

**UNIVERZITET U BEOGRADU – ELEKTROTEHNIČKI FAKULTET  
KATEDRA ZA ELEKTRONIKU**

# **DIGITALNA ELEKTRONIKA 1**

*Materijali za računске vežbe*

**SINTEZA KOMBINACIONIH MREŽA POMOĆU KOLA SREDNJEG  
STEPENA INTEGRACIJE**

**Vežbe 2**

**Pripremio:**

**Haris Turkmanović ( [haris@etf.bg.ac.rs](mailto:haris@etf.bg.ac.rs) )**

**Beograd 2022**

# Sadržaj

1. Uvod .....	3
2. Zadaci sa časova vežbi .....	4
Zadatak 2.1.....	4
Zadatak 2.2.....	6
Zadatak 2.3.....	8
Zadatak 2.4.....	11
3. Zadaci za samostalni rad.....	22
Zadatak 3.1.....	22
Zadatak 3.2.....	22

# **1. Uvod**

## 2. Zadaci sa časova vežbi

### Zadatak 2.1.

a) Pomoću multipleksera 4/1 i NI, ILI i I logičkih kola realizovati kombinacionu mrežu koja generiše izlaze  $Y_0$  i  $Y_1$  definisane logičkih funkcijama predstavljenim u tabeli 2.1.1.

Tabela 2.1.1. - Tabela logičkih funkcija za zadata 2.1.

$C_1C_0$	$Y_1$	$Y_0$
00	$\overline{A \oplus B}$	$A \oplus B$
01	$A \oplus B$	$\overline{A \oplus B}$
10	$\overline{AB}$	$\overline{AB}$
11	$AB$	$A\overline{B}$

b) Predstaviti logičku šemu multipleksera 4/1 korišćenog u tački a)

---

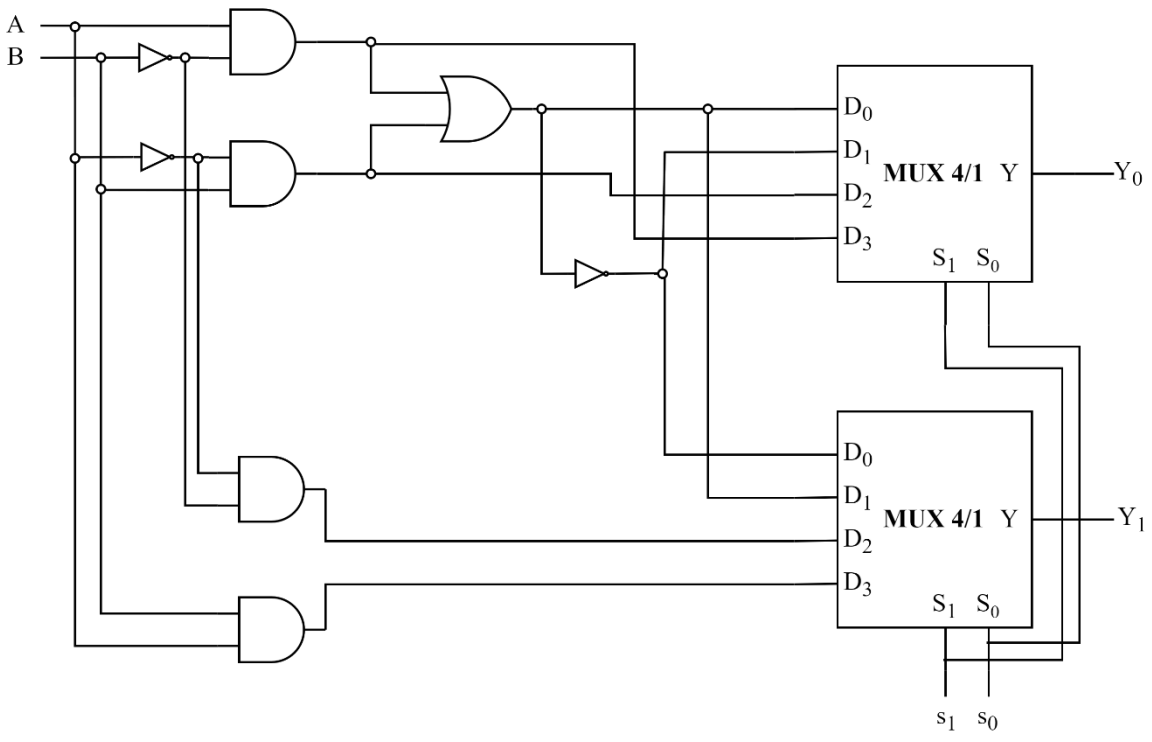
### Rešenje:

a) Izlaze kombinacione mreže realizovaćemo korišćenjem dva multipleksera 4/1. Ukoliko funkcije koje se dovode na ulaz multipleksera nisu izražene u formi pogodnoj za realizaciju I, ILI ili NI logičkim kolima, najpre je neophodno izvesti njihovu odgovarajuću formu. Za slučaj  $\overline{A \oplus B}$  i  $A \oplus B$  dobijamo ekvivalentne izraze (2.1.2.) i (2.1.2.).

$$A \oplus B = B\overline{A} + \overline{B}A \quad (2.1.1.)$$

$$\overline{A \oplus B} = \overline{B\overline{A} + \overline{B}A} \quad (2.1.2.)$$

Nakon izvođenja ekvivalentnih izraza možemo nacrtati kombinacionu mrežu čiji izlazi su definisani funkcijama predstavljenim u Tabeli 3. Ta kombinaciona mreža je predstavljena na slici 2.1.1.

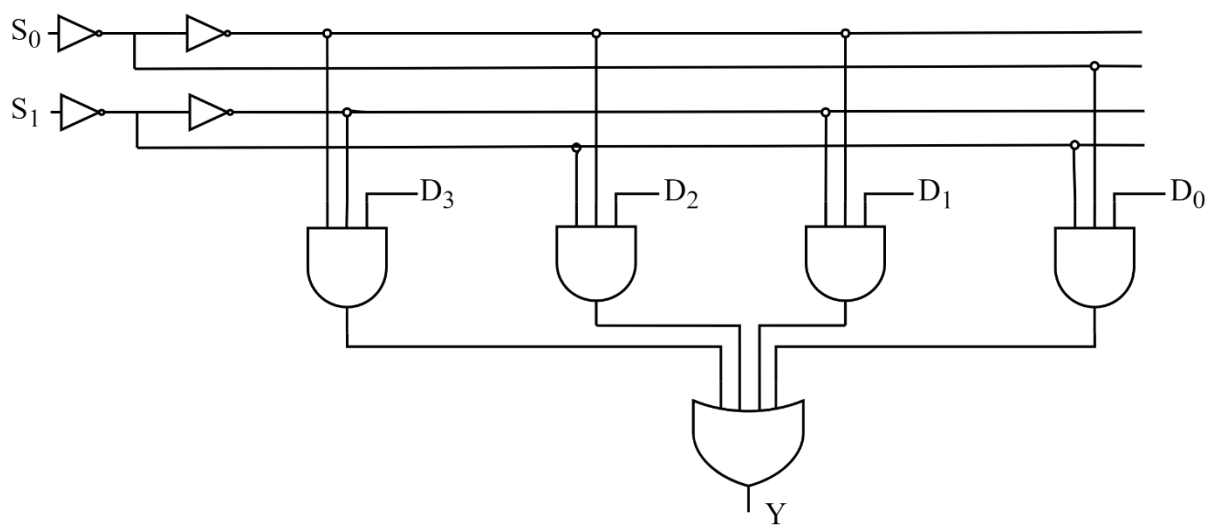


Slika 2.1.1. - Realizacija kombinacione mreže za zadataka 1-a)

b) Izlaz Y kombinacione mreže mutlipleksera 4/1, korišćenog za realizaciju u tački a) je definisan sledećom funkcijom:

$$Y = \bar{S}_1\bar{S}_0D_0 + \bar{S}_1S_0D_1 + S_1\bar{S}_0D_2 + S_1S_0D_3 \quad (2.1.3)$$

Na osnovu ove funkcije dobijamo realizaciju prikazanu na slici 2.1.2:



Slika 1.1.2. - Implementacija multipleksera 4/1

## Zadatak 2.2.

a) Pomoću jednog multipleksera 4/1 i potrebnog broja logičkih kola projektovati kolo koje generiše izlazni signal Y definisan logičkom funkcijom:

$$Y = CB\bar{A} + CA + \bar{B}A \quad (2.2.1)$$

- b) Projektovati kolo u minimalnoj formi ako su dostupna isključivo dvoulazna NI kola.  
c) Uporediti realizaciju pod a) i pod b) u pogledu složenosti i kašnjenja ako je vreme propagacije signala kroz multiplekser  $t_{d-max}=20ns$  a kroz logička kola  $t_{dl} = 10ns$ .  
d) Da li se u realizaciji pod b) javljaju lažne nule, i ukoliko se javljaju jasno naznačiti pri kojim prelazima. Predstaviti realizaciju, u vidu logičke funkcije, kojom se otklanjanju lažne nule.
- 

### Rešenje:

a) Prilikom realizacije logičke funkcije koja podrazumeva korišćenje isključivo multipleksera, za signale na selekcionim linijama  $S_1S_0$  (u slučaju multipleksera 4/1) treba postaviti one ulaze koji su u logičkoj funkciji zastupljeni sa najviše različitih kombinacija. U slučaju funkcije (2.2.1) možemo primetiti da su AB i AC podjednako zastupljeni i u nastavku ćemo za selekzione ulaze uzeti AB.

Logičku funkciju odgovarajućeg ulaza  $D_i$  multipleksera 4/1 određujemo primenjujući proceduru koja podrazumeva zamenu promenljivih AB u jednačini:

$$Y(A, B, C) = CB\bar{A} + CA + \bar{B}A \quad (2.2.2)$$

sa binarnim kombinacijama od 00 do 11

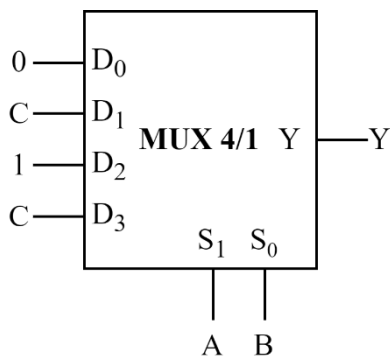
$$Y(0, 0, C) = 0 \quad (2.2.3)$$

$$Y(0, 1, C) = C \quad (2.2.4)$$

$$Y(1, 0, C) = 1 \quad (2.2.5)$$

$$Y(1, 1, C) = C \quad (2.2.6)$$

Na osnovu niza jednakosti od (2.2.3.) do (2.2.6.) dobijamo realizaciju prikazanu na slici 2.2.1.



Slika 2.2.1. – Realizacija funkcije (2.4) korišćenjem isključivo multipleksera 4/1

**Za samostalni rad:** Implementirati logičku funkciju Y ukoliko su ako selekcionni signali uzeti ulazi AC

b) Kao i do sada, prvi korak na putu do minimalne realizacije podrazumeva popunjavanje odgovarajuće Karnoove karte na osnovu koje se izvodi funkcija u minimalnoj formi. Sadržaj Karnoove karte za funkciju (2.2.1) je prikazan na slici 2.2.2

	A		
CB		0	1
00		0	1
01		0	0
11		1	1
10		0	1

Slika 2.2.2 – Sadržaj karnoove karte za funkciju (2.2.1.)

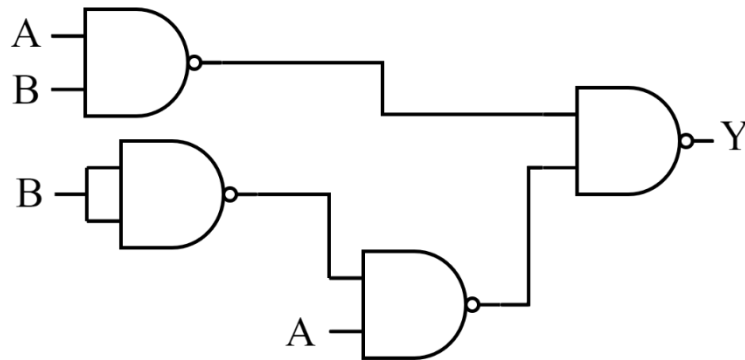
Pošto se zahteva minimalna realizacija korišćenjem isključivo dvoulaznih NI kola, u Karnoovoj karti tražimo oblasti koje sadrže logičke jedinice, tj. glavni cilj je realizacija funkcije (2.2.1) u formi zbira proizvoda. Na osnovu Karnoove karte sa slike 2.2.2 , dobijamo sledeću funkciju:

$$Y = CB + \bar{B}A \quad (2.2.7)$$

Primenom pravila Bulove algebre  $Y = \bar{\bar{Y}}$ , dobijamo:

$$Y = \overline{\overline{CB} \cdot \overline{\bar{B}A}} \quad (2.2.8)$$

Na osnovu (2.2.8) moguće je realizovati logičku funkciju Y kao što je prikazano na slici 2.2.3



Slika 2.2.3 - Realizacija logičke funkcije 2.4 u minimalnoj formi

- c) Kako se u standardnom 14-pinskom integrisanom kolu mogu naći 4 dvoulazna NI kola, potreban broj integrisanih kola za realizaciju podpredstavljenu u tački a) i b) je isti i znosi 1. Međutim, sa strane kašnjenja postoje izvesne razlike. U slučaju realizacije iz tačke a), kašnjenje iznosi  $t_{dy}$  dok u slučaju realizacije pod b) kašnjenje iznosi  $t_{dbY} = 30ns$ . Takođe, realizacije pod a) ne zahteva dodatnu minimizaciju date funkcije obzirom da se vrednosti ulaza multipleksra određuju na osnovu konkretnih vrednosti ulaza A i B.
- d) Prelaz pri kojem može doći do pojave lažne nule je označen na slici 2.2.4

	A		
CB		0	1
00		0	1
01		0	0
11		1	1
10		0	1

Slika 2 – Sadržaj karnoove karte za funkciju (2.2.4) sa naznačenim prelazom koji dovodi do pojave lažne nule

U cilju sprečavanja pojave lažne nule, uvešćemo dodatnu oblast kojom ćemo preklopiti dve postojeće oblasti. Nakon toga, izvodimo funkciju:

$$Y = CB + \bar{B}A + CA \quad (2.2.9)$$

### Zadatak 2.3.

- a) Izvesti funkciju svakog izlaza dekodera 3/8. Izlazi dekodera su aktivni u logičkoj jedinici i dekodер ne sadrži enable ulaze
- b) Pomoću dekodera 3/8 sa enable ulazom  $EN_1$  aktivnim na nivou logičke jedinice i  $EN_2$  i  $EN_3$  aktivnim na nivou logičke nule, projektovati dekodер 4/16. Izlazi dekodera 3/8 su aktivni na nivou logičke nule, dok su ostali ulazi (osim  $EN_2$  u  $EN_3$ ) aktivni na nivou logičke jedinice. Izlazi dekodera 4/16 takođe treba da budu aktivni na nivou logičke nule.
- c) Pomoću dekodera 3/8 iz prethodne tačke, i dvoulaznih I logičkih kola, realizovati sledeću funkciju:

$$Y = \bar{B}A + C\bar{B}\bar{A} + \bar{C}BA \quad (2.3.1)$$



d) Pomoću dekodera iz tačke a) realizovati dekodera 6/64

---

**Rešenje:**

a) Tabela 2.3.1 predstavlja funkcionalnu tabelu kojom se opisuje način rada dekodera 3/8

*Tabela 2.3.1 - Funkcionalna tabela dekodera 3/8 u opštem slučaju*

A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>	Y <sub>7</sub>	Y <sub>6</sub>	Y <sub>5</sub>	Y <sub>4</sub>	Y <sub>3</sub>	Y <sub>2</sub>	Y <sub>1</sub>	Y <sub>0</sub>
0	0	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	1	0	0	0	0	0	0	1	0	0
0	1	1	0	0	0	0	1	0	0	0
1	0	0	0	0	0	1	0	0	0	0
1	0	1	0	0	1	0	0	0	0	0
1	1	0	0	1	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0	0

Na osnovu funkcionalne tabele možemo izvesti logičku funkciju svakog od izlaza Y<sub>7</sub>.

o:

$$Y_0 = \bar{A}_2 \bar{A}_1 \bar{A}_0 \quad (2.3.2)$$

$$Y_1 = \bar{A}_2 \bar{A}_1 A_0 \quad (2.3.3)$$

$$Y_2 = \bar{A}_2 A_1 \bar{A}_0 \quad (2.3.4)$$

$$Y_3 = \bar{A}_2 A_1 A_0 \quad (2.3.5)$$

$$Y_4 = A_2 \bar{A}_1 \bar{A}_0 \quad (2.3.6)$$

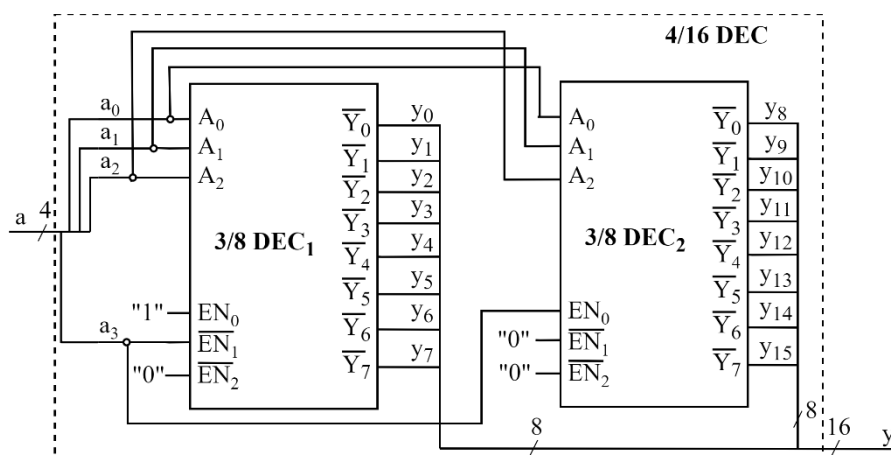
$$Y_5 = A_2 \bar{A}_1 A_0 \quad (2.3.7)$$

$$Y_6 = A_2 A_1 \bar{A}_0 \quad (2.3.8)$$

$$Y_7 = A_2 A_1 A_0 \quad (2.3.9)$$

b) U ovoj tački se zahteva realizacija dekodera 4/16 korišćenjem dekodera 3/8. Za razliku od dekodera 3/8 iz prethodne tačke, ovaj dekodera sadrži tri ulazna *enable* signala dok su mu izlazi aktivni u logičkoj nuli. Funkcije kojima se opisuju izlazni signali ovog dekodera predstavljaju komplemente funkcija (2.3.2) – (2.3.9).

Na slici 2.3.1 je prikazana realizacija dekodera 4/16 korišćenjem dva dekodera 3/8.



Slika 2.3.1 - Realizacija dekodera 4/16 korišćenjem dvadekodera 3/8

c) Pošto se u ovoj tački zahteva realizacija funkcije (2.3.1) korišćenjem dekodera, navedenu funkciju je neophodno izraziti u formi koja sadrži neke od komplementiranih proizvoda (2.3.1) – (2.3.9). Do ove forme je najlakše doći analizirajući Karnoovu kartu koja odgovara funkciji 2.3.1 i izdvajanjem izlaza aktivnih u logičkoj nuli koji se formiraju potpunim zbiovima. Sadržaj Karnoove karte, sa naznačenim oblastima koji se realizuju u formi potpunih zbirova je prikazan na slici 2.3.2

		A	
		0	1
CB	00	0	1
	01	0	1
	11	1	0
	10	1	1

Slika 2.3.2. – Sadržaj Karnoove karte za funkciju (2.3.1)

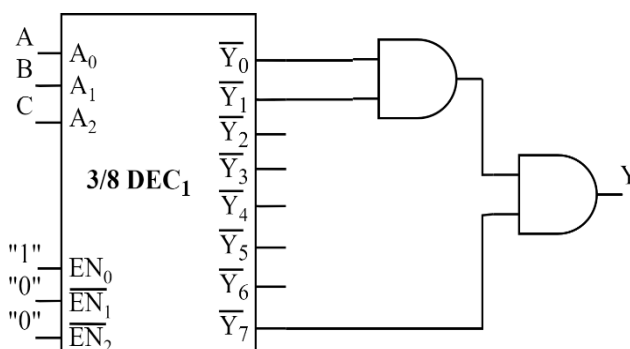
Na osnovu oblasti označenih na slici 2.3.2, izvodimo sledeću formu funkcije (2.3.1):

$$Y = (C + B + A)(C + \bar{B} + A)(\bar{C} + \bar{B} + \bar{A}) \quad (2.3.10)$$

Ukoliko funkciju (2.3.10) izrazimo koristeći funkcije (2.3.2.) – (2.3.9) dobijamo

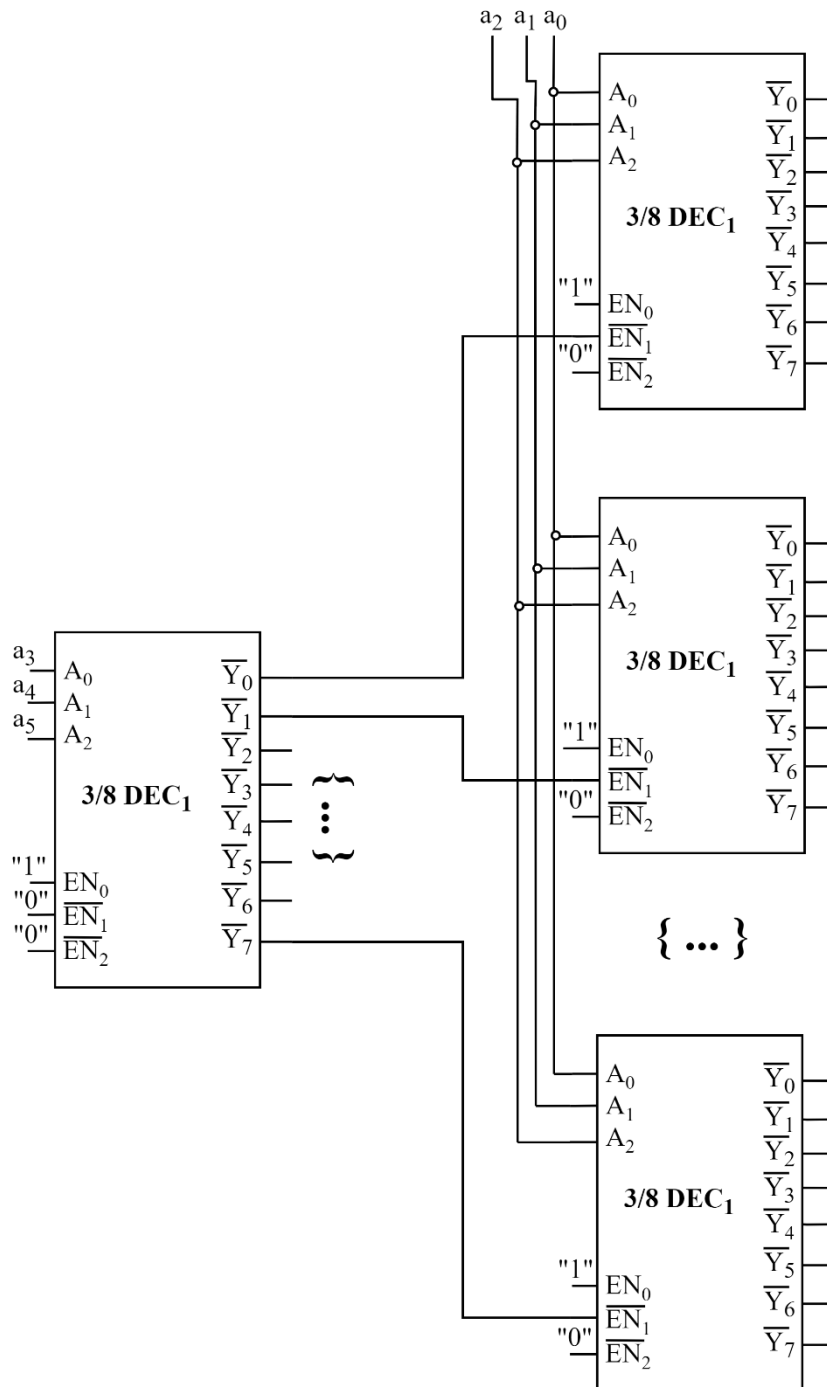
$$Y = \bar{Y}_0 \bar{Y}_1 \bar{Y}_7 \quad (2.3.11)$$

Na osnovu (2.3.11) realizujemo kombinacionu mrežu prikazanu na slici 2.3.3



Slika 2.3.3 - Kombinaciona mreža funkcije (2.3.1) realizovana korišćenjem dekodera 3/8

d) Na slici 2.3.4 je prikazana realizacija dekodera 6/64 korišćenjem dekodera 3/8.



Slika 2.3.4. - Realizacija dekodera 6/64 korišćenjem dekodera 3/8

### Zadatak 2.4.

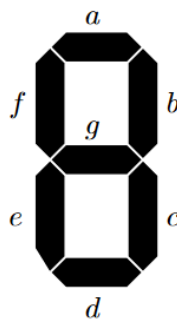
a) Projektovati konvertor BCD koda u sedmobitni kod za pobudu svetlosnog sedmosegmentnog LED indikatora sa zajedničkom katodom. Ako se na ulazu pojave kombinacije od 0000 do 1001 na indikatoru se prikazuju cifre 0-9. Ako se na ulazu pojave nedozvoljene kombinacije 1010-1111, na idikatoru se prikazuje slovo E (*Error*). Na

raspolaganju su samo NI i NILI logička kola. Težiti da broj upotrebljenih logičkih kola bude minimalan.

b) Projektovati kombinacionu mrežu, ako je poznato da se na ulazu konvertora koda ne mogu pojaviti nedozvoljene kombinacije ulaza, tj. ulazni podatak je u sigurno u opsegu 0000-1001, Na raspolaganju su logička kola proizvoljnog tipa.

c) Kako treba modifikovati dobijenu šemu pod tačkom a) da bi se, u slučaju kada se konvertora koristi u više cifarskom indikatoru, obezbedilo gašenje vodećih nula, i slova E, u slučaju da je na ulazu bilo kog indikatora, nalazi neka od nedozvoljenih kombinacija. Na raspolaganju su logička kola proizvoljnog tipa.

Ulaz u višecifreni BCD konvertor	Sadržaj na višecifrenom sedmosegmentnom LE displeju
0231	231
00A5	E
0504	504



Slika 2.4.1 - Sedmosegmentni displej sa označenim segmentima

### Rešenje:

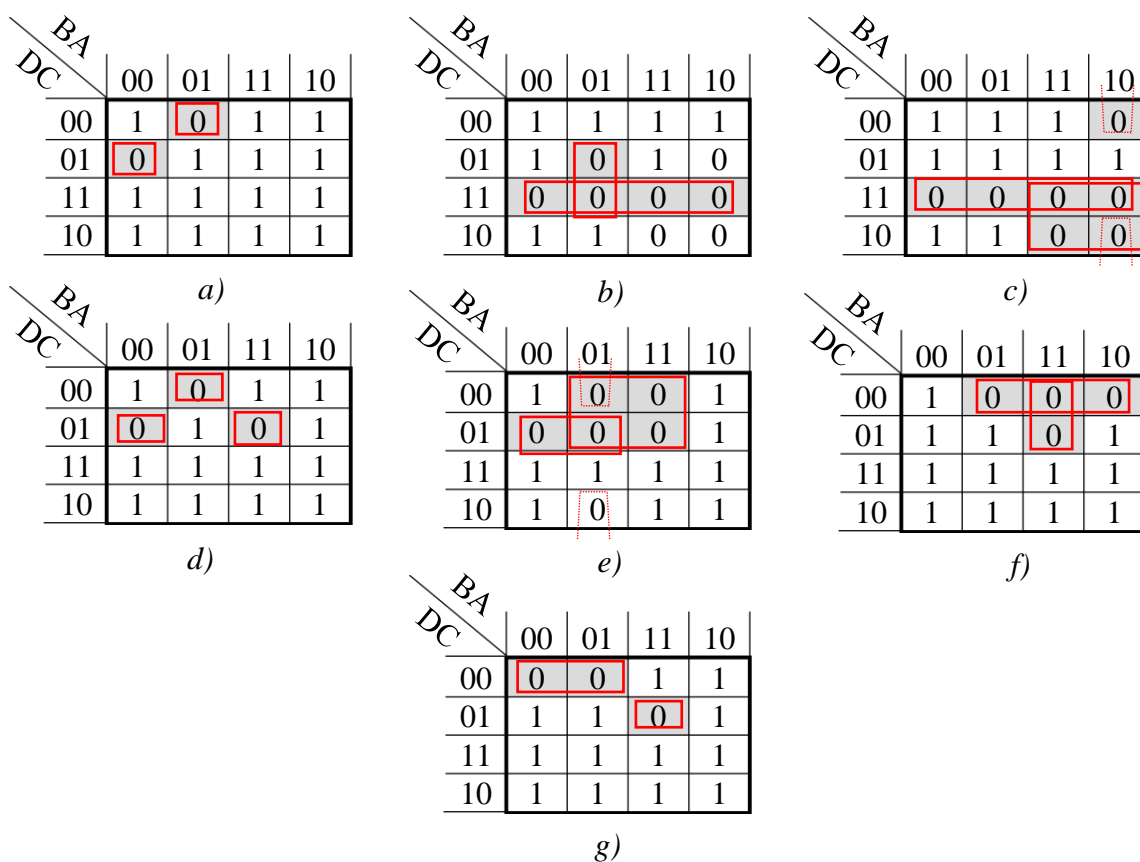
a) Najpre određujemo funkcionalnu tabelu koja definiše vrednosti izlaznih signala za sve kombinacije ulaznih logičkih nivoa. Na izlazu kombinacione mreže imamo sedmobitni kod (*abcdefg*) koji se koristi za pobudu led segmenata. Obzirom da je, prema tekstu zadatka, indikator sa zajedničkom katodom, aktivan logički nivo za pobudu nekog segmenta indikatora je nivo logičke jedinice.

Tabela 2.4.1 - Funkcionalna tabela uz zadatak 2.4.-a

D	C	B	A	a	b	c	d	e	f	g
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	0	1	1	0	0	0	0
0	0	1	0	1	1	0	1	1	0	1
0	0	1	1	1	1	1	1	0	0	1
0	1	0	0	0	1	1	0	0	1	1
0	1	0	1	1	0	1	1	0	1	1
0	1	1	0	1	0	1	1	1	1	1
0	1	1	1	1	1	1	0	0	0	0
1	0	0	0	1	1	1	1	1	1	1

1	0	0	1	1	1	1	1	0	1	1
1	0	1	0	1	0	0	1	1	1	1
1	0	1	1	1	0	0	1	1	1	1
1	1	0	0	1	0	0	1	1	1	1
1	1	0	1	1	0	0	1	1	1	1
1	1	1	0	1	0	0	1	1	1	1
1	1	1	1	1	0	0	1	1	1	1

Na osnovu kombinacione tabele 2.4.1, popunjavamo odgovarajuće Karnoove karte a zatim određujemo funkciju svakog od izlaza. U nastavku su date Karnoove karte za svaki od izlaza BCD konvertora.



Slika 2.4.2 - Sadržaj Karnoovih karti za svaki od izlaza BCD konvertora

Na osnovu sadržaja Karnoovih karti moguće je odrediti funkcije koje podrazumevaju minimalan broj logičkih kola sa minimalnim brojem ulaza. Svaki od Izlaza logičkog kola je pogodan za implementaciju korišćenjem NILI logičkih kola i zbog toga se u Karnoovim kartama posmatraju logičke nule. U tabeli 2.4.2 je dat pregled funkcija izvedenih iz Karnoovih karti za svaki od izlaza

Tabela 2.4.2 - Pregled funkcija za svaki od izlaza BCD konvertora

Izlazni signal	Funkcija
<i>a</i>	$(D + \bar{C} + B + A)(D + C + B + \bar{A})$
<i>b</i>	$(\bar{D} + \bar{C})(\bar{B} + \bar{D})(\bar{B} + A + \bar{C})(B + \bar{A} + \bar{C})$
<i>c</i>	$(\bar{D} + \bar{C})(\bar{B} + \bar{D})(\bar{B} + A + C)$
<i>d</i>	$(D + \bar{C} + B + A)(D + C + B + \bar{A})(D + \bar{C} + \bar{B} + \bar{A})$

$e$	$(\bar{A} + D)(D + \bar{C} + B)(C + B + \bar{A})$
$f$	$(D + \bar{B} + \bar{A})(D + C + \bar{A})(D + C + \bar{B})$
$g$	$(D + C + B)(D + \bar{C} + \bar{B} + \bar{A})$

**!Za samostalni rad:** Izvesti funkciju za svaki od izlaza ako se umesto NILI logičkih kola koriste NI logička kola. Uporediti dobijene funkcije sa funkcijama u tabeli 6 sa stanovišta broja logičkih kola.

Primenom Bulovih transformacija nad funkcijom svakog od signala potencijalno je moguće dodatno uprostiti izraze. U slučaju funkcije izlaznog signala  $a$  važe sledeće jednakosti:

$$a = (D + \bar{C} + B + A)(D + C + B + \bar{A}) \quad (2.4.1)$$

$$a = (D + B + \bar{C} + A)(D + B + C + \bar{A}) \quad (2.4.2)$$

primenom pravila Bulove algebre dobija se

$$a = D + B + (\bar{C} + A)(C + \bar{A}) \quad (2.4.3)$$

$$a = D + B + (\bar{C} \bar{A} + CA) \quad (2.4.4)$$

$$a = D + B + \overline{\overline{C} \oplus A} \quad (2.4.5)$$

Na osnovu dobijenog izraza (2.4.5) jasno je da se funkcija  $a$  može realizovati sa manjim brojem logičkih kola. Međutim, za takvu realizaciju potrebno je koristiti EXNILI logička kola koja nisu dozvoljena u ovoj tački zadatka.

**!Za samostalni rad:** Za ostale signale  $b-g$ , primenom Bulovih transformacija, naći formu koja sadrži manji broj logičkih kola

Pošto se u ovoj tački zadatka traži realizacija korišćenjem isključivo NI ili NILI logičkih kola, potrebno je funkcije izlaza iz Tabele napisati u odgovarajućoj formi. Pošto smo funkcije, u navedenoj tabeli, izrazili koristeći formu proizvoda zbirova, navedene funkcije je poželjno realizovati korišćenjem NILI logičkih kola. Zbog toga što broj ulaza nije ograničen tekstom zadatka, koristićemo NILI kola sa proizvoljnim brojem ulaza. Procedurom sličnom kao u prethodnim zadacima, dolazimo do forme logičke funkcije izlaznog signala  $a$  koja je pogodna za realizaciju NILI logičkim kolima

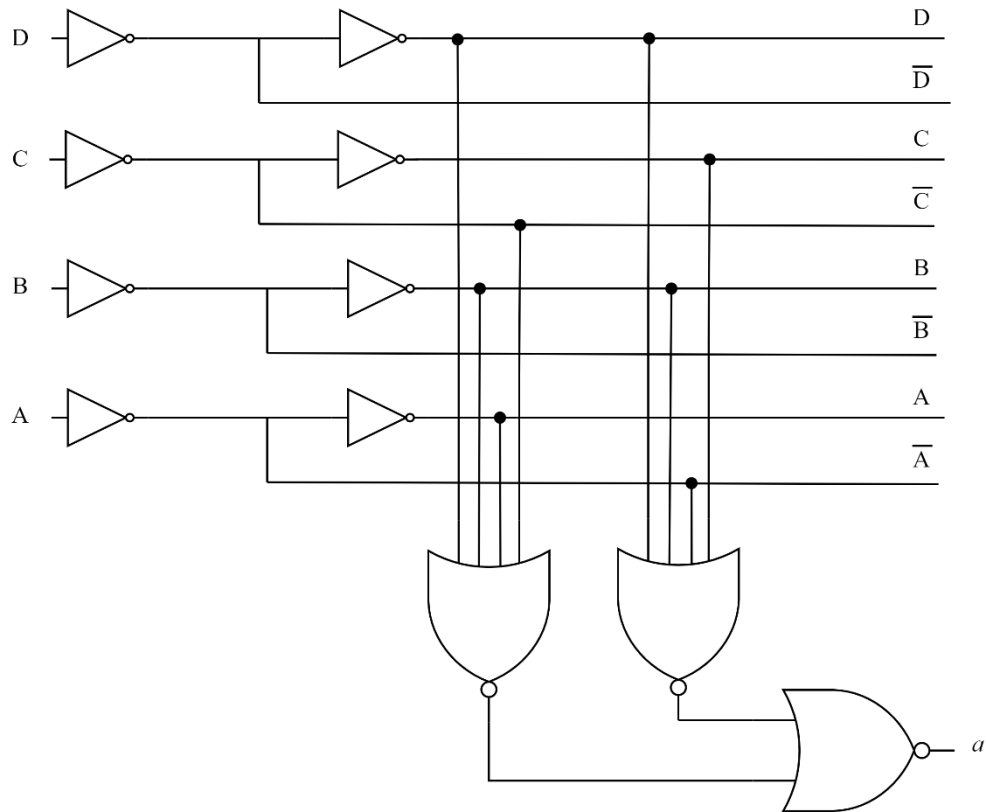
$$a = \bar{\bar{\bar{a}}} = \overline{\overline{(D + \bar{C} + B + A)(D + C + B + \bar{A})}} \quad (2.4.6)$$

$$a = \bar{\bar{\bar{a}}} = \overline{\overline{D + \bar{C} + B + A + D + C + B + \bar{A}}}$$

Na slici 2.4.3 je ilustrovana implementacija izlaznog signala  $a$  korišćenjem isključivo NILI logičkih kola.

**!Za samostalni rad:**

- 1) Za ostale signale  $b-g$  izvesti algebarski zapis pogodan za implementaciju korišćenjem isključivo NILI logičkih kola.
- 2) Za ostale signale  $b-g$  izvesti algebarski zapis pogodan za implementaciju korišćenjem isključivo NI logičkih kola.
- 3) Uporediti realizacije sa stanovišta broja logičkih kola



Slika 2.4.3 - Implementacija izlaza  $a$  korišćenjem NILI logičkih kola

b) Najpre određujemo kombinacionu tabelu koja definiše vrednosti izlaznih signala za sve kombinacije ulaznih logičkih nivoa. Na izlazu imamo sedmobitni binarni broj ( $abcdefg$ ). Kako je poznato da na ulazu konvertora može biti samo neka od kombinacija 0000-1001, vrednosti izlaza, u slučaju nedozvoljene kombinacije signala na ulazu kola, mogu imati proizvoljnu vrednost (označenu sa  $bbbbbb$  u funkcionalnoj tabeli) te ih je preporučljivo koristiti prilikom određivanja logičke funkcije minimalne kompleksnosti

Tabela 2.4.3 - Funkcionalna tabela uz zadatak 2.4-b

D	C	B	A	a	b	c	d	e	f	g
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	0	1	1	0	0	0	0
0	0	1	0	1	1	0	1	1	0	1
0	0	1	1	1	1	1	1	0	0	1
0	1	0	0	0	1	1	0	0	1	1
0	1	0	1	1	0	1	1	0	1	1

0	1	1	0	1	0	1	1	1	1	1
0	1	1	1	1	1	1	0	0	0	0
1	0	0	0	1	1	1	1	1	1	1
1	0	0	1	1	1	1	1	0	1	1
1	0	1	0	b	b	b	b	b	b	b
1	0	1	1	b	b	b	b	b	b	b
1	1	0	0	b	b	b	b	b	b	b
1	1	0	1	b	b	b	b	b	b	b
1	1	1	0	b	b	b	b	b	b	b
1	1	1	1	b	b	b	b	b	b	b

Na slici 2.4.4 je predstavljen sadržaj Karnoove karte za izlaz  $a$  na osnovu koga zaključujemo da je u ovom slučaju jednostavnija implementacija korišćenjem NI logičkih kola.

		BA			
		00	01	11	10
DC	00	1	0	1	1
	01	0	1	1	1
	11	b	b	b	b
	10	1	1	b	b

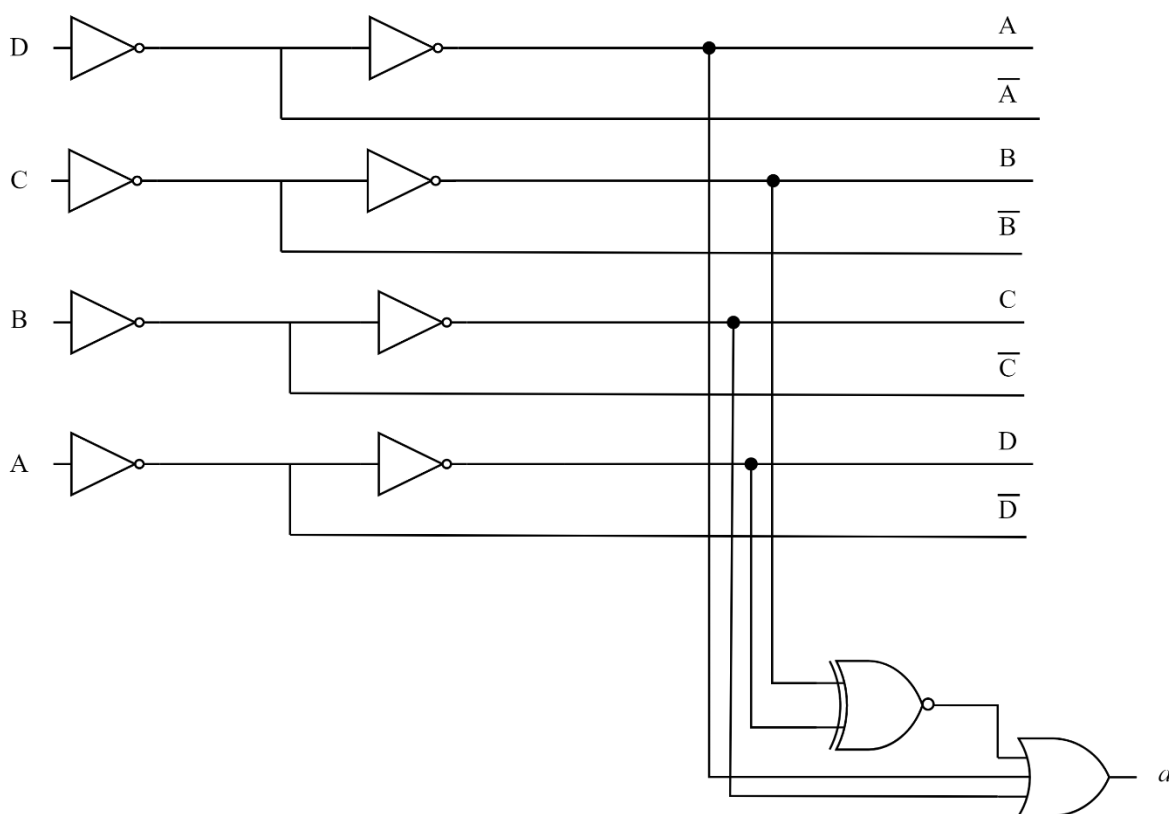
Slika 2.4.4 - Sadržaj Karnoove karte za izlaz  $a$

$$a = D + B + CA + \overline{C} \overline{A} \quad (2.4.7)$$

$$a = D + B + \overline{C \oplus A} \quad (2.4.8)$$



Pošto u ovoj tački zadatka nismo ograničeni na NI i NILI logička kola, moguće je implementirati funkciju izlaza. Na slici 2.4.5 je predstavljena ova implementacija



Slika 2.4.5 - Implementacija funkcije izlaza *a*

**!Za samostalni rad:** Za ostale izlazne signale *b-g*, popuniti odgovarajuće Karnoove karte i implementirati minimalne funkcije. Uporediti dobijene implementacije sa implementacijama u tački *a*).

c) Celokupnu funkcionalnost definisanu u okviru ove tačke zadatka moguće je dekomponovati na sledeće manje zahteve:

- (1) Kontrola uključenosti segmenata
- (2) Detekcija nedozvoljenog ulaza
- (3) Detekcija vodećih nula

Zahtev (1) podrazumeva uvođenje dodatnog signala u okviru kombinacione mreže BCD konvertora koji bi omogućio isključivanje svih segmenata LE displeja. Ovaj signal bi predstavljao ulazni signal u BCD konvertor dok bi logika za kontrolu ovog signala bila izmeštena izvan samog BCD konvertora. Funkcionalni zahtev (2) podrazumeva uvođenje još jednog dodatnog izlaznog signala koji ima za cilj prosleđivanje informacije o nevalidnosti ulaza ostalim BCD konvertorima koji čine višecifreni BCD konvertor. Jedan od najkopleksnijih funkcionalnih zahteva podrazumeva implementaciju logike za detekciju vodećih nula koja ima za cilj isključivanje segmenata onih LE displeja koji su povezani na vodeće nule. Za potpunu implementaciju ove logike neophodno je da do svakog BCD konvertora, u nizu konvertora koji čine višecifreni BCD konvertor, propagira informacija o tome da li su prethodni BCD konvertori viših cifara imali niz vodećih nula. Na taj način bi se omogućilo isključivanje segmenata BCD konvertora koji je poslednji u nizu ukoliko bi se i na njegovom ulazu pojavio binarni zapis cifre 0.

Na osnovu analize tri glavna funkcionalna zahteva, moguće je identifikovati sledeće signale u okviru svakog od BCD konvertora:

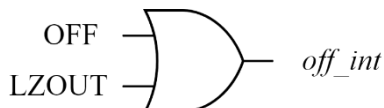
Tabela 2.4.4 - Tabela opisa signala BCD konvertora

Naziv signala	Ulazni/Izlazni	Aktivni logički nivo	Opis
<i>Error</i>	Izlazni	1	Označava da se na ulazu BCD konvertora pojavila nedozvoljena kombinacija bita
<i>OFF</i>	Ulazni	1	Kontroliše uključenost segmenata. Kada je postavljen na „0“ svi segmenti su isključeni dok je u suprotnom stanje segmenata određeno drugim signalima
<i>LZIN</i>	Ulazni	1	Leading Zeros ulazni signal BCD konvertora niže cifre prihvata Leading Zeros sa BCD konvertora više cifre. Ukoliko su vodeće nule isključene signal je postavljen na jedinicu dok je u suprotnom postavljen na nulu
<i>LZOUT</i>	Izlazni	1	Leading Zeros izlazni signal BCD konvertora više cifre se prosleđuje na BCD konvertor niže cifre. Ukoliko su vodeće nule isključene signal je postavljen na jedinicu od strane BCD konvertora više cifre dok je u suprotnom postavljen na nulu

*Error* signal predstavlja izlazni signal iz jednog BCD konvertora koji se generiše kada se na ulazu pojavi binarni sadržaj u opsegu od 1010 -1111. Na osnovu analize sprovedene u tačkama a) i b) i uvidom u Karnoove karte svakog od izlaznih signala možemo izvesti minimalnu logičku funkciju *Error* signala:

$$Error = D \cdot (C + B) \quad (2.4.9)$$

Na putu do realizacije kompletne kontrole uključenosti svih segmenata LE displeja, uvešćemo jedan pomoćni signal *off\_int* koji je aktivan u logičkoj jedinici. Logika za generisanje ovog internog signala BCD konvertoraje prikazana na slici 2.4.6



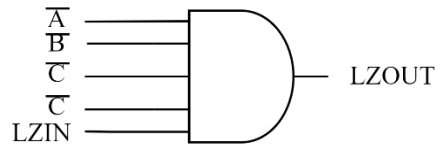
Slika 2.4.6 - Logika za kontrolu *off\_int* signala

Do postavljanja ovog signala na nivo logičke jedinice dolazi ukoliko je eksterni signal *OFF* postavljen na nivo logičke jedinice ili ukoliko je *LZOUT* signal postavljen na nivo logičke jedinice. Drugim rečima, segment koji je kontrolisan *off\_int* signalom će biti isključen ako se izvan BCD konvertora aktivira *OFF* signal ili ukoliko je BCD konvertor u nizu vodeći nula.

Logika za generisanje *LZOUT* signala se nalazi unutar BCD konvertora i definisana je izrazom (2.4.10) do je implementacija predstavljena na slici 2.4.7.

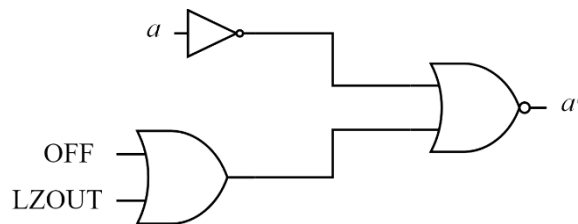
$$LZOUT = LZIN \cdot \bar{A} \bar{C} \bar{B} \bar{A} \quad (2.4.10)$$

Dakle, ukoliko su u okviru BCD konvertora viših cifara detektovane vodeće nule ( LZIN = 1) i ako se na ulazu trenutnog BCD konvertora nalazi binarna kombinacija koja odgovara cifri nula, LZOUT se postavlja na nivo logičke jedinice.



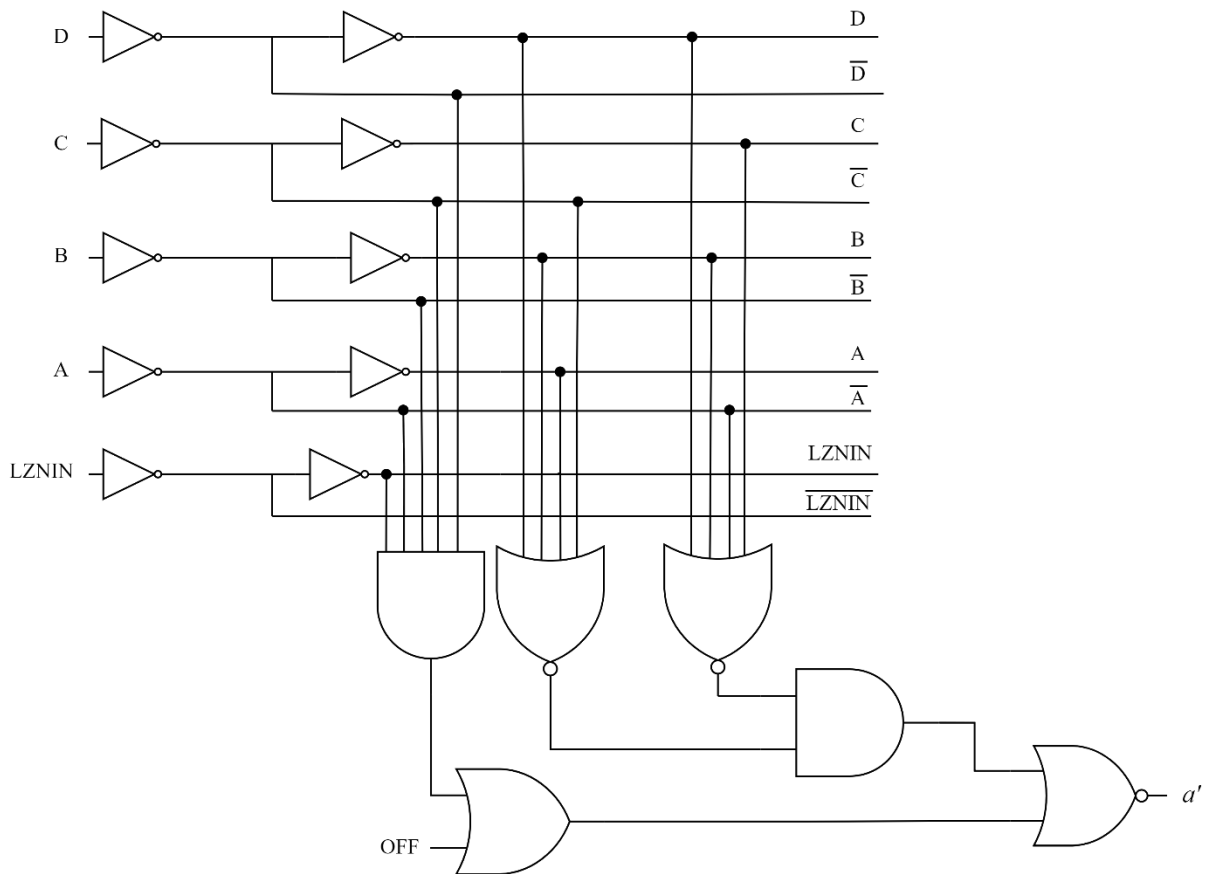
Slika 2.4.7 - Logika za kontrolu LZOUT signala

Logika ilustrovana na slikama 2.4.6 i 2.4.7 predstavlja glavne komponente kontrole uključenosti svakog od segmenata. Na slici 2.4.8 je prikazana logika kojom se kontroliše uključenost segmenta *a*. Kao što se sa slike 2.4.8 može i uočiti, signal *a*' predstavlja dodatno kontrolisan signal *a* čija je funkcija već predstavljena u prethodnim tačkama.



Slika 2.4.8 - Logika za kontrolu uključenosti segmenta *a*

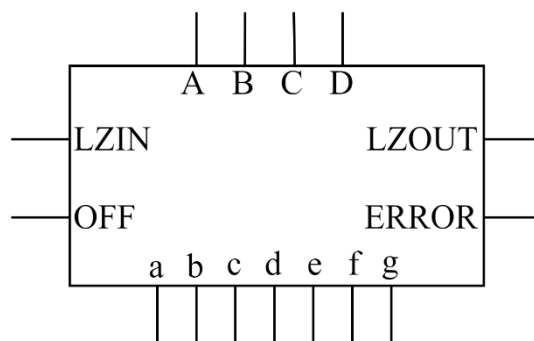
Na slici 2.4.9 je predstavljena logička šema BCD konvertora koja implementira set funkcionalnosti potreban za realizaciju višecifrenog BCD konvertora.



Slika 2.4.9 - Logika jednog BCD konvertora sa dodatnim kontrolnim signalima iz tabele

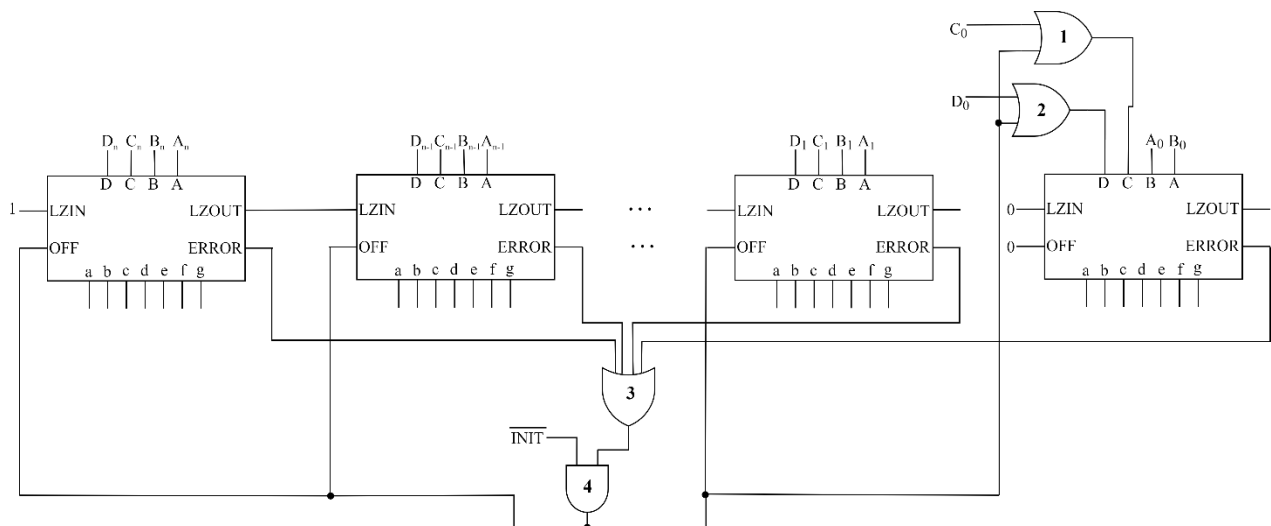
**Za samostalni rad:** Za ostale izlazne signale  $b'$ - $g'$  nacrtati odgovarajuće logičke šeme.

Logika predstavljena na slici 2.4.9 je enkapsulirana u funkcionalni blok BCD konvertora koji će biti iskorišćen za sintezu višecifrenog BCD konvertora. Ovaj funkcionalni blok je predstavljen na slici 2.4.10.



Slika 2.4.10 - Funkcionalni blok BCD konvertora

Funkcionalni blok predstavljen na slici 2.4.10 je iskorišćen za realizaciju višecifrenog BCD konvertora kao što je prikazano na slici 2.4.11.



Slika 2.4.11 - Višebitni BCD konvertor

Kao što se sa slike 2.4.11 može videti, ERROR signali svih blokova su dovedeni na jedno OR kolo (3) što funkcionalno označava da, ukoliko se pojavi nedozvoljena kombinacija na bilo kom ulazu, generisaće se globalni OFF signal. Ovaj signal kontroliše isključenje svih segmenata u okviru svakog od LE displeja osim displeja najniže cifre višecifrenog BCD konvertora. Najniža cifra je uvek uključena bez obzira šta je na ulazu BCD konvertora viših cifara. Međutim, u slučaju da je OFF signal postavlje na nivo logičke jedinice (što označava da je bilo nedozvoljene kombinacije na nekom od ulaza BCD konvertora koji odgovaraju višim ciframa), koristeći OR logička kola (1 i 2) doći će do propagiranja te informacije i postavljanja nedozvoljene kombinacije na ulaz BCD konvertora najniže cifre što za posledicu ima ispis slova E na LE displej najniže cifre.

**Napomena:** Zbog pojave povratne sprege (na primer signal ERROR je, sa izlaza BCD konvertora najniže cifre, posredstvom logičkih kola 3, 4, 2 ili 3,4,1 vraćen na ulaz istog BCD konvertora) može doći do pojave „pamćenja“ vrednosti u povratnoj sprezi koja za posledicu ima nemogućnost promene ispisa na LE displeju najniže cifre. O ovoj pojavi će biti više reči na kursu Digitalna Elektronika 2. Da bi se kolo izvelo iz navedenog stanja, uvodimo kontrolni signal E koji ima za cilj da resetuje vrednost zapamćenu u okviru povratne sprege kombinacione mreže.

### 3. Zadaci za samostalni rad

#### Zadatak 3.1.

- a) Pomoću dekodera  $n/2^n$  i  $m/2^m$  projektovati dekodera  $(n+m)/2^{(n+m)}$ . Na raspolaganju su dvoulazna logička I kola. Dekodere  $n/2^n$  i  $m/2^m$  crtati kao blokove.
- b) Pomoću dvoulaznih I kola i invertora realizovati dekodera 4/16 kod koga su i ulazni i izlazni signali aktivni na logičkoj jedinici. Težiti da broj upotrebljenih logičkih kola bude minimalan.
- c) Projektovati kolo dekodera koje ima 13 ulaza tako da realizacija bude izvedena u maksimalno 4 nivoa. Na raspolaganju su dvoulazna I kola i invertori. Težiti da broj upotrebljenih logičkih kola bude minimalan. Vreme propagiranja kroz logičko kolo iznosi  $t_{dlk}$  dok se kašnjenje signala invertora može zanemariti.

#### Zadatak 3.2.

F-ju:

$$Y = \bar{D}C\bar{A} + \bar{B}A + \bar{C}A \quad (2.3.1)$$

realizovati isključivo korišćenjem osnovnih logičkih kola sa proizvoljnim brojem ulaza i multipleksera 4/1 za čije selekzione signale  $S_1S_0$  važi:

- a)  $S_1S_0 = DC$   
b)  $S_1S_0 = BA$   
c)  $S_1S_0 = DB$   
d)  $S_1S_0 = CA$

Koja od realizacija a), b), c) i d) zahteva minimalan broj logičkih kola?