

ETF U BEOGRADU, KATEDRA ZA ELEKTRONIKU

Predmet: OSNOVI DIGITALNE ELEKTRONIKE

OCENA _____

Ispit: 08.02.2012.

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

DEŽURNI:

KANDIDAT:

Sala _____
Vreme početka _____
Vreme završetka _____
Potpis _____

Ime _____
Prezime _____
Broj indeksa _____
Potpis _____

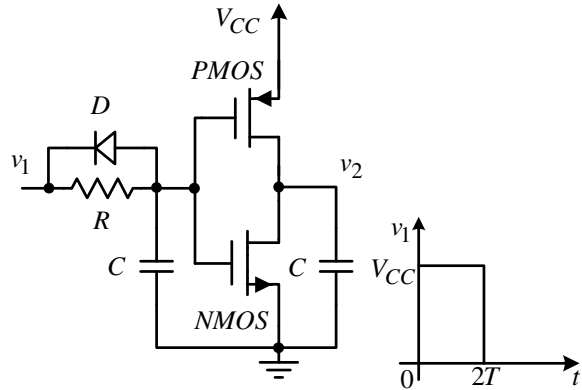
USLOVI ISPITA

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] Na ulaz kola sa slike se dovodi napon v_1 čiji je vremenski dijagram prikazan. Odrediti i nacrtati talasni oblik napona v_2 u vremenskom intervalu $-\infty < t < +\infty$. Smatrati da je u provodnom stanju otpornost NMOS tranzistora r_{nmos} konstantna i iznosi $r_{nmos} = R$, dok u neprovodnom stanju NMOS tranzistor ima beskonačnu otpornost, otpornost PMOS tranzistora u provodnom stanju r_{pmos} je konstantna i iznosi $r_{pmos} = R$, dok u neprovodnom stanju PMOS transistor ima



beskonačnu otpornost. ($\frac{V_{CC}}{2} = |V_T|$, $|V_T|$ – napon uključenja NMOS, PMOS tranzistora; napon provođenja diode $V_D = \frac{V_{CC}}{10}$ (dioda je idealna); $RC = T$).

Rešenje:

Do momenta promene napona v_1 , kondenzator na ulazu CMOS invertora je razelektrisan tako da je PMOS tranzistor uključen a NMOS tranzistor isključen. Napon v_2 ima vrednost $v_2(0^-) = V_{CC}$. Promenom napona v_1 počinje punjenje kondenzatora na ulazu CMOS invertora (napon počinje da raste). Promena napona kondenzatora na ulazu je opisana relacijom

$$v_C(t) = V_{CC} - V_{CC} e^{-\frac{t}{CR}}$$

Napon v_2 , zadržava vrednost $v_2 = V_{CC}$ sve dok napon v_C ne postane $v_C(t_1) = \frac{V_{CC}}{2}$, kada se isključuje PMOS tranzistor i uključuje NMOS tranzistor. Vremenski trenutak t_1 se može odrediti iz

$$v_C(t_1) = \frac{V_{CC}}{2} = V_{CC} - V_{CC} e^{-\frac{t_1}{CR}} \Rightarrow t_1 = CR \ln 2 = T \ln 2 \approx 0.7T.$$

Od tog momenta, kondenzator na izlazu se prazni i talasni oblik napona je opisan relacijom

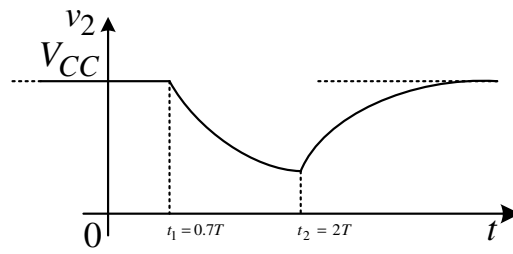
$$v_2(t) = V_{CC} e^{-\frac{t-t_1}{CR}}.$$

Kada se napon na ulazu u momentu

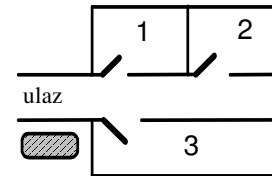
$t_2 = 2T$, naglo promeni, kroz diodu će proteći veća struja kojom se kondenzator momentalno prazni na napon $v_C(t_2) = V_D$ što dovodi do ponovnog uključenja PMOS tranzistora i isključenja NMOS tranzistora. U nastavku, kondenzator na izlazu se puni prema relaciji

$$v_2(t) = V_{CC} - (V_{CC} - v_2(t_2)) e^{-\frac{t-t_2}{CR}}, \text{ pri čemu je napon } v_2(t_2) = V_{CC} e^{-\frac{t_2-t_1}{CR}} \approx 0.27V_{CC}.$$

Vremenski oblik napona $v_2(t)$ je prikazan na slici



2. [20] Jedna banka poseduje nekoliko sefova za smeštanje novca komitenata i raznih poverljivih dokumenata banke. Sefovi su raspoređeni u tri prostorije kao na slici. Zaposleni u banci su podeljeni u 7 kategorija . Na ulazu u grupu prostorija sa sefovima svaki zaposleni iz određene kategorije može da ubaci karticu u automat koji otvara odgovarajuća vrata prema sledećoj tabeli (kartice generišu višebitni paralelni kod koji označava kategoriju zaposlenih):



Kategorija	Otvorena vrata		
Direktor banke	S_1	S_2	S_3
Zamenik direktora	S_1	S_2	
Sektor pravnih poslova	S_1		S_3
Sektor za naplatu		S_2	S_3
Sektor za marketing	S_1	S_2	
Sektor za kreditiranje		S_2	
Sektor za platni promet	S_1		S_3

Projektovati minimalnu kombinacionu mrežu koja zavisno od usvojenog koda otvara vrata prostorija sa sefovima i dozvoljava pristup raznim kategorijama zaposlenih u zavisnosti od tabele. U realizaciji koristiti standardna logička kola. Predvideti rešenje za neovlašćen pristup (kartica nije u sistemu).

Rešenje:

Kategorija	Kod kategorije $X Y Z$	Otvorena vrata $S_1 S_2 S_3$
Neovlašćeni	0 0 0	0 0 0
Direktor banke	0 0 1	1 1 1
Zamenik direktora	0 1 0	1 1 0
Sektor pravnih poslova	0 1 1	1 0 1
Sektor za naplatu	1 0 0	0 1 1
Sektor za marketing	1 0 1	1 1 0
Sektor za kreditiranje	1 1 0	0 1 0
Sektor za platni promet	1 1 1	1 0 1

Jedan od mogućih načina za kodiranje odgovarajuće grupe zaposlenih prilikom pristupa automatu za otvaranje vrata je prikazan u tabeli. Kombinacija 000 je usvojena u slučaju da neko neovlašćeno lice sa lažnom karticom (ili koja je nevažeća iz nekog razloga, osoba više nije zaposlena u firmi itd...) pokuša da pristupi prostoriji sa sefovima.

Minimizacijom se dobija:

$$S_1 = Z + \overline{X}Y$$

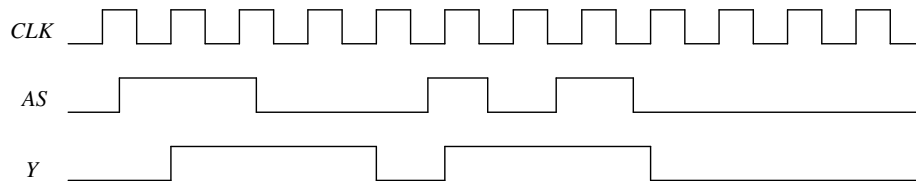
$$S_2 = \overline{Y}Z + Y\overline{Z} + X\overline{Y}$$

$$S_3 = \overline{X}Z + YZ + X\overline{Y}\overline{Z}$$

Drugo rešenje za neovlašćen pristup bio bi da automat za čitanje kartica ne generiše nikakav kod prilikom ubacivanja nevažeće kartice (kao da kartica nije ni ubačena), u tom slučaju u koloni za kod otvaranja rampi, umesto koda „0 0 0“ može da stoji „x x x“, što bi znatno uprostilo dobijene izraze.

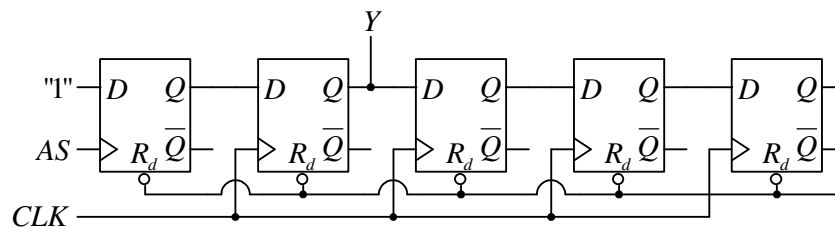
3. [20] Realizovati sekvencijalno kolo koje vrši detekciju pojave uzlazne ivice asinhronog signala AS. Nakon pojave uzlazne ivice signala AS, sekvencijalno kolo treba da generiše impuls Y, trajanja 3 periode takta CLK, sinhrono sa uzlaznom ivicom signala takta CLK. Uzlazna ivica signala Y treba da bude sinhrona sa prvom uzlaznom ivicom signala takta CLK koja se pojavi nakon uzlazne ivice signala AS. Sekvencijalno kolo ne treba da registruje novu uzlaznu ivicu signala AS ako se ona pojavi za vreme dok je signal Y na nivou logičke jedinice. Smatrati da je trajanje asinhronog signala AS (dok je na nivou logičke jedinice) duže od 1 periode takta signala CLK.

Na raspolaganju je isključivo minimalan potreban broj D flip-flopova sa ulazom za asinhroni reset. Primer vremenskih dijagrama odgovarajućih signala koji opisuju rad sekvencijalne mreže je dat na slici.



Rešenje:

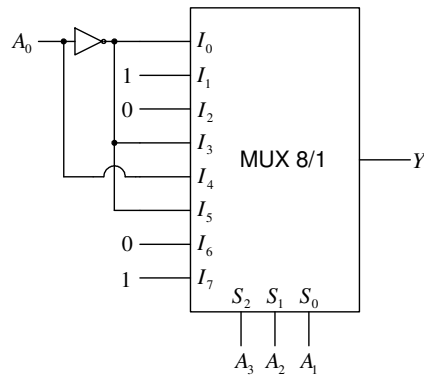
Traženo rešenje je prikazano na sledećoj slici.



Nakon pojave uzlazne ivice asinhronog signala AS, izlaz prvog D flip-flopa (krajnjeg levog) postaje „1“. Prva nastupajuća uzlazna ivica signala takta CLK postavlja izlaz drugog flip-flopa (koji je ujedno i izlazni signal Y) na nivo logičke jedinice. Svaka od tri naredne uzlazne ivice signala takta CLK uzrokuje propagiranje logičke jedinice kroz svaki naredni D flip-flop. Za to vreme je signal Y i dalje na nivou logičke jedinice. U trenutku kada izlaz Q petog D flip-flopa postane „1“, preko ulaza za asinhroni reset će se resetovati stanja svih D flip-flopova i time će i izlaz Y pasti na nivo logičke nule, čime će biti obezbeđeno da impuls na izlazu Y traje tačno 3 periode takta CLK. Takođe, eventualna pojava nove uzlazne ivice signala AS za vreme dok je Y na nivou logičke jedinice neće uzrokovati pojavu novog impulsa na izlazu Y.

4. [20] Na slici je prikazana kombinaciona mreža realizovana pomoću multipleksera 8/1. Koristeći metod dizajniranja na bazi opisa ponašanja sistema, napraviti VHDL model kombinacionog kola, sa jednim ulaznim četvorobitnim vektorom i jednim izlaznim jednobitnim signalom, koje realizuje istu prekidačku funkciju kao i data kombinaciona mreža.

Napomena: Ocenjuje se precizna upotreba sintakse.



Rešenje:

Analizom rada date kombinacione mreže može se zaključiti da odgovarajuća kombinaciona tabela ima sledeći izgled:

A_3	A_2	A_1	A_0	Y
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	1
0	1	1	1	0
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	1
1	1	1	1	1

Traženi VHDL kod je:

```
library IEEE;
use IEEE.std_logic_1164.all;

entity comb is
    port (A: in STD_LOGIC_VECTOR (3 downto 0);
          Y: out STD_LOGIC);
end comb;

architecture behav of comb is
begin
```

```
process (A)
begin
  case A is
    when "0000" => Y <= '1';
    when "0001" => Y <= '0';
    when "0010" => Y <= '1';
    when "0011" => Y <= '1';
    when "0100" => Y <= '0';
    when "0101" => Y <= '0';
    when "0110" => Y <= '1';
    when "0111" => Y <= '0';
    when "1000" => Y <= '0';
    when "1001" => Y <= '1';
    when "1010" => Y <= '1';
    when "1011" => Y <= '0';
    when "1100" => Y <= '0';
    when "1101" => Y <= '0';
    when "1110" => Y <= '1';
    when "1111" => Y <= '1';
    when others => Y <= '0';
  end case;
end process;
end behav;
```