CS120 - ORGANIZACIJA RAČUNARA

Sloj digitalne logike – kombinatorna kola

Lekcija 03

PRIRUČNIK ZA STUDENTE

CS120 - ORGANIZACIJA RAČUNARA

Lekcija 03

SLOJ DIGITALNE LOGIKE – KOMBINATORNA KOLA

- Sloj digitalne logike – kombinatorna kola
- Poglavlje 1: Logička kola i Bulova algebra
- Poglavlje 2: Minimizacija i realizacija logičkih funkcija
- Poglavlje 3: Logičke komponente
- Poglavlje 4: Aritmetičke komponente
- Poglavlje 5: Aritmetičko-Logička jedinica (ALU)
- Poglavlje 6: Pokazne vežbe
- Poglavlje 7: Primeri za samostalni rad
- Poglavlje 8: Domaći zadatak
- Zaključak

UVOD

Uvod u lekciju #3

U ovoj lekciji biće opisani uvodni pojmovi vezani za digitalnu elektroniku. Prvo se uvodi pojam binarnih digitalnih signala.

Zatim je opisana funkcionalnost logičkih kola koja obavljaju osnovne logičke operacije, manipulišući vrednostima takvih signala.

Nakon pregleda osnovnih zakona Bulove algebre, kao fundamentalne oblasti matematike na kojoj se zasniva rad digitalnih sistema, dato je nekoliko primera realizacije logičkih funkcija.

Poglavlje 1

Logička kola i Bulova algebra

OBRADA I TRANSFORMACIJA PODATAKA

Obrada i transformacija podataka bavljaju se pomoću logičkih kola, Bulove algebre, logičkih funkcija, i kombinacionih komponenti.

Složeni digitalni sistemi se projektuju tako što se najpre ukupna funkcija sistema razloži na višepod-funkcija ili pod-rutina (en. sub-function, subroutine).
Ove pod-funkcije realizuju se u vidu jednostavnijih digitalnih modula,
čijim se povezivanjem konstruiše željeni sistem. Iako je u nekim slučajevima neophodno projektovati module za neke specifične funkcije.

U praksi, složeniji digitalni sistemi se mogu realizovati korišćenjem *standardnih modula* ili *komponenti* (en. component).

Standardne komponente obavljaju funkcije za koje je uočeno da su korisne za veliki broj različitih primena, a dostupne su u vidu integrisanih kola ili bibliotečkih komponenti i kao takve spremne za direktnu ugradnju u sistem koji se projektuje.

Većina digitalnih sistema, uključujući i računare, projektovana je da obrađuje ili transformiše podatke.

Ove obrade i transformacije mogu biti različitih tipova, uključujući aritmetičke i logičke operacije, kodiranje i dekodiranje podataka i reorganizaciju podataka.

U opštem slučaju, navedene transformacije podataka se obavljaju pomoću:

- *logičkih kola* (en. logic gates);
- *Bulove algebre* (en Boolean algebra);
- *logičkih funkcija* (en. logic functions);
- *kombinacionih komponenti* (en. combinatorial components).

U elektronici, pod pojmom signala podrazumeva se električna veličina koja može da menja vrednost tokom vremena. Ta veličina je najčešće napon, a ređe se koriste i strujni signali.

DIGITALNI SIGNALI NASPRAM ANALOGNIH

Binarni digitalni signal tokom vremena može imati ssamo dve vrednosti: visoku vrednost - logičkom jedinicom i nisku vrednost - logičkom nulom.

Kada su u pitanju elektronska kola koja se napajaju iz jednosmernog izvora napona, uobičajeno je da je opseg vrednosti naponskih signala ograničen naponom napajanja.

Analogni signal je signal koji u proizvoljnom trenutku može imati bilo koju vrednost u okviru datog opsega.

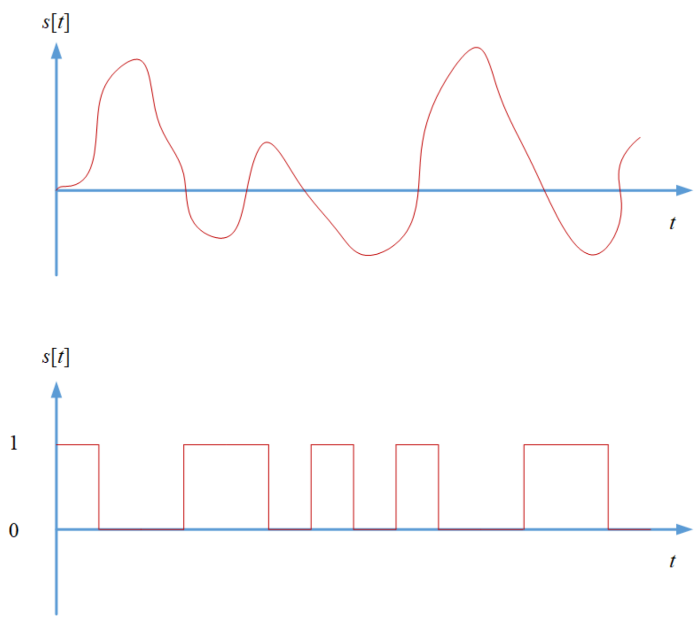
Sa druge strane, digitalni signal u svakom vremenskom trenutku može imati jednu od ***nekoliko unapred određenih diskretnih vrednosti***.

Binarni digitalni signal (en. binary digital signal) tokom vremena može imati ssamo dve vrednosti:*visoku vrednost*, koja se naziva *logičkom jedinicom* i *nisku vrednost*, koja se naziva *logičkom nulom*.

Tačne vrednosti napona koje odgovaraju ovim logičkim nivoima određene su prvenstveno tehnologijom izrade digitalnog kola.

Uobičajeno je da se logički nivoi interpretiraju kao tautološke vrednosti logičkih iskaza: nula (0) predstavlja netačno (en. False), dok jedinica (1) predstavlja tačno (en. True)

Na slici su prikazani analogni (gore) i digitalni (dole) signali u funkciji vremena.



Slika 1.1 Digitalni signal naspram analognog [Izvor: Autor].

LOGIČKA KOLA

Logičko kolo ili gejt je elektronsko kolo sa jednim ili više ulaza i jednim izlazom, koje obavlja određenu logičku operaciju.

Logičko kolo (en. logic circuit, logic gate) je elektronsko kolo sa jednim ili više ulaza i jednim izlazom, koje obavlja logičku operaciju tako što na osnovu stanja signala na ulazima dovodi signal na izlazu u odgovarajuće
logičko stanje.

Svako od osnovnih logičkih kola ima odgovarajući šematski simbol. Ponašanje kola se opisuje *kombinacionom tabelom* ili *tabelom istinitosti* u kojoj je za svaku moguću kombinaciju logičkih stanja na ulazima prikazano kakvo je stanje na izlazu.

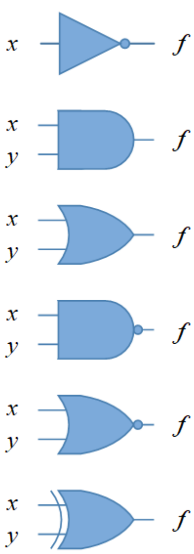
Pošto se binarni digitalni signali mogu naći u dva logička stanja (0 ili 1), kombinaciona tabela za kolo sa n ulaza sadrži ukupno 2^n kombinacija.

Logička kola mogu da se kombinuju u složenije strukture i da imaju i više ulaza na osnovu kojih se vrši transformacija tj. pravi izlazna vrednost po nekom zakonu.

Logička kola mogu se izraditi poluprovodničkim elementima kao što su diode ili tranzistori.

Na integrisanim kolima, logička kola izrađuju se MOFSET (Metal-Oxide-Semiconductor Field Effect Transistor) tranzistorima, pomoću kojih je moguće izgraditi sva bitna logička kola. O detaljima izrade logičkih kola pogledati u dodatnoj literaturi.

Na slici je prikaza tabela sa najčešće korišćenim logičkim kolima, njihovim simbolima, logičkim funkcijama koje izvršavaju, i broj tranzistora koji je potreban za realizaciju kola.



Simbol	Naziv (en.)	Naziv (sr.)	Funkcija	Broj tranzistora
	Invertor	Invertor	$f = \overline{x}$	2
	AND	I	$f = x \cdot y$	6
	OR	ILI	$f = x + y$	6
	NAND	NI	$f = \overline{x \cdot y}$	4
	NOR	NILI	$f = \overline{x + y}$	4
	XOR	Isključivo ILI	$f = x \oplus y$	14

Slika 1.2 Najšećće korišćena logička kola [Izvor: Autor].

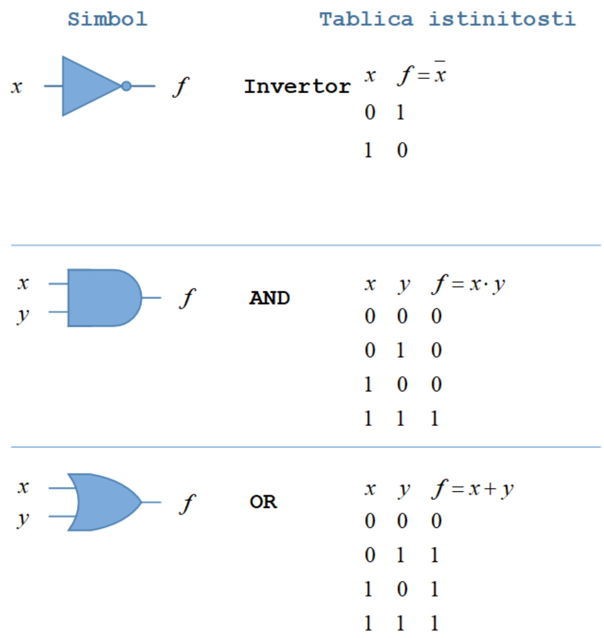
LOGIČKE FUNKCIJE

Logičke funkcije dobijaju se izvođenjem osnovnih logičkih operacija.

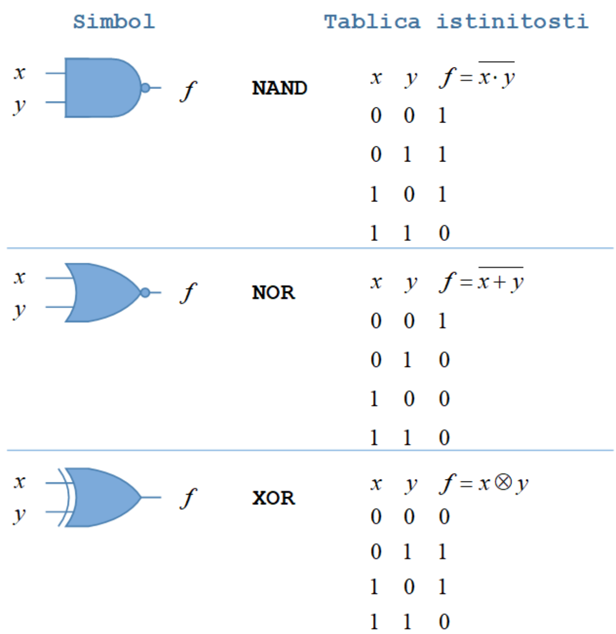
Logičke funkcije (en. logicgate functions) su funkcije koje se dobijaju izvođenjem i kombinovanjem osnovnih logičkih
operacija nad vrednostima ulaznih signala, koje u ovom kontekstu predstavljaju logičke promenljive.

U nastavku date su osnovne logičke funkcije sa tablicama istinitosti koje odgovaraju osnovnim logičkim kolima.

Sva kola osim kola invertora imaju dva (ili više ulaza).



Slika 1.3 Tablice istinitosti za Invertor, AND i OR kola [Izvor: Autor].



Slika 1.4 Tablice istinitosti za NAND, NOR i XOR kola [Izvor: Autor].

BULOVA ALGEBRA

Bulova algebra je matematička disciplina koja se bavi logičkim funkcijama i njihovim svojstvima.

Logičke funkcije se obično opisuju analitičkim izrazima u kojima se pojavljuju ulazne promenljive obeležene, konstante 0 i 1 i operacije između njih.

Bulova algebra je matematička disciplina koja se bavi logičkim funkcijama i njihovim svojstvima.

U nastavku su (bez dokazivanja) dati osnovni postulati Bulove algebre.

U daljem tekstu će se za negaciju koristiti *crtica iznad oznake* (en. hat) ili *apostrof* (') kao znak za komplement.

$$\overline{x} = x'$$

Osnovni postulati Bulove algebre:

Primer - operacije sa nulom:

```
0 * x = 0
0 + x = x
```

Primer - operacije sa jedinicom:

```
1 * x = x
1 + x = 1
```

Primer - dvostruka negacija:

```
x'' = (x')' = x
```

Primer - suprotne vrednosti:

```
x * x' = 0
x + x' = 1
```

Primer - iste vrednosti:

```
x * x = x
x + x = x
```

Svojstva Bulove algebre:

Komutativnost:

```
x * y = y * x
x + y = y + x
```

Asocijativnost:

```
x * (y * z) = (x * y) * z
x + (y + z) = (x + y) + z
```

Distributivnost:

```
x * (y + z) = (x * y) + (x * z)
x + (y * z) = (x + y) * (x + z)
```

Apsorpcija:

```
x * (x + y) = x
x + (x * y) = x
```

DE MORGANOVI ZAKONI

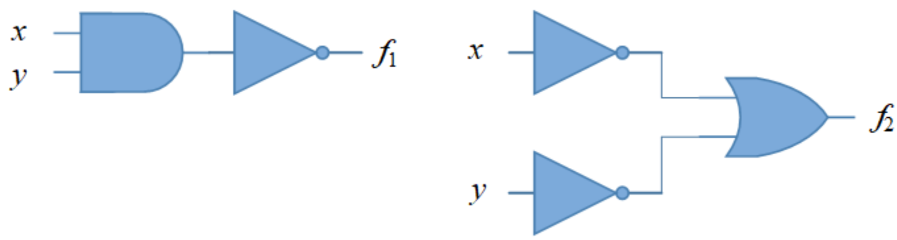
De Morganovi zakoni govore o invertovanjem izlaza

De Morganovi zakoni govore da invertovanjem izlaza bilo kojeg logičkog kola rezultovaće suprotnom tipu funkcije (AND u OR i obrnuto) sa pojedinačno invertovanim ulazima.

```
(x * y)' = x' + y'
(x + y)' = x' * y'
```

Primer:

Korišćenjem običnih logičkih funkcija ispitati tačnost De Morganovih zakona. U nastavku je provera De Morganovog zakona za inverziju AND kola.



x	y	*posle AND*	f_1
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

x	y	$\bar{x}$	$\bar{y}$	f_2
0	0	1	1	1
0	1	1	0	1
1	0	0	1	1
1	1	0	0	0

Slika 1.5 Tablice istinitosti za De Morganov zakon za inverziju AND kola [Izvor: Autor].

Zadatak za samostalni rad (10 minuta):

Samostalno proveriti De Morganov zakon za inverziju OR kola.

De Morganovi zakoni (video):

Ova lekcija sadrži video materijal. Ukoliko želite da pogledate ovaj video morate da otvorite LAMS lekciju.

STANDARDNE FORME LOGIČKIH FUNKCIJA

Primenom postulata i svojstava logičkih funkcija, kao i De Morganovih zakona, moguće je bilo koju funkciju svesti na neku od normalnih formi.

Logičke funkcije se u praksi obično realizuju svođenjem na jednu od dve standardne forme:

- zbir proizvoda;
- proizvod zbirova.

Pošto je pri realizaciji funkcije poželjno koristiti minimalan mogući broj logičkih kola, javlja se potreba za njihovim uprošćavanjem tj. minimizacijom funkcija koje opisuju logičko kolo.

Funkcija zadata u nekoj od standardnih formi je pogodna za grafički postupak minimizacije pomoću *Karnoovih mapa* (en. Karnaugh map).
Svaka logička funkcija može biti izražena na više
različitih, međusobno ekvivalentnih načina.

Proizvoljna logička funkcija zadata u algebarskom obliku, može se primenom zakona Bulove algebre svesti na jednu od dve standardne forme:

- Disjunktivna forma (en. disjunctive form)
- Konjunktivna forma (en. conjunctive form)

Disjunktivna standardna(ili normalna) forma (DNF) je zbir proizvoda u čijem formiranju učestvuju ulazne promenljive i njihove invertovane vrednosti.

Konjunktivna standardna (ili normalna) forma (KNF) je proizvod zbirova u čijem formiranju učestvuju ulazne promenljive i njihove invertovane vrednosti.

Primenom postulata i svojstava logičkih funkcija, kao i De Morganovih zakona, moguće je bilo koju funkciju svesti na neku od normalnih formi.

PRIMERI STANDARDNIH FORMI LOGIČKIH FUNKCIJA

Pojedine logičke funkcije lakše je obično svesti na jednu od normalnih formi.

Primer #1 (5) minuta)

Primenom postulata i svojstava logičkih funkcija, kao i De Morganovih zakona, sledeću logičku funkciju napisati u DNF i KNF obliku:

$$f = \overline{x \cdot \bar{y} + \bar{z} \cdot w}$$

Rešenje:

Disjunktivna normalna forma (DNF):

```
f = (x * y' + z' * w)'
  = (x * y')' * (z' * w)'
  = (x' + y'') * (z'' + w')
  = (x' + y) * (z + w')
  = x' * z + x' * w' + y * z + y * w'
```

Konjunktivna normalna forma (KNF):

```
f = (x * y' + z' * w)'
  = (x * y')' * (z' * w)'
  = (x' + y'') * (z'' + w')
  = (x' + y) * (z + w')
```

Napomena:

Pojedine logičke funkcije lakše je obično svesti na jednu od normalnih formi. Najčešće je potreban još jedan korak koji predstavlja jednu primenu svojstava logičkih funkcija.

Primer #2 (5 minuta)

Primenom postulata i svojstava logičkih funkcija, kao i De Morganovih zakona, sledeću logičku funkciju napisati u DNF i KNF obliku:

$$f = \overline{(\bar{x} + y)} \cdot z + \bar{w}$$

Rešenje:

Disjunktivna normalna forma (DNF):

```
f = (x' + y)' * z + w'
  = x'' * y' * z + w'
  = x * y' * z + w'
```

Konjuktivna normalna forma (KNF):

```
f = (x' + y)' * z + w'
  = x'' * y' * z + w'
  = x * y' * z + w'
  = (x + w') * (y' + w') * (z + w')
```

POTPUNE NORMALNE FORME

Ukoliko u svakom sabirku ili činiocu imamo sve ulazne promenljive, onda su u pitanju potpune normalne forme.

Za standardnu formu se kaže da je potpuna ukoliko u formiranju proizvoda, odnosno zbirova koji je sačinjavaju učestvuju sve ulazne promenljive.

Potpuna disjunktivna normalna forma (*PDNF*) predstavlja zbir proizvoda, a u svakom sabirku postoje sve ulazne promenljive.

Potpuna konjunktivna normalna forma (*PKNF*) predstavlja proizvod zbirova, a u svakom činiocu postoje sve ulazne promenljive.

Primer:

Datu funkciju proširiti u PDNF i PKNF:

```
f = x + y * z'
```

Rešenje:

PDNF:

```
f = x + y * z'
  = x * 1 * 1 + 1 * y * z'
  = x * (y + y') * (z + z') + (x + x') * y  z'
  = (x * y * z) + (x * y * z') + (x * y' * z) + (x * y' * z') + (x * y * z') + (x'
* y * z')
```

PKNF:

```
f = x + y * z'
  = (x + y) * (x + z')
  = (x + y + 0) * (x + 0 + z')
  = (x + y + z * z') * (x + y * y' + z')
  = (x + y + z) * (x + y + z') * (x + y + z') * (x + y' + z')
```

Poglavlje 2

Minimizacija i realizacija logičkih funkcija

PREDSTAVLJANJE FUNKCIJA KROZ TABLICU ISTINITOSTI

U tablicu istinitosti upisuju se vrednost funkcije za svaku moguću kombinaciju ulaznih promenljivih.

Jedan od alternativnih načina prevođenja funkcije iz jedne u drugu standardnu formu je pomoću kombinacione tabele tj. tabele istinitosti. U ovu tabelu se upisuje vrednost funkcije za svaku moguću kombinaciju ulaznih promenljivih.

Često se broj kombinacije ili indeks piše u koloni ispred kombinacije prve ulazne promenljive.

Primer:

Data je tablica istinitosti funkcije sa tru ulazne promenljive. Napisati funkciju u PNKF i PDNF.

indeks	x	y	z	f
0	0	0	0	0
1	0	0	1	1
2	0	1	0	0
3	0	1	1	0
4	1	0	0	1
5	1	0	1	1
6	1	1	0	1
7	1	1	1	1

Slika 2.1 Primer logičke funkcije kroz tablicu istinitosti [Izvor: Autor].

Ukoliko pišemo funkciju u obliku potpune disjunktivne forme, pratimo vrednost funkcije čiji su izlazi jedinice.

Za kombinaciju gde funkcija ima vrednost jedinice, ulazne promenljive pišemo u neinvertovanom obliku ukoliko su za tu kombinaciju jedinice, a u invertovanom obliku ako su nule.

$$f = \bar{x}\bar{y}z + x\bar{y}\bar{z} + x\bar{y}z + xy\bar{z} + xyz$$

Ukoliko funkciju pišemo u obliku potpune standardne konjunktivne forme, pratimo vrednost funkcije čiji su izlazi nule.

Za kombinaciju gde funkcija ima vrednost nule, ulazne promenljive pišemo u neinvertovanom obliku ukoliko su za tu kombinaciju nule, a u invertovanom obliku ako su jedinice.

$$f = (x + y + z) \cdot (x + \bar{y} + z) \cdot (x + \bar{y} + \bar{z})$$

KARNOOVE MAPE

Karnoova mapa je grafička metoda za minimizaciju logičkih funkcija.

Pošto je već pokazano da svaka logička funkcija može da se nađe u više međusobno ekvivalentnih formi, postavlja se pitanje kako doći do optimalne forme koja za realizaciju zahteva minimum resursa, odnosno minimalan broj potrebnih logičkih kola.

Problem svođenja logičke funkcije na optimalnu, tj. minimalnu formu naziva se minimizacija funkcije (en. function minimization).

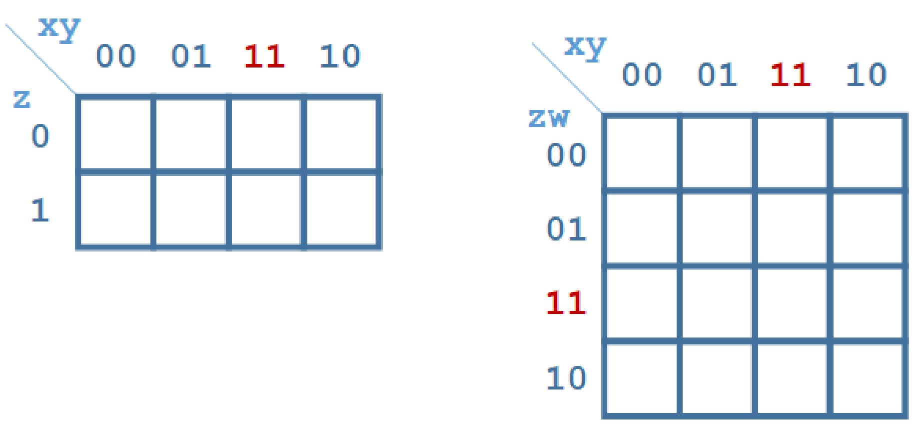
Postoji više metoda minimizacije, kojima je zajednička karakteristika da su zasnovani na zakonima Bulove algebre. Ovde će biti prikazana grafička metoda minimizacije funkcija *Karnoovih mapa* (en. Karnaugh maps)

https://en.wikipedia.org/wiki/Karnaugh_map

Karnoova mapa je zapravo pravougaona tabela koja sadrži 2^n polja. Svako od polja odgovara jednoj kombinaciji ulaznih promenljivih, i u otgovarajuće polje upisuje se vrednost funkcije za tu kombinaciju (0 ili 1).

Pri obeležavanju polja, treba voditi računa da svako susedno polje koje predstavlja dve ulazne kombinacije ima vrednosti koje se za jedan bit razlikuje od prethodne.

Na slici su prikazane prazne Karnoove mape za logičku funkciju sa tri i četiri ulaza.



Slika 2.2 Prazne Karnoove mape za logičke funkcije koje imaju tri (levo) i četiri (desno) ulaznih promenljivih [Izvor: Autor].

Vrednosti funkcije upisuju se u Karnoovu mapu, vodeći računa o redosledu izlaza.

Minimalnu logičku funkciju možemo dobiti u disjunktivnoj i u konjunktivnoj formi.

POPUNJAVANJE I ČITANJE KARNOOVIH MAPA - DISJKUNKTIVNA FORMA

Ukoliko hoćemo da dobijemo dijkunktivnu formu, onda treba da pokrijemo sve jedinice u mapi koristeći maske obliku četvorougaonih polja koja su stepeni broja 2.

Ukoliko hoćemo da dobijemo disjunktivnu formu, onda treba da pokrijemo sve jedinice u mapi koristeći maske u obliku četvorougaonih polja koja su stepeni broja 2 (1, 2, 4, 8, ...).

Maske treba da pokriju sve jedinice, i ne smeju prelaziti preko nula. Maske mogu da se delimično preklapaju, i mogu se *wrap*-ovati.

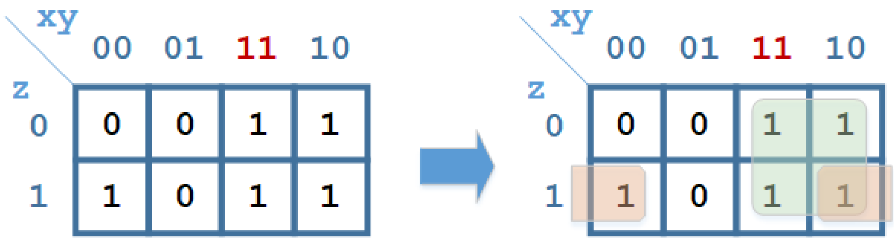
Sada gledamo samo maske. Svakoj od maski pridružuje se jedan proizvod ulaznih promenljivih, a činioci se formiraju na sledeći način:

- promenljive koje u svim poljima maske imaju vrednost 1 biće neinvertovane;
- promenljive koje imaju vrednost 0 se invertuju;
- promenljive koje menjaju vrednost u okviru maske ne učestvuju u formiranju proizvoda.

Primer:

Za funkciju datu na početku, iz tablice istinitosti popuniti Karnoovu mapu i uraditi minimizaciju funkcije u disjunktivnoj formi.

Rešenje:



Slika 2.3 Karnoova mapa i minimizacija funkcije u disjunktivnoj formi [Izvor: Autor].

Minimizovana disjunktivna forma logičke funkcije je sledeća:

$$f = x + \bar{y}z$$

POPUNJAVANJE I ČITANJE KARNOOVIH MAPA - KONJUNKTIVNA FORMA

Ukoliko hoćemo da dobijemo konjunktivnu formu, onda treba da pokrijemo sve nule u mapi koristeći maske obliku četvorougaonih polja koja su stepeni broja 2.

Ukoliko hoćemo da dobijemo konjunktivnu formu, onda treba da pokrijemo sve nule u mapi koristeći maske u obliku četvorougaonih polja koja su stepeni broja 2 (1, 2, 4, 8, ...).

Maske treba da pokriju sve nule, i ne smeju prelaziti preko jedinica. Maske mogu da se delimično preklapaju, i mogu se *wrap*-ovati.

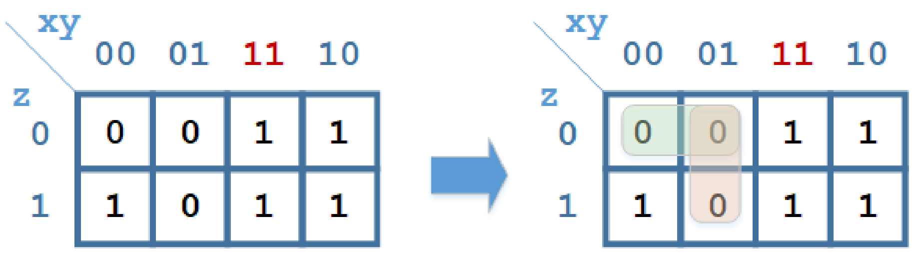
Sada gledamo samo maske. Svakoj od maski pridružuje se jedan zbir ulaznih promenljivih, a sabirci se formiraju na sledeći način:

- promenljive koje u svim poljima maske imaju vrednost 0 biće neinvertovane;
- promenljive koje imaju vrednost 1 se invertuju;
- promenljive koje menjaju vrednost u okviru maske ne učestvuju u formiranju proizvoda.

Primer:

Za funkciju datu na početku, iz tablice istinitosti popuniti Karnoovu mapu i uraditi minimizaciju funkcije u konjunktivnoj formi.

Rešenje:



Slika 2.4 Karnoova mapa i minimizacija funkcije ukonjunktivnoj formi [Izvor: Autor].

Minimizovana konjunktivna forma logičke funkcije je sledeća:

$$f = (x + z) \cdot (x + \bar{y})$$

REALIZACIJA LOGIČKIH FUNKCIJA KROZ LOGIČKA KOLA

Svaka logička funkcija se može predstaviti tj. realizovati logičkim kolima.

Svaka logička funkcija se može predstaviti tj. realizovati logičkim kolima.

Proizvode formiramo pomoću AND kola, zbirove pomoću OR kola, a invertovane vrednosti pomoću invertora tj. NOT kola.

Primer:

Predstaviti logičku funkciju predstavljenom skupom indeksa P tablicom istinitosti, zatim minimizovati funkciju korišćenjem Karnoovih mapa, i realizovati funkciju koristeći logička kola.

$$f\left(x,y,z\right)=^{P}\left(1,4,5,6,7\right)$$

Napomena:

Kada je funkcija predstavljena kroz skup indeksa, na datim indeksima se nalazi određena izlazna vrednosti.

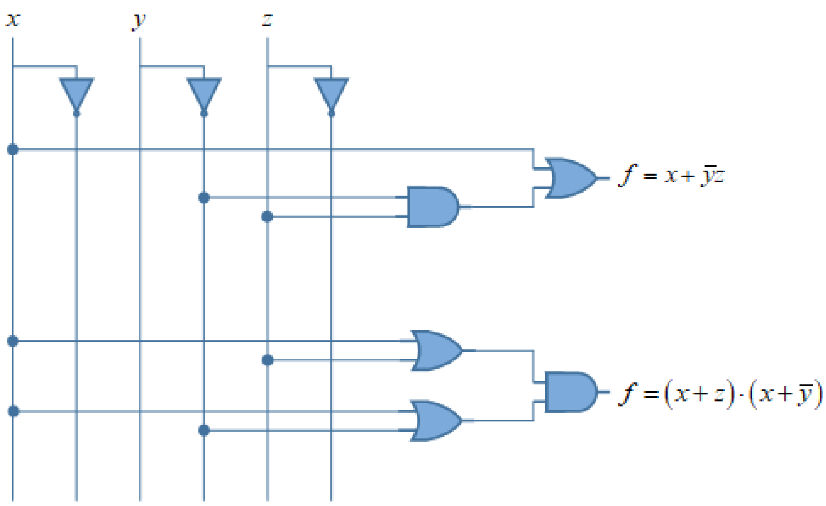
P skup označava jedinice, a Q skup označava nule.

Rešenje:

Može se primetiti iz opisa funkcije da funkcija ima 3 ulazne promenljive (x, y, z), što znači da ima ukupno 8 kombinacija.

Dat je P skup indeksa, tako da kombinacije sa indeksima 1, 4, 5, 6, 7 imaju 1 na izlazu, te će preostale kombinacije imati 0 na izlazu.

Može se primetiti da je ovo upravo funkcija iz prethodnih primera, tako da ostaje samo da se realizuje kroz logička kola. Na slici data je realizacija kroz minimalnu DNF i KNF funkciju.



Slika 2.5 Realizacija logičkih funkcija kroz logička kola (DNF i KNF) [Izvor: Autor].

Poglavlje 3

Logičke komponente

DEKODER 3/8

Dekoder predstavlja logičko kolo koje uzima n-bitni broj kao ulazni signal i postavlja (selektuje) tačno jedna od 2n izlaza koji postavlja na jedinicu.

Dekoder (en. decoder) predstavlja logičko kolo koje uzima n-bitni broj kao ulazni signal i postavlja (selektuje) tačno jedna od 2n izlaza koji postavlja na jedinicu.

Dekoder se koristi, između ostalog, pri selektovanju tačno određenog čipa na RAM memoriji.

Logičko kolo koje koduje signal i suprotno je dekoderu naziva se koder.

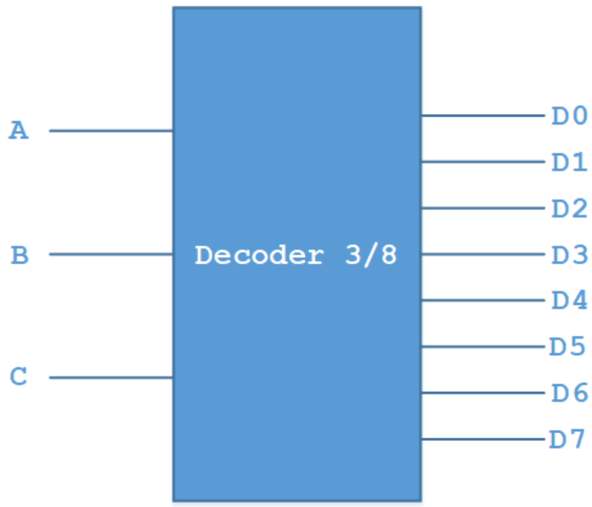
Primer: Dekoder 3/8

Dekoder 3/8 na osnovu kodovanog signala sa tri bita (A, B, i C) selektuje odgovarajući izlaz (D0 - D7), i postavlja vrednost na jedinicu, dok su ostali izlazi neaktivni, tj. na nuli.

Tablica istinitosti ovakvog dekodera data je u nastavku:

A	B	C	D0	D1	D2	D3	D4	D5	D6	D7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

Na slici je prikazan simbol dekodera 3/8.



Slika 3.1 Simbol dekodera 3/8 [Izvor: Autor].

REALIZACIJA DEKODERA 3/8

Na osnovu tablice istinitosti, može se primetiti da dekoder ima 3 ulaza i 8 izlaza od koji se svaki posebno računa.

Na osnovu tablice istinitosti, može se primetiti da dekoder ima 3 ulaza i 8 izlaza.

Prateći tablicu istinitosti, funkcije izlaza se mogu izračunati na sledeći način:

$$D_0 = \bar{A} \cdot \bar{B} \cdot \bar{C}$$
$$D_0 = \bar{A} \cdot \bar{B} \cdot C$$
$$D_0 = \bar{A} \cdot B \cdot \bar{C}$$
$$D_0 = \bar{A} \cdot B \cdot C$$
$$D_0 = A \cdot \bar{B} \cdot \bar{C}$$
$$D_5 = A \cdot \bar{B} \cdot C$$
$$D_6 = A \cdot B \cdot \bar{C}$$
$$D_7 = A \cdot B \cdot C$$

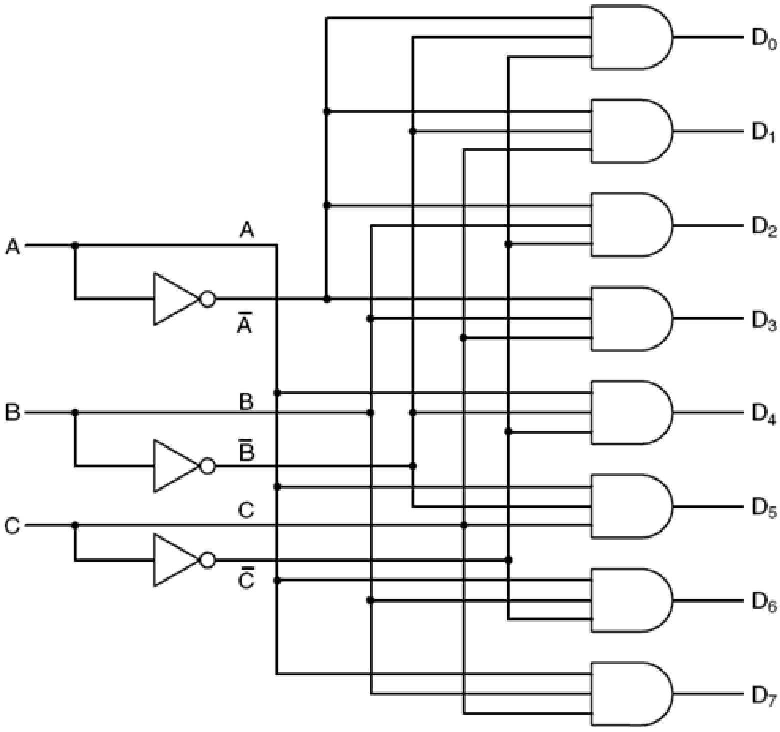
Pitanja:

U kojojsu formi napisani izlazi D0-D7?

Zadatak za samostalni rad:

Izračunati izlaze dekodera u drugoj formi i napraviti realizaciju druge forme kroz odgovarajuće logička kola.

Realizacija dekodera 3/8 kroz invertore i logička AND kola data je na slici.



Slika 3.2 Realizacija dekodera 3/8. [Izvor: Autor]

MULTIPLEKSER

Multiplekser se često koristi kao konvertor paralelnog prenosa u serijski prenos podataka.

Multiplekser (en. multiplexer, *MUX*) predstavlja logičko kolo sa **2$^{\mathbf{n}}$** ulaza, jednim izlazom i **n** kontrolnih ulaza koji selektuju određeni izlaz. Multiplekser se često koristi kao konvertor paralelnog prenosa u serijski prenos podataka. Kolo inverzno multiplekseru je demultiplekser.

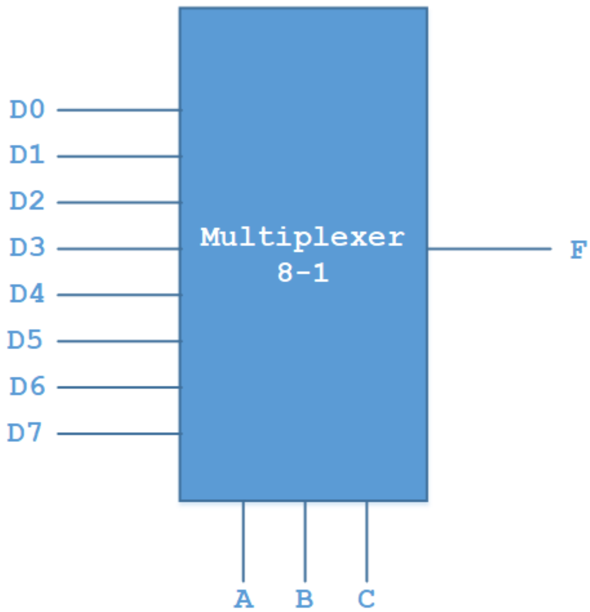
Primer: Multiplekser 8-1

Multiplekser 8-1 ima 8 ulaza, 3 kontrolna ulaza i 1 izlaz. Tablica istinitosti ovakvog multipleksera data je u nastavku:

```
| A | B | C | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | F |
|---|---|---|----|----|----|----|----|----|----|----|---|
| 0 | 0 | 0 | 0  | X  | X  | X  | X  | X  | X  | X  | 0 |
| 0 | 0 | 0 | 1  | X  | X  | X  | X  | X  | X  | X  | 1 |
|---|---|---|----|----|----|----|----|----|----|----|---|
| 0 | 0 | 1 | X  | 0  | X  | X  | X  | X  | X  | X  | 0 |
| 0 | 0 | 1 | X  | 1  | X  | X  | X  | X  | X  | X  | 1 |
|---|---|---|----|----|----|----|----|----|----|----|---|
| 0 | 1 | 0 | X  | X  | 0  | X  | X  | X  | X  | X  | 0 |
| 0 | 1 | 0 | X  | X  | 1  | X  | X  | X  | X  | X  | 1 |
|---|---|---|----|----|----|----|----|----|----|----|---|
```

```
| 0 | 1 | 1 | X  | X  | X  | 0  | X  | X  | X  | X  | 0 |
| 0 | 1 | 1 | X  | X  | X  | 1  | X  | X  | X  | X  | 1 |
|---|---|---|----|----|----|----|----|----|----|----|---|
| 1 | 0 | 0 | X  | X  | X  | X  | 0  | X  | X  | X  | 0 |
| 1 | 0 | 0 | X  | X  | X  | X  | 1  | X  | X  | X  | 1 |
|---|---|---|----|----|----|----|----|----|----|----|---|
| 1 | 0 | 1 | X  | X  | X  | X  | X  | 0  | X  | X  | 0 |
| 1 | 0 | 1 | X  | X  | X  | X  | X  | 1  | X  | X  | 1 |
|---|---|---|----|----|----|----|----|----|----|----|---|
| 1 | 1 | 0 | X  | X  | X  | X  | X  | X  | 0  | X  | 0 |
| 1 | 1 | 0 | X  | X  | X  | X  | X  | X  | 1  | X  | 1 |
|---|---|---|----|----|----|----|----|----|----|----|---|
| 1 | 1 | 1 | X  | X  | X  | X  | X  | X  | X  | 0  | 0 |
| 1 | 1 | 1 | X  | X  | X  | X  | X  | X  | X  | 1  | 1 |
|---|---|---|----|----|----|----|----|----|----|----|---|
```

Na slici je prikazan simbol multipleksera 8-1.



Slika 3.3 Simbol multipleksera 1-8 [Izvor: Autor].

REALIZACIJA MULTIPLEKSERA 8-1

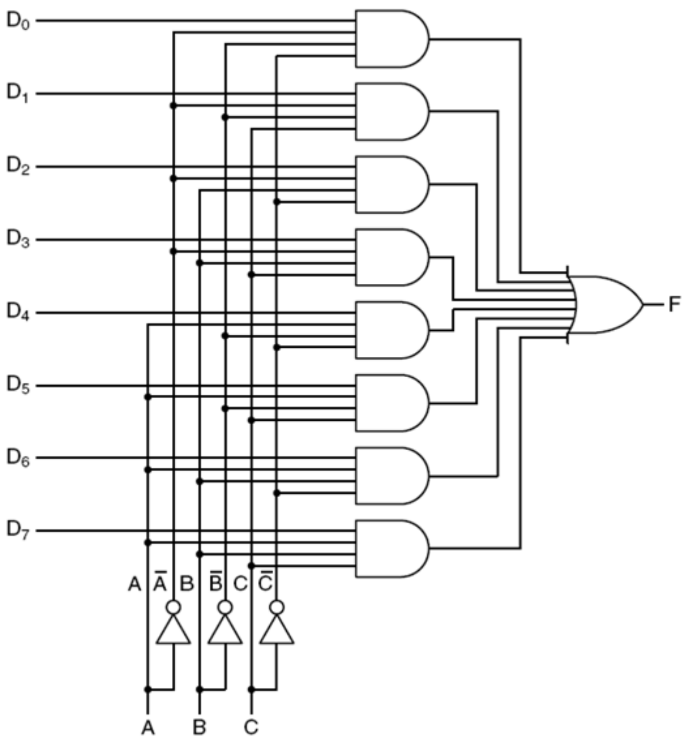
Na osnovu tablice istinitosti, može se primetiti da multiplekser ima 8 ulaza, 1 izlaz i 3 kontrolna ulaza.

Na osnovu tablice istinitosti, može se primetiti da dekoder ima 3 ulaza i 8 izlaza.

Prateći tablicu istinitosti, funkcija izlaza se može izračunati na sledeći način:

$$F = \bar{A}\cdot\bar{B}\cdot\bar{C}\cdot D_0 + \bar{A}\cdot\bar{B}\cdot C\cdot D_1 + \bar{A}\cdot B\cdot\bar{C}\cdot D_2 + \bar{A}\cdot B\cdot C\cdot D_3$$
$$+A\cdot\bar{B}\cdot\bar{C}\cdot D_4 + A\cdot\bar{B}\cdot C\cdot D_5 + A\cdot B\cdot\bar{C}\cdot D_6 + A\cdot B\cdot C\cdot D_7$$

Realizacija multipleksera 8-1 kroz invertore i logička AND i OR kola data je na slici.



Slika 3.4 Realizacija multipleksera 8-1. [Izvor: Autor]

KOMPARATOR

Komparator predstavlja logičko kolo koje upoređuje dve ulazne reči.

Komparator (en. comparator) predstavlja logičko kolo koje upoređuje dve ulazne reči iste dužine

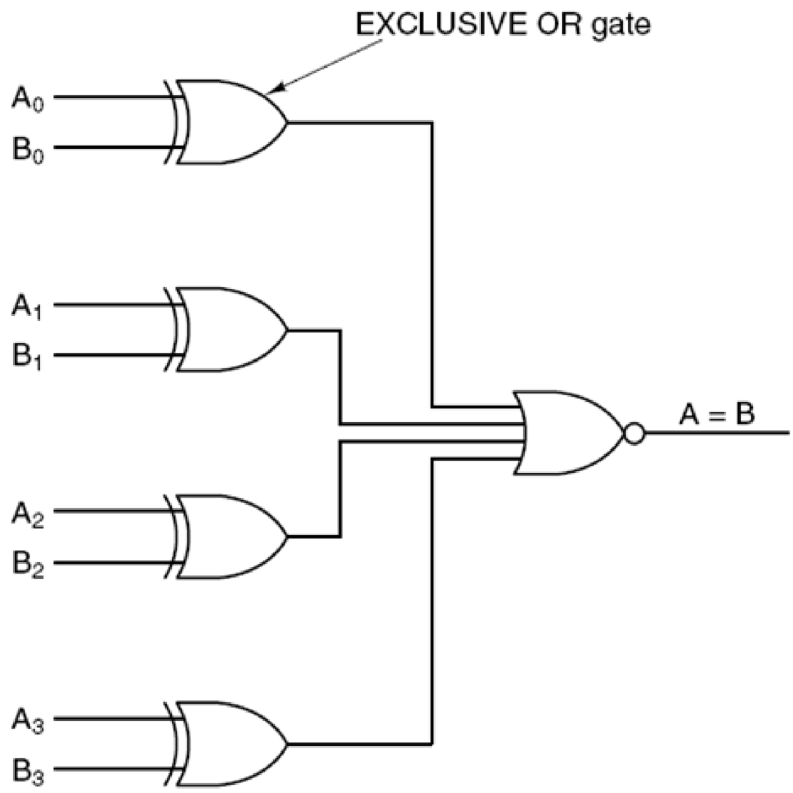
Ovo kolo bazirano je logičkim kolima *isključivo ILI* (*XOR*).

Primer: 4-bitni komparator.

Četvorobitni komparator ima 8 ulaza (za bitove A0-A3 i B0-B3), i jedan izlaz koji daje jedinicu ako su svi bitovi sa istim indeksom jednaki.

Zadatak za samostalni rad:

Napisati tablicu istinitosti za komparator dve četvorobitne reči i izračunati funkciju na izlazu.



Slika 3.5 Realizacija četvorobitnog komparatora [Izvor: Autor].

Poglavlje 4

Aritmetičke komponente

POMERAČ

Pomerač predstavlja aritmetičko kolo koje pomera bitove u levo ili u desno.

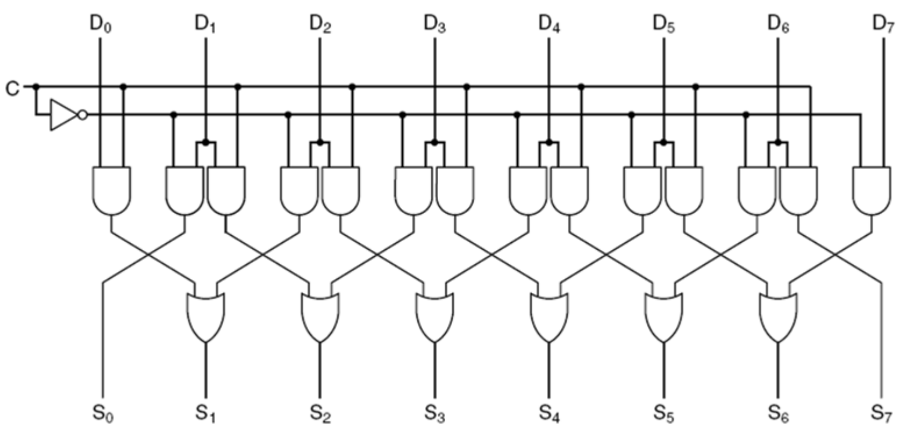
Pomerač ili šifter (en. shifter) predstavlja aritmetičko kolo koje pomera unete bitove (D0-D7) za jedno mesto u levo ili u desno (S0-S7).

Poseduje kontrolnu liniju C koje određuje pomeranje u levo ili u desno.

Pitanje:

Zašto ovo kolo spada u aritmetička kola ako pomera bitove u levo ili u desno?

Na slici je prikazana realizacija pomerača sa 8 bitova:



Slika 4.1 Realizacija pomerača od 8 bitova [Izvor: Autor].

SABIRAČ

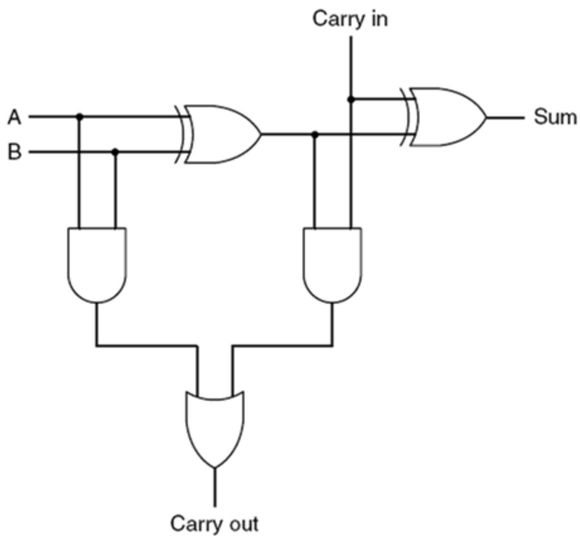
Sabirač predstavlja aritmetičko kolo koje sabira dva bita, daje na izlazu sumu i, ukoliko postoji, i prenet bit.

Sabirač (en. adder) predstavlja aritmetičko kolo koje sabira dva bita. Na izlazu daje zbir u bitu najmanje značnosti, i poseduje još jedan izlaz za tzv. carry bit, odnosno za prenet bit.

Ukoliko sabirač ima poseban ulaz za preneti bit iz drugog sabirača, onda je u pitanju *potpuni sabirač* (en. full adder). Ukoliko su samo ulazi dva bita, onda je to *polu-sabirač* (en. half adder).

Na slici je prikazana tablica istinitosti, kao i realizacija potpunog sabirača kroz logička kola.

A	B	Carry in	Sum	Carry out
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1



Slika 4.2 Tablica istinitosti i realizaciaj potpunog sabirača [Izvor: Autor].

Poglavlje 5

Aritmetičko-Logička jedinica (ALU)

STRUKTURA ALU

Aritmetičko-logička jedinica - ALU sastoji se od međusobno povezanih aritmetičkih i logičkih kola.

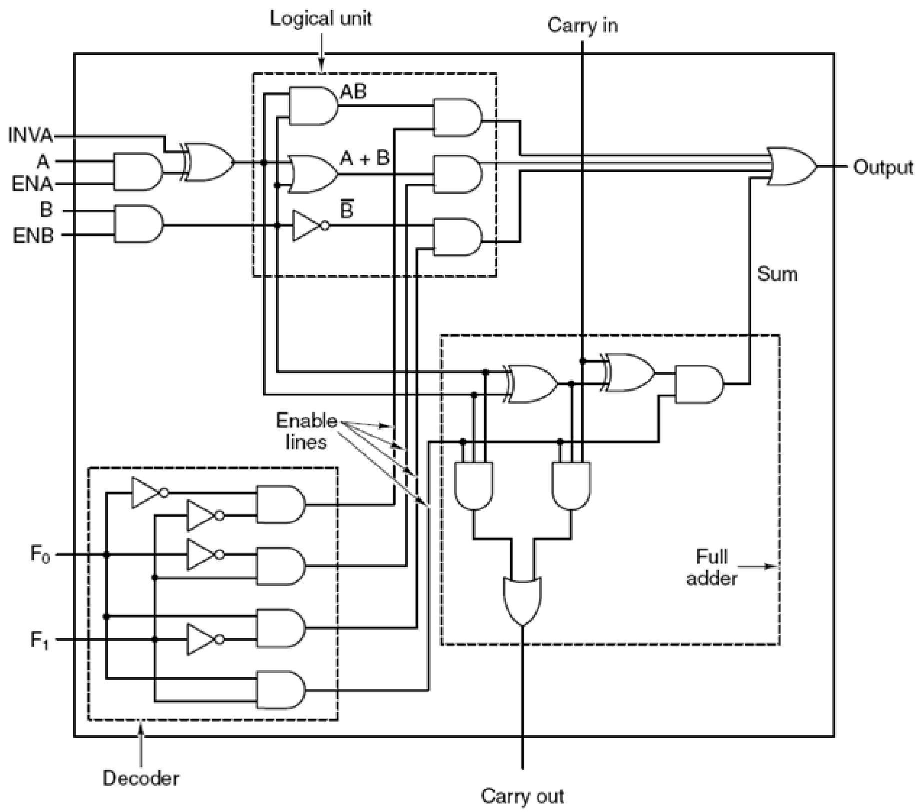
Aritmetičko-logička jedinica - ALU sastoji se od međusobno povezanih aritmetičkih i logičkih kola.

Jednostavna aritmetičko-logička jedinica nad rečima jednog bita može izračunati jednu od četiri funkcije:

- A ***AND*** B (logičko množenje)
- A ***OR*** B (logičko sabiranje)
- B’ (inverzija)
- A + B (aritmetičko sabiranje)

Odabir funkcije vrši se kontrolnim signala F0 i F1 koje se dovode na dekoder.

Na slici je prikazana jednostavna ALU za dva ulazna bita A i B.



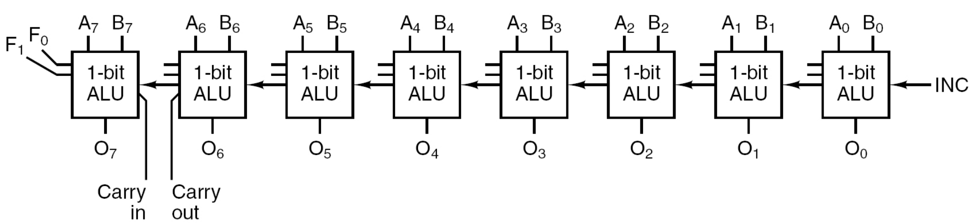
Slika 5.1 Ralizacija ALU sa dva ulazna bita [Izvor: Autor].

POVEZIVANJE JEDNOBITNIH ALU

Rednim povezivanjem jednobitnih ALU moguće je dobiti višebitnu ALU.

Rednim povezivanjem jednobitnih ALU moguće je dobiti višebitnu ALU.

Na slici je prikazana 8-bitna ALU sačinjena od 8 jednobitnih ALU.



Slika 5.2 8-bitna ALU [Izvor: Autor]

Poglavlje 6

Pokazne vežbe

MINIMIZACIJA LOGIČKIH FUNKCIJA - VEŽBE

Pokazne vežbe okvirno traju 90 minuta.

Primer #1 (50 minuta)

Data je sledeća funkcija:

Uraditi sledeće

1. Odrediti broj ulaza,
2. Napisati tablicu istinitosti za funkciju,
3. Uraditi minimalizaciju funkcije koristeći Karnoove mape u DNF formi,
4. Uraditi minimalizaciju funkcije koristeći Karnoove mape u KNF formi,
5. Realizovati minimalizovanu funkciju u DNF kroz logička kola,
6. Realizovati minimalizovanu funkciju u KNF kroz logička kola,

$$f\left(w,x,y,z\right)=^{P}\left(1,3,4,5,7,10,11,12,15\right)$$

Rešenje:

1. Funkcija ima 4 ulaza jer ima četiri ulazne promenljive (w, x, y, z), što znači da će imati ukupno 16 kombinacija sa indeksima od 0 do 15.

Tablica istinitosti na osnovu P skupa je sledeća:

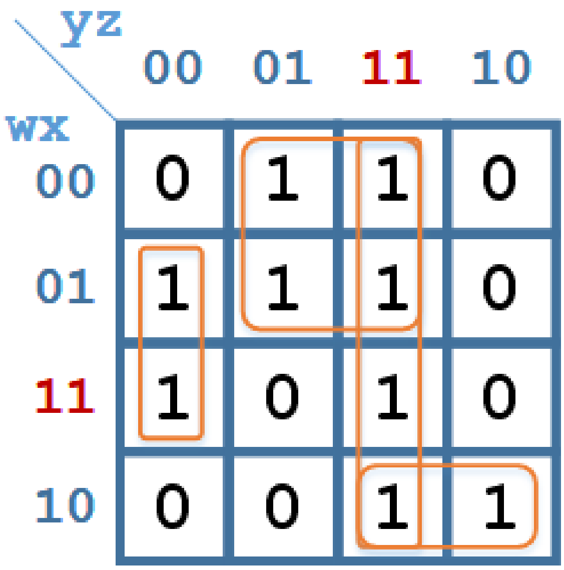
indeks	w	x	y	z	f
0	0	0	0	0	0
1	0	0	0	1	1
2	0	0	1	0	0
3	0	0	1	1	1
4	0	1	0	0	1
5	0	1	0	1	1
6	0	1	1	0	0
7	0	1	1	1	1
8	1	0	0	0	0
9	1	0	0	1	0
10	1	0	1	0	1
11	1	0	1	1	1
12	1	1	0	0	1
13	1	1	0	1	0
14	1	1	1	0	0
15	1	1	1	1	1

Slika 6.1 Tablica istinitosti za funkciju iz primera #1 [Izvor: Autor]

PRIMER #1 - MINIMALIZACIJA FUNKCIJE

Primer #1 - nastavak zadatka -minimizacija funckije

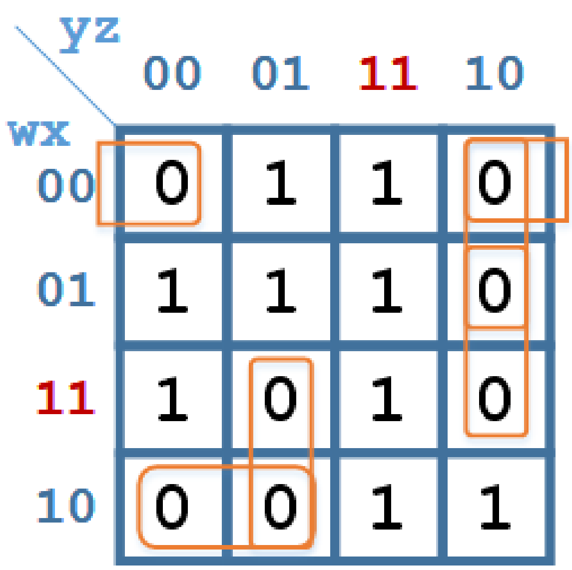
3. Crtanjem, popunjavanjem Karnoovih mapa i selektovanjem jedinica dobijamo minimalizovanu funkciju u DNF formi:



Slika 6.2 Minimizacija funkcije u DNF formi [Izvor: Autor].

$$f = xz + yz + x\bar{y}\bar{z} + w\bar{x}z$$

4. Crtanjem, popunjavanjem Karnoovih mapa i selektovanjem nula dobijamo minimalizovanu funkciju u KNF formi:



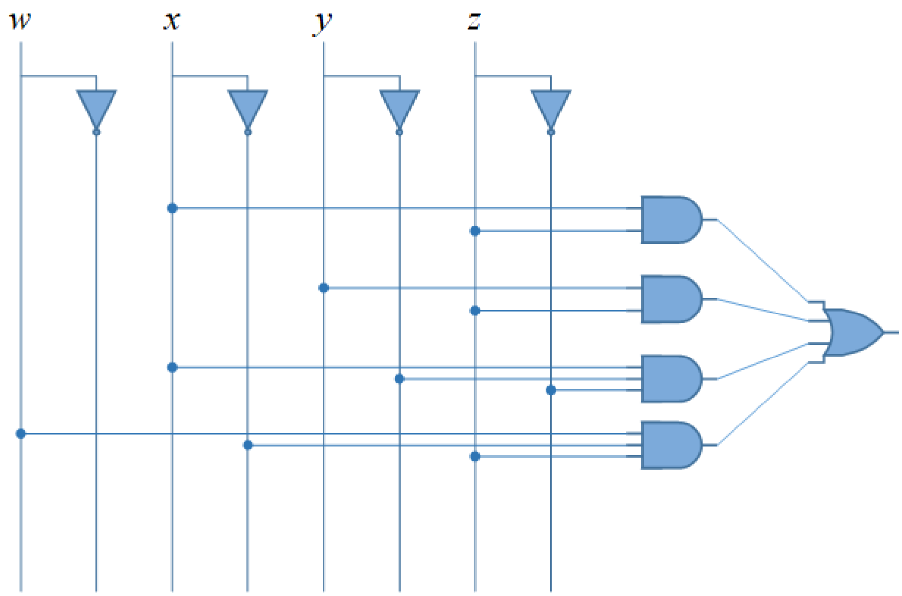
Slika 6.3 inimizacija funkcije u KNF formi [Izvor: Autor].

$$f = (w + x + z) \cdot (\bar{w} + x + y)$$
$$\cdot (\bar{w} + y + \bar{z}) \cdot (w + \bar{y} + z) \cdot (\bar{x} + \bar{y} + z)$$

PRIMER #1 - REALIZACIJA FUNKCIJE

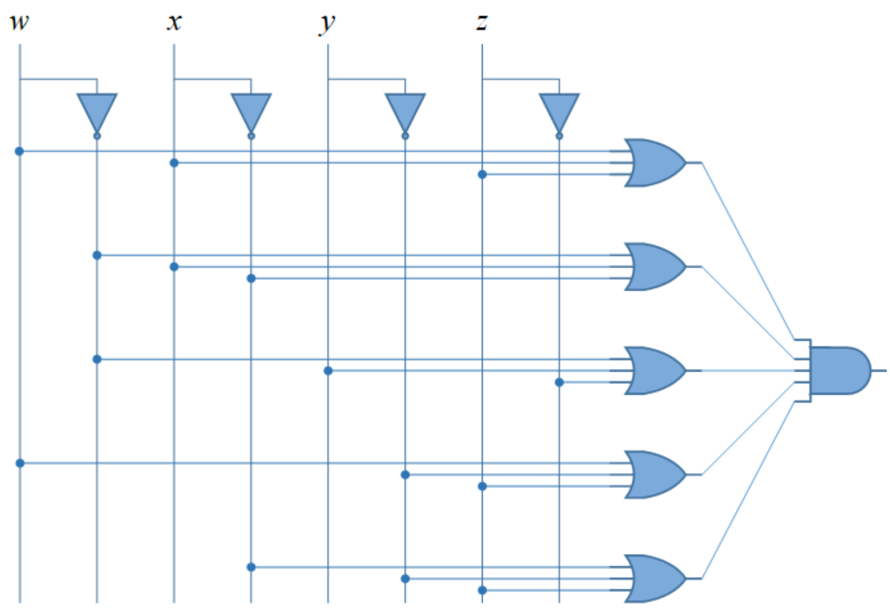
Ista funkcija se može realizovati kroz logička kola bilo u DNF ili KNF formi.

5. Realizaija funkcije u DNF formi uz pomoć logičkih AND i OR kola.



Slika 6.4 Realizacija funkcije u DNF formi [Izvor: Autor].

6. Realizaija funkcije u KNF formi uz pomoć logičkih OR i AND kola.



Slika 6.5 Realizacija funkcije u KNF formi [Izvor: Autor].

PRIMER #2 - MINIMIZACIJA LOGIČKE FUNKCIJE

Primer #2 okvirno traje 40 minuta

Primer #2 (40 minuta)

Data je funkcija u PDNF.

$$f = \bar{A}\bar{B}\bar{C}\bar{D} + \bar{A}B\bar{C}\bar{D} + \bar{A}\bar{B}C\bar{D} + \bar{A}BC\bar{D} + ABC\bar{D} + ABCD + A\bar{B}CD + A\bar{B}C\bar{D}$$

Uraditi sledeće:

1. Odrediti broj ulaznih promenljivih
2. Nacrtati tablicu istinitosti za funkciju
3. Minimalizovati funkciju u DNF formi korišćenjem Karnoovih mapa
4. Realizovati funkciju kroz logička kola

Rešenje:

1. Funkcija ima 4 ulaznih promenljivih, A, B, C, D

2. Na osnovu funkcije u PDNF obliku moguće je direktno napraviti tablicu istinitosti, što je prikazano slikom.

Pitanje:

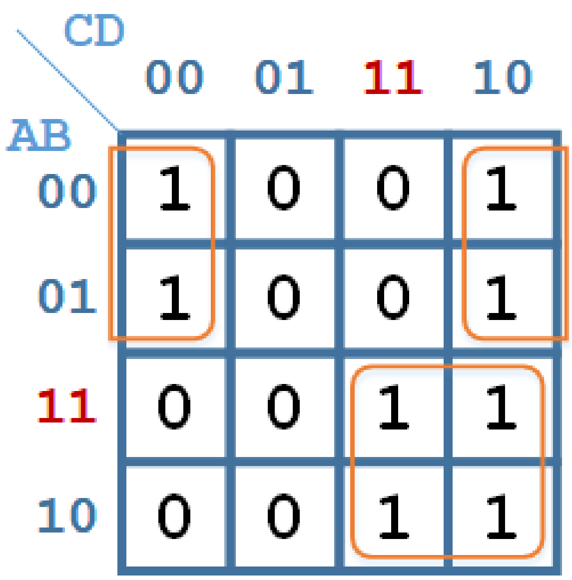
Da li je moguće napraviti tablicu istinitosti ukoliko je data funkcija u minimalnom obliku?

indeks	A	B	C	D	f
0	0	0	0	0	1
1	0	0	0	1	0
2	0	0	1	0	1
3	0	0	1	1	0
4	0	1	0	0	1
5	0	1	0	1	0
6	0	1	1	0	1
7	0	1	1	1	0
8	1	0	0	0	0
9	1	0	0	1	0
10	1	0	1	0	1
11	1	0	1	1	1
12	1	1	0	0	0
13	1	1	0	1	0
14	1	1	1	0	1
15	1	1	1	1	1

Slika 6.6 Tablica istinitosti za funkciju iz primera #2 [Izvor: Autor]

PRIMER #2 - MINIMIZACIJA FUNKCIJE

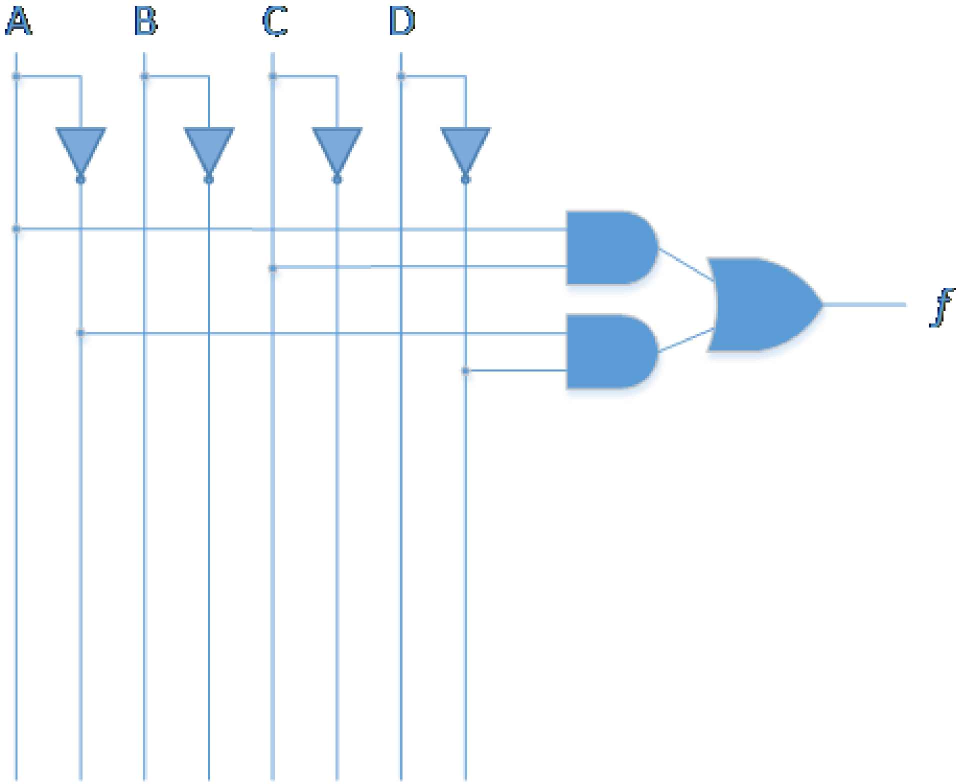
Primer #2 - nastavak zadatka -minimizacija funckije



Slika 6.7 Minimizacija funkcije u DNF formi [Izvor: Autor].

Mimimizovana funkcija ima sledeći oblik:

$$f = AC + \bar{A}\bar{D}$$



Slika 6.8 Realizacija funkcije u DNF formi [Izvor: Autor].

Poglavlje 7

Primeri za samostalni rad

ZADACI ZA SAMOSTALNI RAD IZ LEKCIJE #3

Zadaci se ukupno rade za okvirno 30 minuta

Zadatak za samostalni rad #1

Korišćenjem De Morganovih zakona predstaviti sledeću funkciju u DNF i KNF:

$$f = \overline{A + B\left(\overline{C + \overline{D}}\right)}$$

Zadatak za samostalni rad #2

Data je funkcija u DNF. Prebaciti funkciju PDNF.

$$f = A + \overline{AB}C + A\overline{BC} + BC$$

Zadatak za samostalni rad #3

Pretvoriti broj 0056 u pakovani BCD sistem. BCD broj predstavljaće izlaz logičke funkcije sa četiri ulaza.

Uraditi sledeće:

1. Napisati tablicu istinitosti za funkciju,
2. Uraditi minimalizaciju funkcije koristeći Karnoove mape u DNF formi
3. Uraditi minimalizaciju funkcije koristeći Karnoove mape u KNF formi
4. Realizovati minimalizovanu funkciju u DNF kroz logička kola
5. Realizovati minimalizovanu funkciju u KNF kroz logička kola

Poglavlje 8

Domaći zadatak

DOMAĆI ZADATAK #3

Domaći zadatak #2 okvirno se radi 40 minuta.

Napomena:

Svaki student dobija jedinstveni domaći zadatak jer zavisi od broja indeksa.

Domaći zadatak #3:

Vaš broj indeksa konvertovati u pakovani BCD sistem.

Dobijeni BCD broj predstavljaće izlaz logičke funkcije sa četiri ulaza.

Uraditi sledeće:

1. Napisati tablicu istinitosti za funkciju,
2. Iz tablice istinitosti napisati funkciju u PDNF,Iz tablice istinitosti napisati funkciju u PKNF,
3. Uraditi minimalizaciju funkcije koristeći Karnoove mape u DNF formi,
4. Uraditi minimalizaciju funkcije koristeći Karnoove mape u KNF formi,
5. Realizovati minimalizovanu funkciju u DNF kroz logička kola,
6. Realizovati minimalizovanu funkciju u KNF kroz logička kola.

Predaja domaćeg zadatka:

Domaći zadatak slati odgovarajućem predmetnom asistentu, sa predmetnim profesorom u CC.

Predati domaći zadatak koristeći .doc/docx uputstvo dato u prvoj lekciji.

Poglavlje 9

Zaključak

ZAKLJUČAK

Rezime lekcije #3

U lekciji #3 najpre je bilo reči o logičkim kolima i Bulovoj algebri sa postulatima i svojstvima.

Zatim, bilo je reči o logičkim funkcijama, standardnim formama logičkih funkcija i o proširenju u potpune normalne forme.

Da bi se logička funkcija realizovala kroz logička kola, potrebno je najpre tu logičku funkciju minimizovati, pa je uvedena tehnika minimizacije korišćenjem Karnoovih mapa.

Predstavljena su konkretna logička dekoder, multiplekser, komparator, kao i aritmetička kola pomerača i sabirača.

Spajanjem ovih kola tj. komponenti, dobija se aritmetičko-logička jedinica.

Na vežbama obrađivala se minimizacija logičkih funkcija i realizacija kroz logička kola.

Literatura:

A. Tanenbaum, Structured Computer Organization, Chapter 03, pp. 147 - 168.