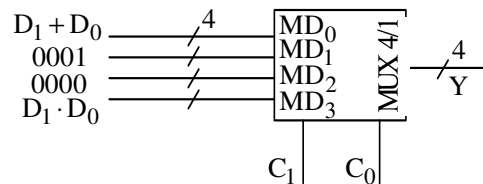
C_1	C_0	D_1	D_0	Y
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

Minimizacijom uz pomoć Karnoove karte dobija se Bulova funkcija za promenjivu Y.

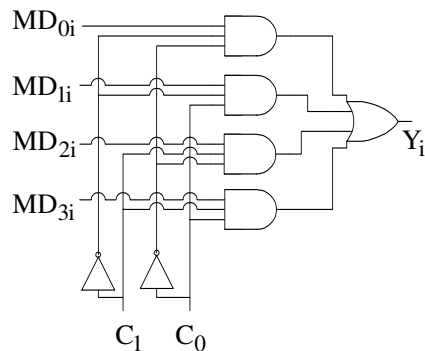
$$Y = \overline{C_1} C_0 + \overline{C_1} D_0 + \overline{C_1} D_1 + C_0 D_1 D_0$$

b) U slučaju da su podaci D_1 i D_0 dvobitni, izlaz Y će na osnovu tabele koja opisuje rad mreže u najgorem slučaju imati vrednost devet koja se može predstaviti sa minimalno četiri bita. Pošto u ovom slučaju postoji šest jednobitnih promenljivih koje utiču na izlaz Y , korišćenje Karnoovih karti bi bilo komplikovano pa se tražena mreža može realizovati u formi četvorokanalnog 4/1 multipleksera. Na ulaze ovog multipleksera ($MD_j, j=0, 1, 2, 3$) se dovode redom vrednosti $D_1 + D_0$, 0001 (vrednost 1), 0000 (vrednost 0), $D_0 \cdot D_1$, iz tabele koja opisuje rad mreže u zapisu sa četiri bita. Koja od ovih vrednosti će biti prosleđena na izlaz zavisi od vrednosti kontrolnih bita C_1 i C_0 .

Principska šema mreže je prikazana na slici.

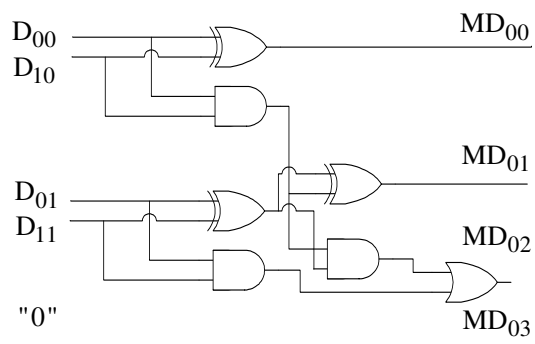


Četvorokanalni 4/1 multiplekser se može realizovati paralelizovanjem četiri jednokanalna 4/1 multipleksera za zajedničkim kontrolnim ulazima C_1 i C_0 . Realizacija jednog 4/1 multipleksera je prikazana na slici.

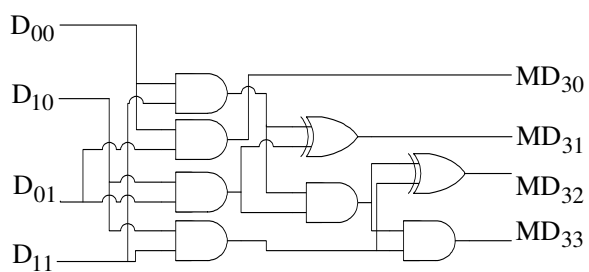


Gde i uzima vrednost $i=0, 1, 2, 3$ i označava pojedini bit iz skupa od četiri bita za sve promenjive (ulazne $MD_j, j=0, 1, 2, 3$ i izlaznu Y sa prethodne slike).

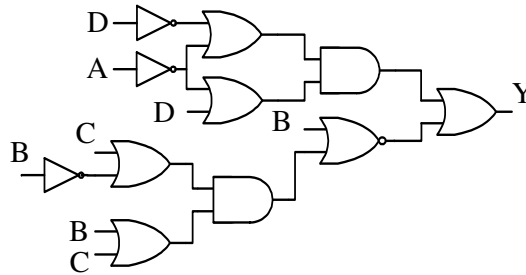
MD_0 se dobija kao izlaz dvobitnog sabirača, realizacija je prikazana na slici



MD₁ i MD₂ su fiksni i iznose 0001 (vrednost 1), 0000 (vrednost 0), dok se MD₃ dobija slično kao MD₀, samo je sada u pitanju izlaz dvobitnog množača čije je realizacija prikazana na slici.



2. a) [10] Koristeći NMOS i PMOS tranzistore, isprojektovati najprostije statičko CMOS kolo koje realizuje bulovu funkciju kao kolo sa slike.

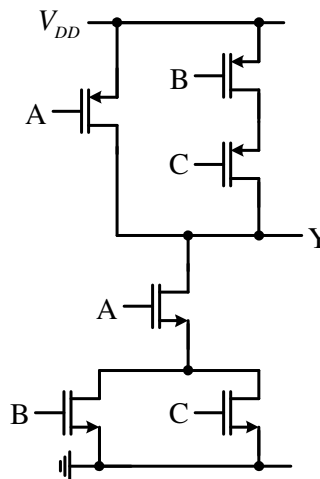


b) [10] Ukoliko je otpornost provodnih MOS tranzistora u prethodno realizovanoj strukturi $R_{on}=50\Omega$, odrediti maksimalnu struju koje realizovano CMOS kolo može obezbediti na izlazu u stanju logičke jedinice. Minimalni naponski nivo na izlazu CMOS kola koji se tumači kao logička jedinica je $V_{OH} = 3.3V$, a napon napajanja je $V_{CC} = 5V$.

Rešenje:

a) Kod CMOS logičkih kola, NILI funkcija se ostvaruje paralelnom vezom NMOS tranzistora i rednom vezom PMOS tranzistora, dok se NI funkcija ostvaruje rednom vezom NMOS tranzistora i paralelnom vezom PMOS tranzistora.

Funkcija koju treba realizovati je $Y=A(B+C)$, što se može dobiti primenom osnovnih operacija bulove algebre nad promenjivama koje učestvuju u formiranju izlaza Y. Statičko CMOS kolo koje realizuje ovu funkciju je prikazano na slici. Kako ulaz D ne utiče na vrednost izlazne promenjive Y on je izostavljen u realizaciji CMOS kola.

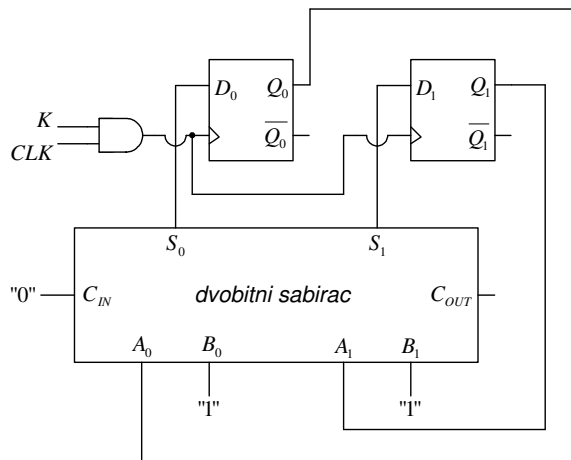


b) Najnepovoljniji slučaj u pogledu maksimalnog strujnog opterećenja CMOS kola u stanju logičke jedinice je kada su PMOS tranzistori kontrolisani sa B i C signalima uključeni a PMOS tranzistor kontrolisan signalom A isključen. Tada je, $I_{\max} = \frac{V_{CC} - V_{OH}}{2R_{ON}} = 17mA$ maksimalna

struja kojom se kolo sme opteretiti a da napon na izlazu ne padne ispod praga napona logičke jedinice.

3. a) [14] Za digitalno kolo na slici odrediti sekvencu stanja na izlazima flip-flopova Q_1Q_0 ako je vrednost kontrolnog signala $K=1$. Analizu početi od stanja u kome su oba flipflopa resetovana. Ulazi dvobitnog sabirača A_1A_0 i B_1B_0 predstavljaju ulaze za dvobitne sabirke, dok C_{IN} predstavlja ulaz na koji se dovodi bit ulaznog prenosa. Na izlazu C_{OUT} se generiše bit izlaznog prenosa, a na izlazima S_1S_0 se formira rezultat sabiranja. Koju funkciju obavlja dato kolo za $K=1$?

b) [6] Korišćenjem potrebnog broja digitalnih kola iz prethodne tačke kao i dodatnih logičkih kola, realizovati četvorobitni sinhroni brojač unazad.



Rešenje:

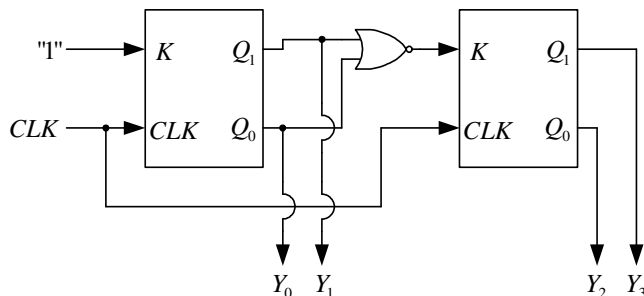
a) Analizom rada datog kola mogu se odrediti logički nivoi svih relevantnih signala u kolu. Pri analiziranju slučaja kada je kontrolni signal $K=1$, ako se krene od situacije u kojoj je stanje na izlazima flip-flopova $Q_1Q_0=00$, može se zaključiti da su logički nivoi na ulazima sabirača $C_{IN}=0$, $A_1A_0=00$, $B_1B_0=11$. Tada je rezultat sabiranja $S_1S_0=11$, $C_0=0$ (gde je C_0 prenos iz nultog u prvi razred sabirača), $C_{OUT}=0$. Nailaskom prve naredne uzlazne ivice signala takta stanje na izlazima flip-flopova postaje $Q_1Q_0=11$. Ovim i jedan od sabiraka promeni vrednost (i postane $A_1A_0=11$). S obzirom da je i dalje $C_{IN}=0$ kao i $B_1B_0=11$, zaključuje se da je $S_1S_0=10$, $C_0=1$ i $C_{OUT}=1$. Nailaskom prve naredne uzlazne ivice signala takta stanje na izlazima flip-flopova postaje $Q_1Q_0=10$.

Na ovaj način se može sprovesti kompletna analiza kola. Rezultati analize su prikazani u sledećoj tabeli:

C_{IN}	A_0	B_0	S_0	C_0	A_1	B_1	S_1	C_{OUT}	Q_1	Q_0
0	0	1	1	0	0	1	1	0	0	0
0	1	1	0	1	1	1	1	1	1	1
0	0	1	1	0	1	1	0	1	1	0
0	1	1	0	1	0	1	0	1	0	1

Iz rezultata analize se može zaključiti da za vrednost kontrolnog signala $K=1$ dato kolo obavlja funkciju dvobitnog sinhronog brojača unazad, koji menja stanja sinhrono sa uzlaznom ivicom signala takta CLK . Ukoliko bi bilo $K=0$, brojač bi zadržavao konstantnu binarnu vrednost jer signal takta ne bi dolazio na ulaze flip-flopova.

b) Realizacija traženog četvorobitnog brojača unazad je prikazana na sledećoj slici:



Korišćena su dva digitalna kola iz tačke **a**), pri čemu je svako od njih predstavljeno blokom sa ulazima K i CLK i izlazima Q_0 i Q_1 . Biti traženog četvorobitnog brojača su označeni sa $Y_3Y_2Y_1Y_0$. S obzirom da će dvobitni brojači iz prethodne tačke brojati unazad samo ako je $K = 1$, ulaz K za desni dvobitni brojač je iskorišćen za dovođenje signala dozvole brojanja samo u slučaju kada je na izlazu levog dvobitnog brojača $Q_1Q_0 = 00$ (i u tu svrhu se koristi dvoulazno NILI kolo). Ovim se obezbeđuje da dvobitni brojač koji predstavlja dva bita veće težine unutar četvorobitnog brojača, dekrementira svoju vrednost samo onda kada je susedni dvobitni brojač koji predstavlja dva bita manje težine u stanju $Q_1Q_0 = 00$.

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;

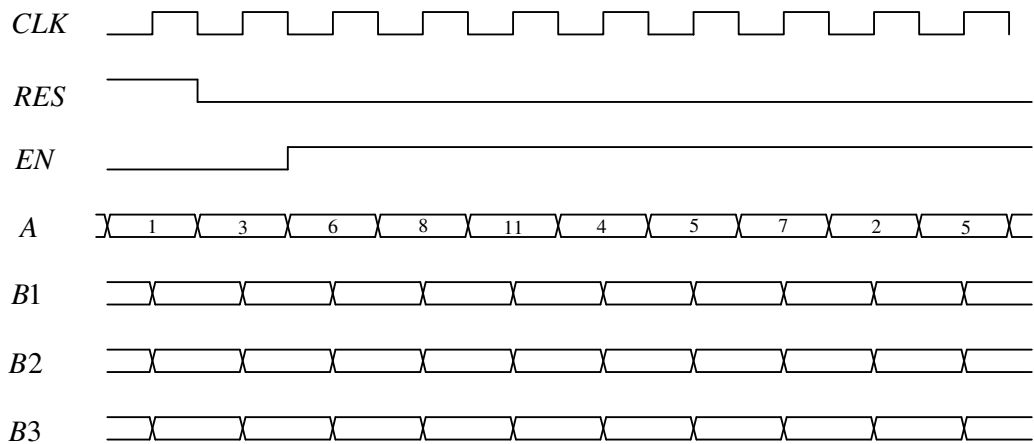
ENTITY seq IS PORT
(
  clk,res,en          : IN STD_LOGIC;
  a                   : IN STD_LOGIC_VECTOR(5 DOWNTO 0);
  b1                  : OUT STD_LOGIC_VECTOR(5 DOWNTO 0);
  b2                  : OUT STD_LOGIC_VECTOR(5 DOWNTO 0);
  b3                  : OUT STD_LOGIC_VECTOR(5 DOWNTO 0)
);
END seq;

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:

