

ETF U BEOGRADU, KATEDRA ZA ELEKTRONIKU

Predmet: OSNOVI DIGITALNE ELEKTRONIKE

OCENA _____

Ispit: 19.09.2008.

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

DEŽURNI:

KANDIDAT:

Sala _____
Vreme početka _____
Vreme završetka _____
Potpis _____

Ime _____
Prezime _____
Broj indeksa _____
Potpis _____

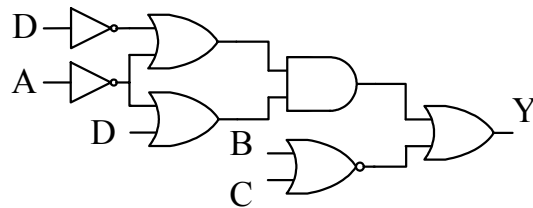
USLOVI ISPITA

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 kreiranja rešenja.
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					

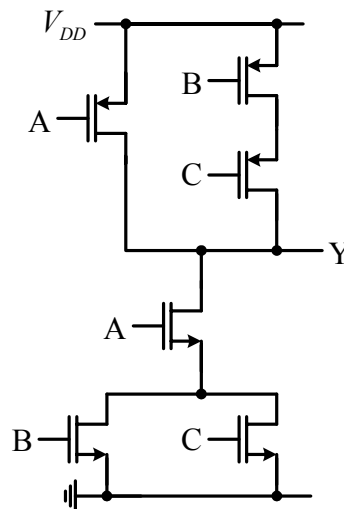
1. [20] Koristeći NMOS i PMOS tranzistore, isprojektovati najprostije statičko CMOS kolo koje realizuje bulovu funkciju kao kolo sa slike.



Rešenje:

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 $Y = \overline{A(B+C)}$, 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.



2. [20] Projektovati kombinacionu mrežu koja na svom izlazu generiše petobitni binarni broj Z kodovan u prirodnom binarnom kodu, a koji je određen izrazom $Z=\max(A,B,2C)$, gde su A , B i C četvorobitni binarni brojevi kodovani u prirodnom binarnom kodu, koji se dovode na ulaz kombinacione mreže. Na raspolaganju su četvorobitni komparatori, četverostruki multiplekseri 2/1 i potrebna logička kola.

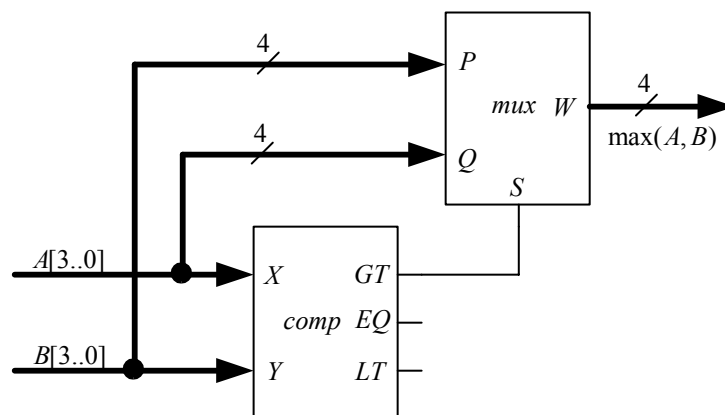
Četvorobitni komparator poredi dva četvorobitna binarna broja X i Y , i jedan od svoja tri izlazna signala postavlja na logičku jedinicu: a. izlaz $GT=L1$ ako je $X>Y$; b. izlaz $LT=L1$ ako je $X<Y$; c. izlaz $EQ=L1$ ako je $X=Y$. Kada je jedan od izlaza ravan $L1$, ostali izlazi su na $L0$.

Četverostruki multiplekser 2/1 poseduje četiri multipleksera 2/1 sa izlazom W , ulazima P i Q i selekcionim signalom S . Ukoliko je $S=0$, na izlazu multipleksera W se prosleđuje ulaz P , a ukoliko je $S=1$, na četvorobitni izlaz multipleksera W se prosleđuje ulaz Q .

Rešenje:

Obzirom da su na raspolaganju komparatori koji porede dva četvorobitna broja, zadatak se može rešiti u dva koraka. Najpre, treba izvršiti poređenje npr. četvorobitnih ulaznih brojeva A i B , zaključiti koji od ta dva broja je veći, a zatim uporediti taj veći broj sa dvostrukom vrednošću četvorobitnog broja C . Veći broj u tom drugom poređenju je ujedno i rezultat tražene operacije $Z=\max(A,B,2C)$.

Poređenjem brojeva A i B pomoću raspoloživog komparatora, može se dobiti signal koji će pokazivati da li je broj A veći od broja B . Taj signal se potom može iskoristiti za selekciju ulaza multipleksera koji će biti prosleđen na izlaz. Konkretno, pomenuta ideja je realizovana na sledećoj slici.



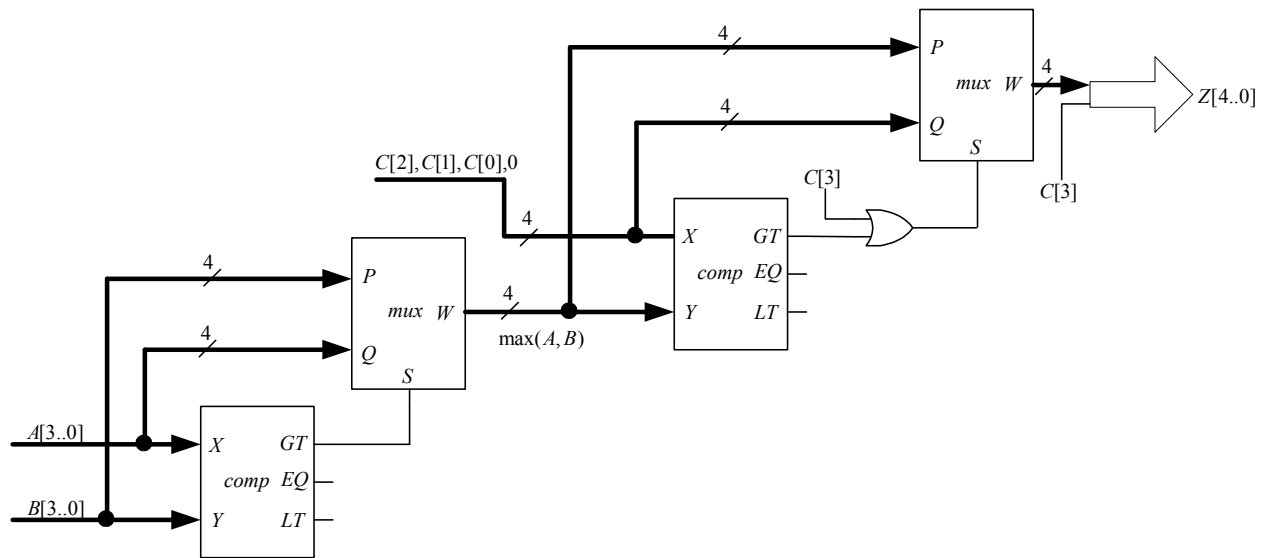
U slučaju da je ulazni broj $A[3..0]$ veći od drugog ulaznog broja $B[3..0]$, signal GT će biti na nivou logičke jedinice. To će omogućiti da se na izlaz multipleksera prosleđi Q ulaz multipleksera (tj. $A[3..0]$). U protivnom (ukoliko ulazni broj $A[3..0]$ nije veći od drugog ulaznog broja $B[3..0]$) signal GT će biti na nivou logičke nule. To će omogućiti da se na izlaz multipleksera prosleđi P ulaz multipleksera (tj. $B[3..0]$). Dakle, u svakom slučaju, na izlazu multipleksera će se naći maksimalna od dve ulazne vrednosti $A[3..0]$ i $B[3..0]$.

Tu vrednost dalje je potrebno uporediti sa dvostrukom vrednošću četvorobitnog broja $C[3..0]$. Ova dvostruka vrednost se može dobiti pomeranjem četvorobitnog broja $C[3..0]$ za jedno mesto u levo, pri čemu će pozicija bita najmanje težine biti popunjena logičkom nulom. Međutim, pošto je rezultat tog množenja sa 2 dobijanje petobitnog broja u opštem slučaju, jasno je da se svih pet bita ne može dovesti na ulaz četvorobitnog komparatora kako bi se na isti način kao u prvom koraku odredilo da li je veće $\max(A,B)$ ili $2C$.

Stoga se može primeniti sledeći pristup. Ukoliko je $C[3]=1$, jasno je da je $2C$ veće od $\max(A,B)$. U protivnom, ukoliko je $C[3]=0$, tada se može iskoristiti četvorobitni komparator za

utvrđivanje da li je $2C$ veće od $\max(A,B)$ i to poređenjem 4 bita najmanje težine broja $2C$ sa brojem $\max(A,B)$.

Opisani koncept je primenjen u rešenju koje je prikazano na sledećoj slici.



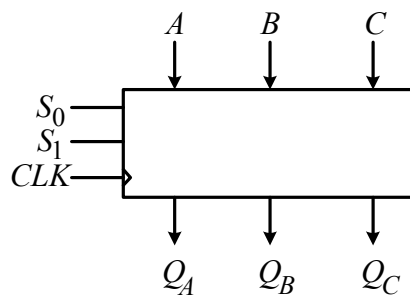
Četiri bita najmanje težine broja $2C$ treba da budu prosleđena na izlaz drugog multipleksera ukoliko je $C[3]=1$ ili ukoliko ta 4 bita predstavljaju binarni broj koji je veći od $\max(A,B)$. To se postiže dovođenjem ili signala koji pokazuje da četiri bita najmanje težine broja $2C$ predstavljaju binarni broj koji je veći od $\max(A,B)$ ili signala $C[3]$ na selekcioni ulaz multipleksera (u tu svrhu je korišćeno ILI kolo). Četvorobitnom rezultatu na izlazu drugog multipleksera je kao peti bit (najveće težine) potrebno pridružiti i bit $C[3]$, jer ukoliko je na izlaz prosleđen broj $\max(A,B)$ taj peti bit će biti sigurno $C[3]=0$ (pa kao takav neće promeniti vrednost rezultata), a ukoliko su na izlaz prosleđena četiri bita najmanje težine broja $2C$, dodavanjem i bita $C[3]$ će se samo kompletirati broj $2C$ koji bi u tom slučaju bio traženi rezultat.

3. [20] Realizovati što jednostavniju sinhronu sekvencijalnu mrežu prikazanu na slici tako da za navedene vrednosti upravljačkih signala S_1S_0 mreža radi u sledećim režimima:

- za $S_1S_0 = 00$ flip-flopovi ne menjaju stanje bez obzira na taktni impuls CLK
- za $S_1S_0 = 01$ mreža radi kao trobitni binarni brojač unazad
- za $S_1S_0 = 10$ u flip-flopove se upisuje sadržaj ABC nakon uzlazne ivice CLK
- za $S_1S_0 = 11$ uzlazna ivica CLK postavlja $Q_A=Q_B=Q_C=0$

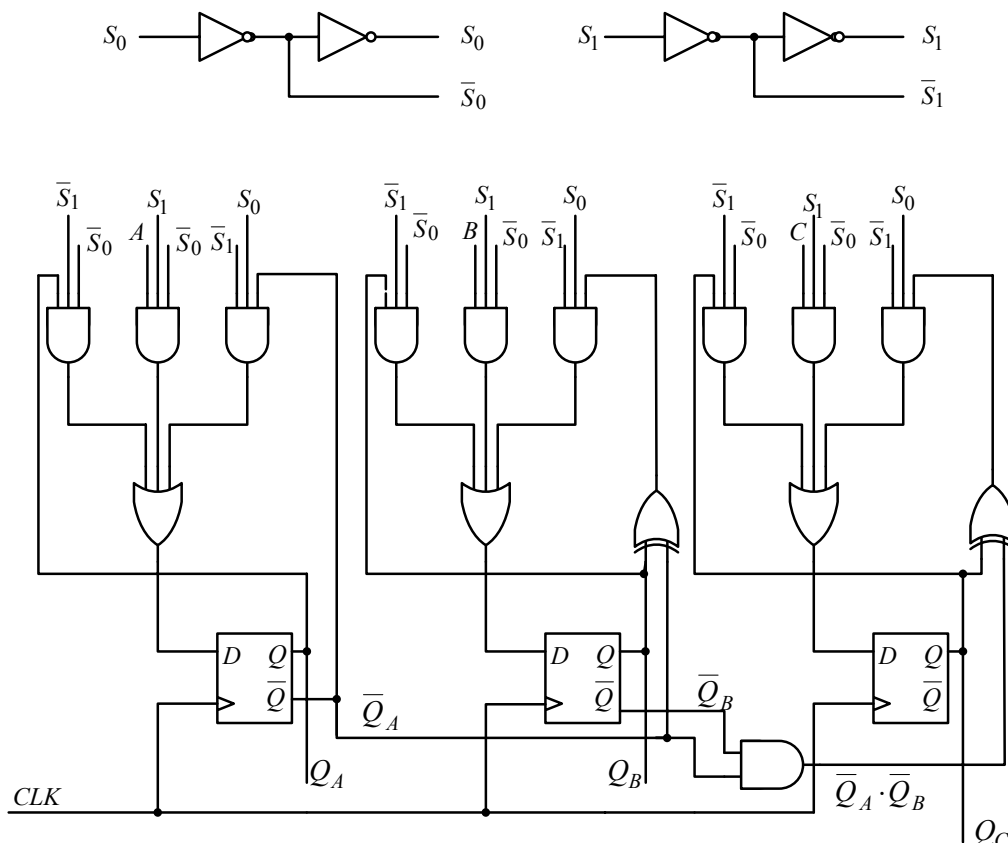
Na raspolaganju su ivični D flip-flopovi koji se okidaju na uzlaznu ivicu, EXILI kola, I kola, ILI kola i invertori.

NAPOMENA: Nacrtati prvo rešenje za registar sa paralelnim upisom i čitanjem za tri bita. Zatim sintetisati trobitni binarni brojač unazad i nacrtati ga na posebnoj slici. Prethodne dve slike spojiti u jednu konstrukcijom kombinacione mreže koja kontroliše vrednost D ulaza flipflopa i zatim na kombinacionoj mreži dograditi kontrole za blokiranje i za reset flipflopova.



Rešenje:

Analizom narednog stanja sekvencijalne mreže u odnosu na sve moguće kombinacije upravljačkih signala za svako postojeće stanje, i korišćenjem karnoovih karata u procesu minimizacije dobija se tražena realizacija kao na slici.



4. [20] Koristeći metod dizajniranja na bazi opisa ponašanja sistema, napraviti VHDL model aritmetičke jedinice prema sledećim specifikacijama.

Kao ulazni podatak u aritmetičku jedinicu se dovodi petobitni binarni broj `in1` kodovan u prirodnom binarnom kodu. Na svom izlazu aritmetička jedinica treba da generiše rezultat `result` sinhrono sa uzlaznom ivicom signala takta `clk` i to u zavisnosti od dvobitnog kontrolnog signala `sel` na sledeći način:

- ukoliko je kontrolni signal `sel="00"`, rezultat treba da bude 0 (tj. izlaz aritmetičke jedinice treba da se resetuje);
- ukoliko je kontrolni signal `sel="01"`, rezultat treba da bude za 50 veći od ulaznog podatka;
- ukoliko je kontrolni signal `sel="10"`, rezultat treba da bude 4 puta veći od ulaznog podatka;
- ukoliko je kontrolni signal `sel="11"`, rezultat treba da bude celobrojni količnik deljenja ulaznog podatka sa 4.

Odrediti potreban broj bita u izlaznom vektoru tako da pri opisanim računskim operacijama ne dođe do prekoračenja.

Biti svih vektora su poređani od MSB ka LSB u smeru sa leva na desno.

Napomena: Ocenjuje se precizna upotreba sintakse.

Rešenje:

Da ne bi došlo do prekoračenja prilikom sprovođenja opisanih računskih operacija, neophodno je da izlazni vektor bude sedmobitni, jer u najkritičnijem slučaju (kada se najveći petobitni binarni broj množi sa 4) sedam bita je dovoljno za korektno predstavljanje rezultata.

```

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

ENTITY arith_unit IS PORT
(
    in1      : IN   STD_LOGIC_VECTOR(4 DOWNTO 0);
    sel      : IN   STD_LOGIC_VECTOR(1 DOWNTO 0);
    clk      : IN   STD_LOGIC;
    result   : OUT  STD_LOGIC_VECTOR(6 DOWNTO 0)
);
END arith_unit;

ARCHITECTURE behav OF arith_unit IS
    SIGNAL tmp : UNSIGNED(6 DOWNTO 0);
BEGIN
    PROCESS (clk) BEGIN
        IF (rising_edge(clk)) THEN
            IF (sel="00") THEN
                tmp <= (OTHERS => '0');
            ELSIF (sel="01") THEN
                tmp <= UNSIGNED("00" & in1) + 50;
            ELSIF (sel="10") THEN
                tmp <= UNSIGNED(in1(4 DOWNTO 0) & "00");
            ELSIF (sel="11") THEN
                tmp <= UNSIGNED("0000" & in1(4 DOWNTO 2));
            END IF;
        END IF;
    END PROCESS;
    result <= STD_LOGIC_VECTOR(tmp);
END behav;

```