Dragana Prokin
Vera V. Petrović
Milan Mijalković

ZBIRKA ZADATAKA IZ OSNOVA RAČUNARSKE TEHNIKE

Visoka škola elektrotehnike i računarstva strukovnih studija
Beograd, 2019.

Autori:	dr Dragana Prokin dr Vera V. Petrović dr Milan Mijalković
Recezenti:	dr Slobodan Obradović dr Zoran Banjac
Izdavač:	Visoka škola elektrotehnike i računarstva strukovnih studija, Beograd Vojvode Stepe 283
Tehnička obrada:	Gabrijela Dimić Divna Popović
Korice:	Gabrijela Dimić
Tiraž:	30
Štampa:	Razvojno-istraživački centar grafičkog inženjerstva TMF, Beograd
Godina izdanja:	2019.
Izdanje:	Četvrto izmenjeno izdanje

CIP - Каталогизација у публикацији - Народна библиотека Србије, Београд

004(075.8)(076)

ПРОКИН, Драгана, 1963-
Zbirka zadataka iz osnova računarske tehnike / Dragana Prokin, Vera V. Petrović, Milan Mijalković. - 4. izmenjeno izd. - Beograd : Visoka škola elektrotehnike i računarstva strukovnih studija, 2019 (Beograd : Razvojno-istraživački centar grafičkog inženjerstva TMF). - 161 str. : graf. prikazi ; 25 cm

Tiraž 30. - Bibliografija: str. 161.

ISBN 978-86-7982-305-2

1. Петровић, Вера В., 1965- [аутор] 2. Мијалковић, Милан, 1957- [аутор]

а) Рачунарство - Задаци
COBISS.SR-ID 274510348

S A D R Ž A J

1. BROJNI SISTEMI I KONVERZIJA BROJEVA IZ JEDNOG BROJNOG SISTEMA U DRUGI

Brojni sistemi:

- predstavljaju način prikazivanja bilo kog broja pomoću niza simbola koji se nazivaju cifre brojnog sistema.
- skup pravila po kojima se realizuju osnovne operacije nad brojevima.

U **pozicionom** (**težinskom**) brojnom sistemu vrednost cifre zavisi od pozicije koju cifra ima u zapisu brojne vrednosti.

Za bilo koji broj x u težinskom brojnom sistemu važi zapis:

$x = a_R \cdot S^R + a_{R-1} \cdot S^{R-1} + ... + a_1 \cdot S^1 + a_0 \cdot S^0 + a_{-1} \cdot S^{-1} + ... + a_{-P} \cdot S^{-P}$

S = osnova (baza) brojnog sistema

S^i = težina cifre u brojnom sistemu

i = pozicija cifre (R, R-1, …, 1, 0, -1,…, -P)

a_R , a_{R-1} , ..., a_1 , a_0 , a_{-1} , ..., a_{-P} su cifre broja koje pripadaju skupu {0, 1, …, S-1}

Sažeti oblik prikazivanja broja x:

$x = a_R\ a_{R-1} \ldots a_1\ a_0,\ a_{-1} \ldots a_{-P}.$

Decimalni brojni sistem (DEC) je težinski.

Svaki broj x iz DEC brojnog sistema može da se predstavi kao:

$x = a_R \cdot 10^R + a_{R-1} \cdot 10^{R-1} + ... + a_1 \cdot 10^1 + a_0 \cdot 10^0 + a_{-1} \cdot 10^{-1} + ... + a_{-P} \cdot 10^{-P}$

S = 10 osnova (baza) brojnog sistema

a_R, a_{R-1}, ...,a_1, a_0, a_{-1}, ...,a_{-P} su cifre broja koje pripadaju skupu {0, 1, …, 9}

Konverzija brojeva iz drugih brojnih sistema u decimalan brojni sistem (DEC) obavlja se sumiranjem elementarnih proizvoda cifara u zapisu broja i njihovih težinskih koeficijenata:

- **Konverzija iz HEX u DEC brojni sistem**

$x_{(10)} = a_R \cdot 16^R + a_{R-1} \cdot 16^{R-1} + ... + a_1 \cdot 16^1 + a_0 \cdot 16^0 + a_{-1} \cdot 16^{-1} + ... + a_{-P} \cdot 16^{-P}$

S = 16 osnova (baza) brojnog sistema

a_R, a_{R-1}, ..., a_1, a_0, a_{-1},...,a_{-P} su cifre broja koje pripadaju skupu {0,..., 9, A, B, C, D, E, F}

- **Konverzija iz OCT u DEC brojni sistem**

$x_{(10)} = a_R \cdot 8^R + a_{R-1} \cdot 8^{R-1} + ... + a_1 \cdot 8^1 + a_0 \cdot 8^0 + a_{-1} \cdot 8^{-1} + ... + a_{-P} \cdot 8^{-P}$

S = 8 osnova (baza) brojnog sistema

a_R , a_{R-1}, ...,a_1, a_0, a_{-1}, ...,a_{-P} su cifre broja koje pripadaju skupu {0, 1, …, 7}

- **Konverzija iz BIN u DEC brojni sistem**

$x_{(10)} = a_R \cdot 2^R + a_{R-1} \cdot 2^{R-1} + ... + a_1 \cdot 2^1 + a_0 \cdot 2^0 + a_{-1} \cdot 2^{-1} + ... + a_{-P} \cdot 2^{-P}$

S = 2 osnova (baza) brojnog sistema

a_R , a_{R-1}, ..., a_1, a_0, a_{-1}, ...,a_{-P} su cifre broja koje pripadaju skupu {0, 1}

Primer 1. Izvršiti konverziju heksadecimalnog broja $x_{(16)}$ = 2E3A$_{(16)}$ u decimalni brojni sistem, $x_{(16)} \rightarrow x_{(10)}$.

Rešenje:

$x_{(16)}$ = 2E3A$_{(16)}$

$x_{(10)} = 2 \cdot 16^3 + 14 \cdot 16^2 + 3 \cdot 16^1 + 10 \cdot 16^0 = \mathbf{11834_{(10)}}$

Primer 2. Izvršiti konverziju oktalnog broja $x_{(8)} = 3217_{(8)}$ u decimalni brojni sistem, $x_{(8)} \rightarrow x_{(10)}$.

Rešenje:

$x_{(8)} = 321_{(8)}$

$x_{(10)} = 3 \cdot 8^3 + 2 \cdot 8^2 + 1 \cdot 8^1 = \mathbf{1672_{(10)}}$

Primer 3. Izvršiti konverziju 8-bitnog binarnog broja $x_{(2)} = 10111011_{(2)}$ u decimalni brojni sistem, $x_{(2)} \rightarrow x_{(10)}$.

Rešenje:

$x_{(2)} = 10111011_{(2)}$

$x_{(10)} = 1 \cdot 2^7 + 0 \cdot 2^6 + 1 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = \mathbf{187_{(10)}}$

Konverzija brojeva iz DEC brojnog sistema u brojni sistem sa osnovom S obavlja se:

- metodom **sukcesivnih deljenja** celobrojnog dela sa osnovom brojnog sistemametodom **sukcesivnih množenja** decimalnog (razlomljenog) dela sa osnovom brojnog sistema **S**

Primer 4. Decimalni broj $x_{(10)} = 240{,}375_{(10)}$ pretvoriti u binarni, sa 3 decimale tačnosti $x_{(10)} \rightarrow x_{(2)}$.

Rešenje:

240 : 2 = 120	0	↑
120 : 2 = 60	0	
60 : 2 = 30	0	
30 : 2 = 15	0	
15 : 2 = 7	1	
7 : 2 = 3	1	
3 : 2 = 1	1	
1 : 2 = 0	1	

0,375 · 2 = 0,75	0	
0,75 · 2 = 1,5	1	
0,5 · 2 = 1,0	1	↓

$240{,}375_{(10)} = \mathbf{11110000{,}011_{(2)}}$

Primer 5. Decimalni broj x(10) = 4859,237(10) pretvoriti u binarni, sa 5 decimala tačnosti $x_{(10)} \rightarrow x_{(2)}$.

Rešenje:

4859 : 2 = 2429	1	0,237 · 2 = 0,474	0
2429 : 2 = 1214	1	0,474 · 2 = 0,948	0
1214 : 2 = 607	0	0,948 · 2 = 1,896	1
607 : 2 = 303	1	0,896 · 2 = 1,792	1
303 : 2 = 151	1	0,792 · 2 = 1,584	1
151 : 2 = 75	1	0,584 · 2 = 1,168	1
75 : 2 = 37	1		
37 : 2 = 18	1		
18 : 2 = 9	0		
9 : 2 = 4	1		
4 : 2 = 2	0		
2 : 2 = 1	0		
1 : 2 = 0	1		

$4859{,}237_{(10)} = \mathbf{1001011111011{,}00111_{(2)}}$

Primer 6. Decimalni broj $x_{(10)} = 4365{,}136_{(10)}$ pretvoriti u oktalni, sa 4 decimale tačnosti $x_{(10)} \rightarrow x_{(8)}$.

Rešenje:

4365 : 8 = 545	5	0,136 · 8 = 1,088	1
545 : 8 = 68	1	0,088 · 8 = 0,704	0
68 : 8 = 8	4	0,704 · 8 = 5,632	5
8 : 8 = 1	0	0,632 · 8 = 5,056	5
1 : 8 = 0	1		

$4365{,}136_{(10)} = \mathbf{10415{,}1055_{(8)}}$

Primer 7. Decimalni broj $x_{(10)} = 695{,}218_{(10)}$ pretvoriti u oktalni, sa 4 decimale tačnosti $x_{(10)} \rightarrow x_{(8)}$.

Rešenje:

695 : 8 = 86	7	0,218 · 8 = 1,744	1
86 : 8 = 10	6	0,744 · 8 = 5,952	5
10 : 8 = 1	2	0,952 · 8 = 7,616	7
1 : 8 = 0	1	0,616 · 8 = 4,928	4

$695{,}218_{(10)} = \mathbf{1267{,}1574_{(8)}}$

Primer 8. Decimalni broj $x_{(10)} = 845{,}631_{(10)}$ pretvoriti u heksadecimalni, sa 3 decimale tačnosti $x_{(10)} \rightarrow x_{(16)}$.

Rešenje:

845 : 16 = 52	13 = D	0,631 · 16 = 10,096	A
52 : 16 = 3	4	0,096 · 16 = 1,536	1
3 : 16 = 0	3	0,536 · 16 = 8,576	8

$845{,}631_{(10)} = \mathbf{34D{,}A18_{(16)}}$

Primer 9. Decimalni broj $x_{(10)} = 674{,}574_{(10)}$ pretvoriti u heksadecimalni, sa 4 decimale tačnosti $x_{(10)} \rightarrow x_{(16)}$.

Rešenje:

$$\begin{array}{r|l} 674 : 16 = 42 & 2 \\ 42 : 16 = 2 & 10=A \\ 2 : 16 = 0 & 2 \end{array} \uparrow \qquad \begin{array}{r|l} 0{,}574 \cdot 16 = 9{,}184 & 9 \\ 0{,}184 \cdot 16 = 2{,}944 & 2 \\ 0{,}944 \cdot 16 = 15{,}104 & 15=F \\ 0{,}104 \cdot 16 = 1{,}664 & 1 \end{array} \downarrow$$

$$674{,}574_{(10)} = \mathbf{2A2{,}92F1_{(16)}}$$

Primer 10. Decimalni broj $x_{(10)} = 3428{,}435_{(10)}$ pretvoriti u heksadecimalni, sa 4 decimale tačnosti $x_{(10)} \rightarrow x_{(16)}$.

Rešenje:

$$\begin{array}{r|l} 3428 : 16 = 214 & 4 \\ 214 : 16 = 13 & 6 \\ 13 : 16 = 0 & 13=D \end{array} \uparrow \qquad \begin{array}{r|l} 0{,}435 \cdot 16 = 6{,}95 & 6 \\ 0{,}95 \cdot 16 = 15{,}2 & 15=F \\ 0{,}2 \cdot 16 = 3{,}2 & 3 \\ 0{,}2 \cdot 16 = 3{,}2 & 3 \end{array} \downarrow$$

$$3428{,}435_{(10)} = \mathbf{D64{,}6F33_{(16)}}$$

Konverzija brojeva iz **BIN** u **OCT** brojni sistem obavlja se tako što se grupišu po **tri binarne cifre** levo i desno počev od decimalne tačke.

Konverzija brojeva iz **BIN** u **HEX** brojni sistem obavlja se tako što se grupišu po **četiri binarne cifre** levo i desno počev od decimalne tačke.

Primer 11. Binarni broj $x_{(2)} = 11001110{,}01011_{(2)}$ pretvoriti u oktalni, $x_{(2)} \rightarrow x_{(8)}$.

Rešenje:

$011 \mid 001 \mid 110{,}010 \mid 110_{(2)} = \mathbf{316{,}26_{(8)}}$

Primer 12. Binarni broj $x_{(2)} = 11110010110{,}0101011111_{(2)}$ pretvoriti u heksadecimalni, $x_{(2)} \rightarrow x_{(16)}$.

Rešenje:

$0111 \mid 1001 \mid 0110{,}0101 \mid 0111 \mid 1100_{(2)} = \mathbf{796{,}57C_{(16)}}$

Konverzija brojeva iz **OCT** u **BIN** brojni sistem obavlja se tako što se svaka **oktalna** cifra zamenjuje svojim **trocifrenim** binarnim zapisom.

Konverzija brojeva iz **HEX** u **BIN** brojni sistem obavlja se tako što se svaka **heksadecimalna** cifra zamenjuje svojim **četvorocifrenim** binarnim zapisom.

Primer 13. Konvertovati oktalni broj $x_{(8)} = 34752{,}423601_{(8)}$ u binarni, $x_{(8)} \to x_{(2)}$.

Rešenje:

$34752{,}423601_{(8)} = 011\,|\,100\,|\,111\,|\,101\,|\,010{,}100\,|\,010\,|\,011\,|\,110\,|\,000\,|\,001_{(2)}$

U rezultatu konverzije mogu da se izostave nule na krajevima zapisa celbrojnog i razlomljenog dela, tako da rezultat konverzije može da se predstavi u obliku:

$34752{,}423601_{(8)} = \mathbf{11100111101010{,}100010011110000001_{(2)}}$

Primer 14. Konvertovati heksadecimalni broj $x_{(16)} = E1B3C6{,}D4F8_{(16)}$ u binarni, $x_{(16)} \to x_{(2)}$.

Rešenje:

$E1B3C6{,}D4F8_{(16)} = 1110\,|\,0001\,|\,1011\,|\,0011\,|\,1100\,|\,0110{,}1101\,|\,0100\,|\,1111\,|\,1000_{(2)}$

U rezultatu konverzije mogu da se izostave nule na krajevima zapisa celbrojnog i razlomljenog dela, tako da rezultat konverzije može da se predstavi u obliku:

$E1B3C6{,}D4F8_{(16)} = \mathbf{111000011011001111000110{,}1101010011111_{(2)}}$

Konverzija brojeva iz **OCT** u **HEX** brojni sistem vrši se preko binarnog brojnog sistema:

OCT ⟶ **BIN** ⟶ **HEX**

1. Svaka **oktalna** cifra se zamenjuje sa **tri binarne** cifre
2. Grupišu se po četiri binarne cifre ulevo i udesno od decimalne tačke.

Konverzija brojeva iz **HEX** u **OCT** brojni sistem vrši se preko binarnog brojnog sistema:

HEX ⟶ **BIN** ⟶ **OCT**

1. Svaka **heksadecimalna** cifra se zamenjuje sa **četiri binarne** cifre
2. Grupišu se po **tri binarne cifre** ulevo i udesno od decimalne tačke.

Primer 15. Oktalni broj $x_{(8)} = 5716{,}043_{(8)}$ pretvoriti u heksadecimalni, $x_{(8)} \to x_{(16)}$.

Rešenje:

$5716{,}043_{(8)} = 101\ 111\ 001\ 110{,}000\ 100\ 011_{(2)}$

$5716{,}043_{(8)} = 1011\,|\,1100\,|\,1110{,}0001\,|\,0001\,|\,1000_{(2)} = \mathbf{BCE{,}118_{(16)}}$

Primer 16. Konvertovati oktalni broj $x_{(8)} = 4127{,}153_{(8)}$ u heksadecimalni, $x_{(8)} \to x_{(16)}$.

Rešenje:

$4127{,}153_{(8)} = 100\ \ 001\ 010\ 111\ ,\ 001\ 101\ 011_{(2)}$

$4127{,}153_{(8)} = 1000\,|\,0101\,|\,0111,\ 0011\,|\,0101\,|\,1000_{(2)} = \mathbf{857{,}358_{(16)}}$

Primer 17. Heksadecimalni broj $x_{(16)}$ = D5C,13F$_{(16)}$ pretvoriti u oktalni, $x_{(16)} \rightarrow x_{(8)}$.

Rešenje:

D5C,13F$_{(16)}$ = 1101 0101 1100 , 0001 0011 1111$_{(2)}$

D5C,13F$_{(16)}$ = 110 | 101 | 011 | 100,000 | 100 | 111 | 111$_{(2)}$ = **6534,0477$_{(8)}$**

Primer 18. Heksadecimalni broj $x_{(16)}$ = 1CB,81A$_{(16)}$ pretvoriti u oktalni, $x_{(16)} \rightarrow x_{(8)}$.

Rešenje:

1CB,81A$_{(16)}$ = 0001 1100 1011,1000 0001 1010$_{(2)}$

1CB,81A$_{(16)}$ = 000 | 111 | 001 | 011,100 | 000 | 011 | 010$_{(2)}$ = **713,4032$_{(8)}$**

Primer 19. Koji su dekadni brojevi predstavljeni datim brojevima:

a) 10110101$_{(2)}$	d) 1235$_{(8)}$	g) 31B$_{(16)}$
b) 0,111$_{(2)}$	e) 0,14$_{(8)}$	h) 0,A4$_{(16)}$
c) 11,11011$_{(2)}$	f) 24,13$_{(8)}$	i) 1DF,C$_{(16)}$

Rešenje:

a) 10110101$_{(2)}$ = $1{\cdot}2^7+0{\cdot}2^6+1{\cdot}2^5+1{\cdot}2^4+0{\cdot}2^3+1{\cdot}2^2+0{\cdot}2^1+1{\cdot}2^0$=128+32+16+4+1=**181$_{(10)}$**

b) 0,111$_{(2)}$ = $1{\cdot}2^{-1}+1{\cdot}2^{-2}+1{\cdot}2^{-3}$ = **0,875$_{(10)}$**

c) 11,11011$_{(2)}$ = $1{\cdot}2^1+1{\cdot}2^0+1{\cdot}2^{-1}+1{\cdot}2^{-2}+0{\cdot}2^{-3}+1{\cdot}2^{-4}+1{\cdot}2^{-5}$ = **3,84375$_{(10)}$**

d)1235$_{(8)}$ = $1{\cdot}8^3+2{\cdot}8^2+3{\cdot}8^1+5{\cdot}8^0$= 512+128+24+5 = **669$_{(10)}$**

e) 0,14$_{(8)}$ = $1{\cdot}8^{-1}+4{\cdot}8^{-2}$ = **0,1875$_{(10)}$**

f) 24,13$_{(8)}$ = $2{\cdot}8^1+4{\cdot}8^0+1{\cdot}8^{-1}+3{\cdot}8^{-2}$ = 16+4+0,125+0,046875 = **20,171875$_{(10)}$**

g) 31B$_{(16)}$ = $3{\cdot}16^2+1{\cdot}16^1+11{\cdot}16^0$ = 768+16+11 = **795$_{(10)}$**

h) 0,A4$_{(16)}$ = $10{\cdot}16^{-1}+4{\cdot}16^{-2}$ = **0,640625$_{(10)}$**

i) 1DF,C$_{(16)}$ = $1{\cdot}16^2+13{\cdot}16^1+15{\cdot}16^0+12{\cdot}16^{-1}$ = **479,75$_{(10)}$**

Primer 20. Memorija nekog računara podeljena je na sledeće segmente koji su izraženi heksadecimalno u bajtovima:

a) (0000 – BFFF)$_{(16)}$

b) (C000 – CFFF)$_{(16)}$

c) (D000 – FFFF)$_{(16)}$

Koje će decimalne vrednosti u bajtovima imati veličine ovih segmenata?

Rešenje:

a) (0000 – BFFF)$_{(16)}$ = (0 – 49151)$_{(10)}$ $\Rightarrow$ veličina segmenta je **49151**

b) (C000 – CFFF)$_{(16)}$ = (49152 – 53247)$_{(10)}$ $\Rightarrow$ veličina segmenta je **4095**

c) (D000 – FFFF)$_{(16)}$ = (53248 – 65535)$_{(10)}$ $\Rightarrow$ veličina segmenta je **12287**

Primer 21. U decimalnom brojnom sistemu izračunati zbir:

$22120_{(3)}+1531_{(6)}+67_{(9)}+358_{(14)}+10B_{(26)}+9E_{(35)}$.

Rešenje:

$231_{(10)} +415_{(10)}+556_{(10)}+666_{(10)}+687_{(10)}+329_{(10)} = \mathbf{2884_{(10)}}$

Primer 22. Izraziti dekadni broj $550_{(10)}$ u svim brojnim sistemima sa osnovom od 2 do 9.

Rešenje:

$550_{(10)} = 1000100110_{(2)} = 202101_{(3)} = 20212_{(4)} = 4200_{(5)} = 2314_{(6)} = 1414_{(7)} = 1046_{(8)} = \mathbf{671_{(9)}}$

Primer 23. Razlomak $4/5_{(10)}$ prevesti u binarni i heksadecimalni zapis.

Rešenje:

$4/5_{(10)} = 0{,}110011001100..._{(2)} = \mathbf{0{,}CCC..._{(16)}}$

Primer 24. Razlomak $1/7_{(10)}$ napisati kao oktalni, binarni i heksadecimalni broj.

Rešenje:

$1/7_{(10)} = 0{,}1111_{(8)} = 0{,}001001001001_{(2)} = \mathbf{0{,}249249_{(16)}}$

Primer 25. Pretvoriti $3344_{(10)}$ u broj u brojnom sistemu sa osnovom q.

a) q = 4

b) q = 5

c) q = 7

d) q = 9

Rešenje:

a) $\mathbf{310100_{(4)}}$
b) $\mathbf{101334_{(5)}}$
c) $\mathbf{12515_{(7)}}$
d) $\mathbf{4525_{(9)}}$

Primer 26. Mapa operativne memorije nekog računara prikazana je u dekadnoj notaciji:

$(0 – 191)_{(10)}$; $(192 – 199)_{(10)}$; $(200 – 207)_{(10)}$; $(208 – 255)_{(10)}$. Odrediti ekvivalentnu heksadecimalnu notaciju.

Rešenje:

$\mathbf{(00 – BF)_{(16)}}$

$\mathbf{(C0 – C7)_{(16)}}$

$\mathbf{(C8 – CF)_{(16)}}$

$\mathbf{(D0 – FF)_{(16)}}$

Primer 27. Predstaviti sledeće brojeve iz brojnog sistema sa osnovom s u brojnom sistemu sa osnovom q:

a) $11110011{,}1101_{(2)}$ $s = 2$ $q = 8$

b) $110001110{,}001111_{(2)}$ $s = 2$ $q = 16$

c) $614{,}7101_{(8)}$ $s = 8$ $q = 2$

d) $B25{,}A21_{(16)}$ $s = 16$ $q = 2$

e) $3AB{,}43_{(16)}$ $s = 16$ $q = 8$

Rešenje:

a) $\mathbf{363{,}64_{(8)}}$

b) $\mathbf{18E{,}3C_{(16)}}$

c) $\mathbf{110001100{,}111001000001_{(2)}}$

d) $\mathbf{101100100101{,}101000100001_{(2)}}$

e) $\mathbf{1653{,}206_{(8)}}$

Primer 28. Sledeće decimalne brojeve:

a) $124{,}567_{(10)}$	d) $246{,}357_{(10)}$	g) $258{,}401_{(10)}$
b) $456{,}764_{(10)}$	e) $0{,}286_{(10)}$	h) $0{,}025_{(10)}$
c) $645{,}780_{(10)}$	f) $876_{(10)}$	i) $1250{,}67_{(10)}$

pretvoriti sa tri decimale tačnosti u odgovarajuće brojeve:

1) binarnog brojnog sistema, $x_{(10)} \rightarrow x_{(2)}$.

2) oktalnog brojnog sistema, $x_{(10)} \rightarrow x_{(8)}$.

3) heksadecimalnog brojnog sistema, $x_{(10)} \rightarrow x_{(16)}$.

Primer 29. Sledeće decimalne brojeve:

a) $56{,}471_{(10)}$	e) $116{,}229_{(10)}$	i) $89{,}125_{(10)}$	m) $113_{(10)}$
b) $3{,}773_{(10)}$	f) $72{,}025_{(10)}$	j) $51{,}17_{(10)}$	n) $0{,}376_{(10)}$
c) $112{,}34_{(10)}$	g) $108{,}433_{(10)}$	k) $125{,}12_{(10)}$	o) $91_{(10)}$
d) $64{,}82_{(10)}$	h) $44{,}771_{(10)}$	l) $6{,}175_{(10)}$	p) $0{,}455_{(10)}$

pretvoriti sa tri decimale tačnosti u odgovarajuće brojeve:

1) binarnog brojnog sistema, $x_{(10)} \rightarrow x_{(2)}$.

2) oktalnog brojnog sistema, $x_{(10)} \rightarrow x_{(8)}$.

3) heksadecimalnog brojnog sistema, $x_{(10)} \rightarrow x_{(16)}$.

Primer 30. Sledeće brojeve konvertovati iz oktalnog u heksadecimalni brojni sistem, $x_{(8)} \rightarrow x_{(16)}$:

a) $2345,56_{(8)}$	g) $3456,566_{(8)}$	m) $135,447_{(8)}$	t) $3112,3_{(8)}$
b) $12,3333_{(8)}$	h) $0,34567_{(8)}$	n) $50,505_{(8)}$	u) $0,6376_{(8)}$
c) $333,444_{(8)}$	i) $760,054_{(8)}$	o) $707,706_{(8)}$	v) $51765_{(8)}$
d) $47,156_{(8)}$	j) $1643,22_{(8)}$	p) $0,125_{(8)}$	x) $40,55_{(8)}$
e) $6,233_{(8)}$	k) $4672,502_{(8)}$	r) $5150_{(8)}$	y) $12,16433_{(8)}$
f) $33101_{(8)}$	l) $756,1_{(8)}$	s) $324,77_{(8)}$	z) $4774,16_{(8)}$

Primer 31. Sledeće brojeve konvertovati iz heksadecimalnog u oktalni brojni sistem $x_{(16)} \rightarrow x_{(8)}$:

a) $A6B,5C4_{(16)}$	g) $B,4CDE_{(16)}$	m) $9888,65_{(16)}$	t) $1230,ABC_{(16)}$
b) $F3ED3_{(16)}$	h) $2ABC,D_{(16)}$	n) $34A34B_{(16)}$	u) $0,5DF_{(16)}$
c) $99,ABCD_{(16)}$	i) $0,FEBC_{(16)}$	o) $ABC,DEF_{(16)}$	v) $ED34,57_{(16)}$
d) $2,ABB_{(16)}$	j) $891,435_{(16)}$	p) $3F2E,BAD_{(16)}$	x) $5A6B,DF_{(16)}$
e) $CC5_{(16)}$	k) $672,21C_{(16)}$	r) $578,226_{(16)}$	y) $1AB,AB3_{(16)}$
f) $CFA_{(16)}$	l) $70448_{(16)}$	s) $9177CF_{(16)}$	z) $16276,1_{(16)}$

---

2. POJAM KOMPLEMENTA, BINARNI BROJNI SISTEM I BINARNI BROJEVI SA ZNAKOM

Komplement je dopuna datog broja do neke unapred definisane vrednosti. Koristi se:

- Za prikazivanje negativnih označenih brojeva.
- Za realizaciju oduzimanja pomoću sabiranja.

U **binarnom brojnom sistemu** (S = 2) mogu da se definišu **samo dva** komplementa:

- **Komplement jedinice** (prvi komplement), koji se dobija invertovanjem svakog bita polaznog binarnog broja ($\mathbf{0} \rightarrow \mathbf{1}$ i $\mathbf{1} \rightarrow \mathbf{0}$).
- **Komplement dvojke** (drugi komplement), koji se dobija dodavanjem jedinice na prvi komplement.

Primer 1. Izraziti brojeve $0{,}4567_{(10)}$ i $34{,}639_{(10)}$ u komplementu do 10.

Rešenje:

$10^0 - 0{,}4567_{(10)} = 1 - 0{,}4567_{(10)} = \mathbf{0{,}5433_{(10)}}$

$10^2 - 34{,}639_{(10)} = 100 - 34{,}639_{(10)} = \mathbf{65{,}361_{(10)}}$

Primer 2. Izraziti brojeve iz zadatka 1. u komplementu do 9.

Rešenje:

$$\begin{aligned}10^0 - 10^{-4} - 0{,}4567_{(10)} &= (1 - 0{,}0001) - 0{,}4567_{(10)} \\ &= 0{,}9999_{(10)} - 0{,}4567_{(10)} \\ &= \mathbf{0{,}5432_{(10)}}\end{aligned}$$

$$\begin{aligned}10^2 - 10^{-3} - 34{,}639_{(10)} &= (100 - 0{,}001) - 34{,}639_{(10)} \\ &= 99{,}999_{(10)} - 34{,}639_{(10)} \\ &= \mathbf{65{,}360_{(10)}}\end{aligned}$$

Primer 3. Odrediti komplement devetke brojeva:

a) $38{,}25_{(10)}$
b) $876{,}345_{(10)}$
c) $4285{,}12_{(10)}$.

Rešenje:

a) $10^2 - 10^{-2} - 38{,}25_{(10)} = (100 - 0{,}01) - 38{,}25_{(10)} = 99{,}99_{(10)} - 38{,}25_{(10)} = \mathbf{61{,}74_{(10)}}$

b) $10^3 - 10^{-3} - 876{,}345_{(10)} = (1000 - 0{,}001) - 876{,}345_{(10)} = 999{,}999_{(10)} - 876{,}345_{(10)} = \mathbf{123{,}654_{(10)}}$

c) $10^4 - 10^{-2} - 4285{,}12_{(10)} = (10000 - 0{,}01) - 4285{,}12_{(10)} = 9999{,}99_{(10)} - 4285{,}12_{(10)} = \mathbf{5714{,}87_{(10)}}$

Primer 4. Odrediti prvi i drugi komplement sledećih binarnih brojeva $x_{(2)}$ bez znaka:

a) $101_{(2)}$ b) $11001_{(2)}$ c) $1011011_{(2)}$ d) $11001011_{(2)}$ e) $11001010_{(2)}$.

Rešenje:

a) Prvi komplement binarnog broja $x'_{(2)}$ dobija se invertovanjem svakog bita polaznog binarnog broja:

$$x'_{(2)} = \mathbf{010_{(2)}}$$

Drugi komplement binarnog broja $x''_{(2)}$ dobija se dodavanjem jedinice na prvi komplement:

$$x''_{(2)} = x'_{(2)} + 1 = \mathbf{011_{(2)}}$$

b) $x'_{(2)} = \mathbf{00110_{(2)}}$

$x''_{(2)} = x'_{(2)} + 1 = \mathbf{00111_{(2)}}$

c) $x'_{(2)} = \mathbf{0100100_{(2)}}$

$x''_{(2)} = x'_{(2)} + 1 = \mathbf{0100101_{(2)}}$

d) $x'_{(2)} = \mathbf{00110100_{(2)}}$

$x''_{(2)} = x'_{(2)} + 1 = \mathbf{00110101_{(2)}}$

e) $x'_{(2)} = \mathbf{00110101_{(2)}}$

$x''_{(2)} = x'_{(2)} + 1 = \mathbf{00110110_{(2)}}$.

Primer 5. Odrediti prvi i drugi komplement sledećih binarnih brojeva bez znaka:

a) $01110111_{(2)}$ c) $10101100_{(2)}$

b) $11111000_{(2)}$ d) $01010101_{(2)}$

Rešenje:

a) Prvi komplement binarnog broja $x'_{(2)}$ dobija se invertovanjem svakog bita polaznog binarnog broja:

$$x'_{(2)} = 10001000_{(2)}$$

Drugi komplement binarnog broja $x''_{(2)}$ dobija se dodavanjem jedinice na prvi komplement:

$$x''_{(2)} = x'_{(2)} + 1 = 10001001_{(2)}$$

b), c) i d) se rade na isti način kao primer a).

Pravilo za konverziju 8-bitnih binarnih vrednosti u decimalan broj

- **Neoznačena** (unsigned) 8-bitna binarna vrednost konvertuje se u ekvivalentnu decimalnu vrednost primenom formule:

$x_{(10)} = b_7 \cdot 2^7 + b_6 \cdot 2^6 + ... + b_1 \cdot 2^1 + b_0 \cdot 2^0$ $0_{(10)} \le x_{(10)} \le 255_{(10)}$

- **Označena** (signed) 8-bitna binarna vrednost konvertuje se u ekvivalentnu decimalnu vrednost, pod pretpostavkom korišćenja tehnike **drugog komplementa** za zapis označenih brojeva, primenom formule:

$x_{(10)} = - b_7 \cdot 2^7 + b_6 \cdot 2^6 + ... + b_1 \cdot 2^1 + b_0 \cdot 2^0$ $-127_{(10)} \le x_{(10)} \le +128_{(10)}$

Primer 6. Izvršiti konverziju 8-bitnog binarnog broja $x_{(2)} = 11001011_{(2)}$ u decimalni brojni sistem, $x_{(2)} \rightarrow x_{(10)}$, ukoliko je zapis binarnog broja:

a) bez znaka,
b) sa znakom i predstavljen u prvom komplementu,
c) sa znakom i predstavljen u drugom komplementu.

Rešenje:

a) Pošto je format zapisa celobrojan i bez znaka, u opštem slučaju binarni broj može da se prikaže kao:

$x = b_R\, b_{R-1} \ldots b_1\, b_0$, gde su b_R , b_{R-1} , ..., b_1 , $b_0 \in \{0, 1\}$ cifre u binarnom zapisu

Za konverziju u decimalni brojni sistem $x_{(2)} \rightarrow x_{(10)}$, koristi se opšta formula

$$x_{(10)} = b_R{\cdot}2^R + b_{R-1}{\cdot}2^{R-1} + \ldots + b_1{\cdot}2^1 + b_0{\cdot}2^0$$

Primenom formule za konverziju $x_{(2)} \rightarrow x_{(10)}$ na zadati binarni broj $x_{(2)} = 01001011_{(2)}$, dobija se:

$$x_{(10)} = 1{\cdot}2^7 + 1{\cdot}2^6 + 0{\cdot}2^5 + 0{\cdot}2^4 + 1{\cdot}2^3 + 0{\cdot}2^2 + 1{\cdot}2^1 + 1{\cdot}2^0 = \mathbf{203_{(10)}}$$

b) Znak binarnog broja se nalazi na poziciji bita najveće težine (MSB). Pošto je u datom zapisu $x_{(2)} = 11001011_{(2)}$, na mestu bita najveće težine “1”, to znači da je broj negativan. U postupku konverzije u decimalni brojni sistem, treba prvo odrediti osnovnu vrednost binarnog broja, s tim što se ne konvertuje bit najveće težine:

$$x'_{(2)} = 10110100_{(2)}$$

Primenom formule za konverziju $x_{(2)} \rightarrow x_{(10)}$ na binarni broj $x'_{(2)}$,dobija se:

$$x_{(10)} = -\,(0{\cdot}2^6 + 1{\cdot}2^5 + 1{\cdot}2^4 + 0{\cdot}2^3 + 1{\cdot}2^2 + 0{\cdot}2^1 + 0{\cdot}2^0) = \mathbf{-52_{(10)}}$$

c) Znak binarnog broja se nalazi na poziciji bita najveće težine (MSB). Pošto je u datom zapisu $x_{(2)} = 11001011_{(2)}$, na mestu bita najveće težine “1”, to znači da je broj negativan. U postupku konverzije u decimalan brojni sistem, treba prvo odrediti osnovnu vrednost binarnog broja, s tim što se ne konvertuje bit najveće težine. Na osnovu rešenja iz tačke b) sledi:

$$x''_{(2)} = x'_{(2)} + 1 = 10110101_{(2)}$$

Primenom formule za konverziju $x_{(2)} \rightarrow x_{(10)}$ na binarni broj $x''_{(2)}$,dobija se:

$$x_{(10)} = -\,(0{\cdot}2^6 + 1{\cdot}2^5 + 1{\cdot}2^4 + 0{\cdot}2^3 + 1{\cdot}2^2 + 0{\cdot}2^1 + 1{\cdot}2^0) = \mathbf{-53_{(10)}}$$

Primer 7. Ako se u memoriji osmobitnog računara nalazi podatak $10000001_{(2)}$, koji decimalni broj je predstavljen ovim podatkom ako je:

a) podatak neoznačen
b) podatak označen.

Rešenje:

a) Za neoznačeni podatak:

$$1{\cdot}2^7 + 0{\cdot}2^6 + 0{\cdot}2^5 + 0{\cdot}2^4 + 0{\cdot}2^3 + 0{\cdot}2^2 + 0{\cdot}2^1 + 1{\cdot}2^0 = 128 + 1 = \mathbf{129_{(10)}}$$

b) Za označeni podatak:

$$-1{\cdot}2^7 + 0{\cdot}2^6 + 0{\cdot}2^5 + 0{\cdot}2^4 + 0{\cdot}2^3 + 0{\cdot}2^2 + 0{\cdot}2^1 + 1{\cdot}2^0 = -128 + 1 = \mathbf{127_{(10)}}$$

Primer 8. Koji označeni broj je predstavljen 8-bitnim binarnim zapisom $10101001_{(2)}$?

Rešenje:

I način:

$$-1\cdot2^7 + 0\cdot2^6 + 1\cdot2^5 + 0\cdot2^4 + 1\cdot2^3 + 0\cdot2^2 + 0\cdot2^1 + 1\cdot2^0 = -128 + 32 + 8 + 1 = \mathbf{-87_{(10)}}$$

II način:

Kako je broj negativan, izračuna se drugi komplement broja.

$$\begin{array}{lll} x_{(2)} & \rightarrow & 10101001_{(2)} \\ x'_{(2)} & \rightarrow & 01010110_{(2)} \\ & & \underline{\qquad\quad +1} \\ x''_{(2)} & \rightarrow & 01010111_{(2)} \end{array}$$

$x''_{(2)} \rightarrow 01010111_{(2)} \rightarrow 1\cdot2^6+1\cdot2^4+1\cdot2^2+1\cdot2^1+1\cdot2^0 = \mathbf{87}$, tako da je traženi **označeni broj $-87_{(10)}$.**

Primer 9. U memoriji se nalaze brojevi:

a) $14B0_{(16)}$
b) $8011_{(16)}$
c) $83_{(16)}$

O kojim decimalnim brojevima se radi ako su zapisani kao šesnaestobitni označeni, a o kojim ako su zapisani kao šesnaestobitni neoznačeni brojevi?

Rešenje:

a) $14B0_{(16)} = 0\ 0\ 0\ 1\ 0\ 1\ 0\ 0\ 1\ 0\ 1\ 1\ 0\ 0\ 0\ 0_{(2)}$

$\mathbf{2^{12}}\quad \mathbf{2^{10}}\quad \mathbf{2^{7}}\quad \mathbf{2^{5}}\ \mathbf{2^{4}}$

Ispod binarnog koda, prikazane su pozicione vrednosti (težinski koeficijenti) za svaku poziciju na kojoj se nalazi jedinica. Decimalni broj se dobija kao rezultat sabiranja ovih pozicionih vrednosti:

$$14B0_{(16)} = 0001\ 0100\ 1011\ 0000_{(2)} = 2^{12} + 2^{10} + 2^7 + 2^5 + 2^4 = \mathbf{5296_{(10)}}$$

Rezultat je isti i ako se broj tretira kao označen i neoznačen. Označeni brojevi se razlikuju od neoznačenih jedino po koeficijentu uz bit najveće pozicione vrednosti koji je u slučaju označenih brojeva negativan (-2^{15}) a slučaju neoznačenih pozitivan ($+2^{15}$), svi ostali težinski koeficijenti su jednaki. Kako je bit najveće pozicione vrednosti nula, to ovaj zapis odgovara broju 5296 i kao označen i kao neoznačen broj.

b) $8011_{(16)} = 1\,0\,0\,0\,0\,0\,0\,0\ \ 0\,0\,0\,1\,0\,0\,0\,1_{(2)}$

$\mathbf{2^{15}}\qquad \mathbf{2^{4}}\quad \mathbf{2^{0}}$ → Težinski koeficijenti za **neoznačene** brojeve

$\mathbf{-2^{15}}\qquad \mathbf{2^{4}}\quad \mathbf{2^{0}}$ → Težinski koeficijenti za **označene** brojeve

Ispod binarnog koda, prikazane su pozicione vrednosti (težinski koeficijenti) za svaku poziciju na kojoj se nalazi jedinica. U prvom redu su koeficijenti za neoznačene, a u drugom za označene brojeve. Decimalni broj se dobija kao rezultat sabiranja ovih pozicionih vrednosti:

$8011_{(16)} = 1000\ 0000\ 0001\ 0001_{(2)} = 2^{15} + 2^4 + 2^0 = \mathbf{32785_{(10)}}$ kao **neoznačeni** broj

$8011_{(16)} = 1000\ 0000\ 0001\ 0001_{(2)} = -2^{15} + 2^4 + 2^0 = -\mathbf{32751_{(10)}}$ kao **označeni** broj

Da se podsetimo:

Označeni brojevi se razlikuju od neoznačenih jedino po težinskom koeficijentu bita najveće pozicione vrednosti koji je u slučaju označenih brojeva negativan (-2^{15}), a u slučaju neoznačenih, pozitivan ($+2^{15}$). Svi ostali težinski koeficijenti su jednaki

c) $83_{(16)} = 0\,0\,0\,0\,0\,0\,0\,0\;\; \underset{\mathbf{2^7}}{1}\,0\,0\,0\,0\,0\,\underset{\mathbf{2^1}}{1}\,\underset{\mathbf{2^0}}{1}_{(2)}$

Ispod binarnog kôda, prikazane su pozicione vrednosti (težinski koeficijenti) za svaku poziciju na kojoj se nalazi jedinica. Decimalni broj se dobija kao rezultat sabiranja ovih pozicionih vrednosti:

$$83_{(16)} = 0000\ 0000\ 1000\ 0011_{(2)} = 2^7 + 2^1 + 2^0 = \mathbf{131_{(10)}}$$

Rezultat je isti i ako se broj tretira kao označen i neoznačen, jer je bit najveće pozicione vrednosti (za šesnaestobitne zapise to je bit broj 15) nula.

Primer 10. U šesnaestobitnom zapisu predstaviti decimalni broj $-41_{(10)}$. Odgovor dati kao šesnaestobitni (četvorocifreni) heksadecimalni broj.

Rešenje:

Broj $-41_{(10)}$ očigledno treba predstaviti kao označen broj, jer neoznačeni brojevi mogu biti samo pozitivni.

Da se podsetimo:

- *Označeni brojevi mogu biti i pozitivni i negativni.*
- *Kod označenih brojeva bitna je dužina zapisa, jer je samo težinski koeficijent najznačajnije binarne cifre negativan pa nije svejedno koliko taj, jedini negativan težinski koeficijent iznosi. Iz tog razloga, za predstavljanje ovog broja treba koristiti isključivo šesnaestobitni zapis.*
- *Promena znaka označenog broja je računanje drugog komplementa.*
- *U osmobitnom zapisu označenog broja mogu se predstaviti brojevi od -128 do +127 i kad god u bilo kom obliku uključujemo negativne brojeve, bilo da je u pitanju promena znaka ili aritmetičke operacije sa drugim brojevima koji mogu biti negativni, moramo koristiti zapis označenih brojeva.*

! Obratite pažnju: broj +129 se ne može predstaviti kao osmobitni označen broj iako prilikom pretvaranja u binarni kôd dobijamo osam binarnih cifara. Vrlo je važno da pokušate sami da odgovorite i istinski razumete zašto je to tako.

Krenimo od apsolutne vrednosti ovog broja (dakle, od broja +41). Pretvaranjem u binarni zapis dobija se $41_{(10)} = 101001_{(2)}$ (pretvaranje se može obaviti, na primer uzastopnim deljenjem sa 2 kao u ranije datim primerima). Međutim, zbog teksta zadatka, nas zanima isključivo šesnaestobitni zapis ovog broja. Proširenje dužine zapisa **pozitivnog broja** se obavlja jednostavnim dopisivanjem potrebnog broja nula nalevo. *Da ponovimo, ovo važi samo za* ***POZITIVNE BROJEVE.*** Zato je $+41_{(10)}$, kao šesnaestobitni označeni broj:

$$+41_{(10)} = 0000\ 0000\ 0010\ 1001_{(2)}$$

Da bi dobili zapis broja $-41_{(10)}$ treba samo izračunati drugi komplement binarnog zapisa broja $+41_{(10)}$

> *Da se podsetimo:*
> *Operacija drugog komplementa menja znak označenog broja. Operacija drugog komplementa nad neoznačenim brojevima nema aritmetičkog smisla.*

Prvi komplement je $1111\ 1111\ 1101\ 0110_{(2)}$ (invertovanjem, to jest promenom svakog bita pojedinačno sa nule na jedan i obrnuto), pa je drugi komplement (dodavanjem jedinice na prvi komplement):

$$\begin{array}{rr} & 1111\ 1111\ 1101\ 0110_{(2)} \\ + & \underline{0000\ 0000\ 0000\ 0001}_{(2)} \\ \mathbf{-41_{(10)}} = & \mathbf{1111\ 1111\ 1101\ 0111_{(2)}} \end{array}$$

Zapisano kao heksadecimalni četvorocifreni broj (direktno iz tablice heksadecimalnih cifara):

$$\mathbf{-41_{(10)} = 1111\ 1111\ 1101\ 0111_{(2)} = FFD7_{(16)}}$$

> **! Česta greška:** Pogrešno bi bilo poći od kôda broja +41 kakav se dobija direktno pretvaranjem u binarni sistem i nad binarnim zapisom te dužine uraditi drugi komplement. Direktno pretvoren u binarni kôd, $41 = 101001_{(2)}$ što je šestobitni binarni zapis ovog broja. Drugi komplement ovog binarnog zapisa je $010111_{(2)}$ što svakako nije binarni kôd broja - 41 (*uporedite sa rešenjem*). Bit najveće pozicione vrednosti je nula, pa to svakako ne može biti zapis broja koji je negativan. Problem je u tome što se mora poći od zapisa broja +41 kao <u>označenog broja</u> , a za to nije dovoljna dužina zapisa od šest bita. Ako pogledamo ovaj šestobitni zapis, primetićemo da je bit najveće pozicione vrednosti jedinica što bi u svetu označenih brojeva bilo tumačeno kao negativan broj. I zaista sa šest bita se mogu predstaviti označeni brojevi od -2^5 do $+2^5-1$ (-32 do +31), a +41 ne spada u taj opseg. Minimalna dužina zapisa broja +41 kao označenog je sedam bita, tako da sa zapisom dužine sedam ili više bita nema tih problema. Zapisi dužina različitih od 8, 16 ili 32 bita nemaju praktični značaj pa je vrlo korisno raditi samo sa zapisima date dužine. Pri tome se mora voditi računa o opsegu brojeva.

> ***** Za one koji vole "drugačije":*** *Koristeći činjenicu da su i označeni i neoznačeni brojevi u binarnom brojnom sistemu poređani "u krug" (knjiga, poglavlje 2.5.1) binarni kôd negativnih brojeva se može dobiti i na drugi način. Ukupan broj brojeva na krugu za šesnaestobitni brojni sistem je* 65536 *pa se kôd broja* -41 *(normalno, označenog) poklapa sa kôdom broja* 65536-41=65495 *kao neoznačenog broja (ponovo vidi primer na trobitnom sistemu u udžbeniku iz njega se može jasno izvesti ovaj zaključak). Klasičnim konvertovanjem u binarni sistem broja 65495 dobija se tačan kôd broja* -41.

<u>Primer 11.</u> U šesnaestobitnom zapisu predstaviti decimalni broj $-3_{(10)}$. Odgovor dati kao šesnaestobitni (četvorocifreni) heksadecimalni broj.

Rešenje:

Kao i u prethodnom primeru, kada se traži binarni zapis broja $-3_{(10)}$ što svakako mora biti **označeni** broj, moramo poći od šesnaestobitnog binarnog zapisa broja $+3_{(10)}$ kao **označenog** broja. Binarni kod broja $+3_{(10)}$ je $011_{(2)}$ ili, predstavljeno kao šesnaestobitni označeni broj:

$+3_{(10)} = 0000\ 0000\ 0000\ 0011_{(2)}$

Broj –3 se dobija kao drugi komplement ovog binarnog zapisa. *Da podsetimo, drugi komplement se dobija kao prvi komplement (svaki bit invertovan) i uvećan za jeda*

$-3_{(10)} = 1111\ 1111\ 1111\ 1101_{(2)} = \mathbf{FFFD}_{\mathbf{(16)}}$

! Česta greška: Kao često rešenje ovog i sličnih zadataka pojavljuje se odgovor:

$-3_{(10)} = 1000\ 0000\ 0000\ 0011_{(2)}$

Sa idejom da jedinica kao najznačajniji bit pretvara broj u negativan, što je potpuno pogrešno kada se za predstavljanje negativnih brojeva koristi tehnika drugog komplementa, a u savremenim računarima se jedino ona koristi u tu svrhu.

Da se podsetimo:
Jedinica na mestu bita najveće pozicione vrednosti (MSB) zaista znači da je broj negativan, ali ostali biti nisu apsolutna vrednost tog broja ***niti se prostim menjanjem MSB menja znak broju!***

Primer 12. U šesnaestobitnom zapisu predstaviti decimalni broj $1025_{(10)}$. Odgovor dati kao šesnaestobitni (četvorocifreni) heksadecimalni broj.

Rešenje:

$\mathbf{1025_{(10)}} = 1024+1 = 2^{10} + 2^{0} = 0000\ 0100\ 0000\ 0001_{(2)} = \mathbf{0401_{(16)}}$.

Zapis je isti bez obzira da li se broj tretira kao označen ili neoznačen.

Ako se znaju stepeni broja 2, potpuno je neracionalno trošiti vreme na uzastopno deljenje sa 2 (klasično pretvaranje).

Da se podsetimo:
U sistemu šesnaestobitnih brojeva razlike nastaju tek za brojeve veće od +32767 koji se mogu predstaviti kao neoznačeni ali kao označeni ne mogu. Normalno, negativni brojevi do –32768 se mogu predstaviti jedino kao označeni. Brojevi –32769 i manji (negativniji) se ne mogu nikako predstaviti pomoću šesnaestobitnog zapisa.

Primer 13. U šesnaestobitnom zapisu predstaviti decimalni broj $-512_{(10)}$. Odgovor dati kao šesnaestobitni (četvorocifreni) heksadecimalni broj.

Rešenje:

Broj se mora tretirati kao označen. Polazimo od šesnaestobitnog zapisa $+512_{(10)}$ kao označenog broja (u ovom slučaju, isti je i zapis označenog i neoznačenog broja):

$+512_{(10)} = 2^{9} = 0000\ 0010\ 0000\ 0000_{(2)}$

Prvi komplement ovog zapisa je $1111\ 1101\ 1111\ 1111_{(2)}$ pa se konačno rešenje dobija dodavanjem jedinice na ovaj zapis:

$\mathbf{-512_{(10)}} = 1111\ 1110\ 0000\ 0000_{(2)} = \mathbf{FE00_{(16)}}$

Primer 14. U šesnaestobitnom zapisu predstavi sledeće decimalne brojeve:

a) $-6_{(10)}$
b) $-33_{(10)}$
c) $2049_{(10)}$

Rešenje:

a) $6_{(10)} \rightarrow 0000000000000110_{(2)}$

$$
\begin{array}{rl}
1111111111111001_{(2)} & \text{(1. komplement)} \\
\underline{\qquad\qquad\qquad\qquad +1} & \\
\mathbf{1111111111111010_{(2)}} & \text{(2. komplement)}
\end{array}
$$

b) $33_{(10)} \rightarrow 0000000000100001_{(2)}$

$$
\begin{array}{rl}
1111111111011110_{(2)} & \text{(1. komplement)} \\
\underline{\qquad\qquad\qquad\qquad +1} & \\
\mathbf{1111111111011111_{(2)}} & \text{(2. komplement)}
\end{array}
$$

c) $2049_{(10)} \rightarrow \mathbf{0000100000000001_{(2)}}$

Pravila binarnog sabiranja

0 + 0 = 0 **1 + 0 = 1**
0 + 1 = 1 **1 + 1 = 10** 1 = prenos (carry)

Primer 15. Binarno sabrati brojeve $10100011_{(2)}$ i $00111010_{(2)}$ ako su ulazni podaci:

a) Dva neoznačena binarna broja.
b) Dva označena binarna broja.
Oba rezultata predstaviti i kao decimalni broj.

Rešenje:

a)
$$
\begin{array}{r}
10100011_{(2)} \\
+\underline{00111010_{(2)}} \\
\mathbf{11011101_{(2)}}
\end{array}
$$

Zbir je $\mathbf{221_{(10)}}$

b) Postupak isti kao pod a). ***Uvek je isti*!**

Zbir je $\mathbf{-35_{(10)}}$

Primer 16. Binarno sabrati brojeve $01101001_{(2)}$ i $10001010_{(2)}$ ako su ulazni podaci:

a) Dva neoznačena binarna broja.
b) Dva označena binarna broja.
Oba rezultata predstaviti i kao decimalni broj.

Rešenje:

a)
$$
\begin{array}{rl}
01101001_{(2)} & \\
+\ \underline{10001010_{(2)}} & \\
\mathbf{11110011_{(2)}} & \text{zbir je } \mathbf{243_{(10)}}
\end{array}
$$

b) Postupak isti kao pod a). *Uvek je isti*. Zbir je $\mathbf{-13_{(10)}}$.

Posle svake aritmetičke operacije u ALU, procesor postavlja ili briše kontrolne bite u registru stanja (**zastavice**, ***flag***-ovi) čija vrednost može da bude 1 ili 0.

- C (Carry) = 1 označava da postoji prenos iz bita najveće težine.
- **N** (Negative) = 1 označava negativan rezultat kada su podaci označeni brojevi.
- **V** (oVerflow) = 1 signalizira da je rezultat označeni broj van opsega (–128 do +127).
- **Z** (Zero) = 1 signalizira da je rezultat aritmetičke operacije 0.

<u>Primer 17.</u> U osmobitnoj aritmetici binarno sabrati brojeve $129_{(10)}$ i $131_{(10)}$. Koji decimalni broj je rezultat sabiranja? Komentarisati odgovor. Kakvo će biti stanje zastavica V, N, C posle sabiranja?

Rešenje:

$10000001_{(2)}$
$+\ \underline{10000011}_{(2)}$
~~1~~**00000100**$_{(2)}$ zbir je **4**$_{(10)}$

Rezultal sabiranja u mikroračunaru sa 8-bitnom ALU je 8- bitni broj. Deveti bit najveće pozicione vrednosti (predstavljen kao bleđi) biva odsečen i pojavljuje se kao jedinica u zastavici prenosa C.

U ovom slučaju, stanje flegova će biti sledeće: **C** = **1**, **N** = **0** (bit najveće težine u rezultatu je **0**, pa je broj tretiran kao pozitivan), **V** = **1** (fleg V se postavlja pod pretpostavkom da su ulazni podaci označeni brojevi. Prvi broj $10000001_{(2)}$, posmatran kao označen je $-127_{(10)}$, a drugi $10000011_{(2)}$ je $-125_{(10)}$. Sabiranjem ova dva broja dobija se broj van opsega označenog osmobitnog broja (od –128 do $127)_{(10)}$ pa se V fleg postavlja na 1).

Zbog pojave prekoračenja rezultat nije aritmetički tačan.

<u>Primer 18.</u> U osmobitnoj aritmetici binarno sabrati brojeve $126)_{(10)}$ i $-44)_{(10)}$. Koji decimalni broj je rezultat sabiranja? Komentarisati odgovor. Kakvo će biti stanje zastavica V, N, C posle sabiranja?

Rešenje:

$01111110_{(2)}$
$+\ \underline{11010100}_{(2)}$
~~1~~**01010010**$_{(2)}$ zbir je **82**$_{(10)}$

Rezultal sabiranja u mikroračunaru sa 8-bitnom ALU je 8- bitni broj. Deveti bit najveće pozicione vrednosti (predstavljen kao bleđi) biva odsečen i pojavljuje se kao jedinica u zastavici prenosa C.

U ovom slučaju, stanje flegova će biti sledeće: **C = 1**, **N = 0** (bit najveće težine rezultata je nula, pa je broj tretiran kao pozitivan), **V = 0** (fleg V se postavlja pod pretpostavkom da su ulazni podaci označeni brojevi. Prvi broj je $124_{(10)}$ i kada se tretira kao označen i kao neoznačen, a drugi je $-44_{(10)}$ kada se tretira kao označen. Sabiranjem ova dva broja dobija se broj u opsegu označenog osmobitnog broja (od –128 do $127)_{(10)}$ pa se V fleg briše).

Prilikom sabiranja postoji prenos, međutim, to što je došlo do prenosa nema uticaja na aritmetičku tačnost rezultata. Rezultat je aritmetički tačan, jer nema prekoračenja (V=0).

<u>Primer 19.</u> Ilustrovati kako bi se u binarnom sistemu sabrali brojevi $143_{(10)}$ i $114_{(10)}$:

a) U mikroračunaru sa 8-bitnom ALU.

b) U mikroračunaru sa 16-bitnom ALU.

Koji decimalni broj je rezultat sabiranja pod a) i b) ako se ulazni podaci tretiraju kao označeni brojevi?

Koji decimalni broj je rezultat sabiranja pod a) i b) ako se ulazni podaci tretiraju kao neoznačeni brojevi?

Kakvo će biti stanje zastavica V, N i C posle sabiranja pod a) i b)?

Rešenje:

a) $143_{(10)} = 10001111_{(2)}$ 8-bitni rezultat
$+\ 114_{(10)} = \underline{01110010}_{(2)}$
$\cancel{1}\ \mathbf{00000001_{(2)}}$

Rezultal sabiranja u mikroračunaru sa 8-bitnom ALU je 1. Deveti bit najveće pozicione vrednosti (predstavljen kao bleđi) biva odsečen i pojavljuje se kao jedinica u zastavici prenosa C.

Rezultat sabiranja je isti nezavisno da li se brojevi posmatraju kao označeni ili neoznačeni. ALU sabira binarne brojeve, a to da li je broj označen ili ne stvar je tumačenja programera, postupak sabiranja u ALU uopšte ne zavisi od tumačenja programera. Ako se brojevi tretiraju kao označeni, prvi sabirak nije $143_{(10)}$ već $-113_{(10)}$ (binarni kod $10001111_{(2)}$ je $143_{(10)}$ ako se tretira kao neoznačeni a $-113_{(10)}$ kada se tretira kao označeni). Za označene brojeve ovo sabiranje daje aritmetički tačan rezultat, bez obzira što postoji prenos u deveti bit koji se gubi (biva odsečen).

Da se podsetimo:
Prenos u deveti bit ne mora da znači prekoračenje kada se brojevi tretiraju kao označeni, zastavica V ukazuje na prekoračenje (i da rezultat nije aritmetički tačan). Kada se brojevi tretiraju kao neoznačeni, C zastavica nosi informaciju da rezultat nije tačan, a ne fleg V.

Zastavice su sledeće: **C=1**; **N=0** (stanje bita najveće pozicione vrednosti u rezultatu) i V=0 jer kada se podaci tretiraju kao označeni nema prekoračenja: sabiraju se brojevi $114_{(10)}$ i $-113_{(10)}$ i dobija se rezultat 1 koji je u opsegu označenih osmobitnih brojeva (od –128 do $+127)_{(10)}$.

b) $143 = 0000\ 0000\ 1000\ 1111_{(2)}$
$+\ 114 = \underline{0000\ 0000\ 0111\ 0010}_{(2)}$
$\mathbf{0000\ 0001\ 0000\ 0001_{(2)}}$

Rezultat sabiranja u mikroračunaru sa 16-bitnom ALU je $\mathbf{257_{(10)}}$.

Rezultat sabiranja je isti nezavisno da li se brojevi posmatraju kao označeni ili neoznačeni. ALU sabira binarne brojeve, a to da li je broj označen ili ne stvar je tumačenja programera, postupak sabiranja u ALU uopšte ne zavisi od tumačenja programera. Oba sabirka su isti i ako se tretiraju i kao označeni i kao neoznačeni.

Sve zastavice **C**, **V** i **N** imaju vrednost **nula**. Nema prekoračenja, nema prenosa u sedamnaesti bit i najznačajniji bit je nula pa je broj pozitivan.

Primer 20. Ilustrovati kako bi osmobitni računar binarno sabrao brojeve $66_{(10)}$ i $254_{(10)}$ ako su u pitanju:

a) Dva neoznačena broja
b) Dva označena broja

Koji decimalni broj je rezultat sabiranja pod a) i b)?

U kom će stanju biti zastavice C, V i N posle sabiranja pod a) i b)?

Rešenje:

a) $\quad 01000010_{(2)} = 66_{(10)}$

$\quad + \underline{11111110}_{(2)} = 254_{(10)}$

$\quad \cancel{1}\ \mathbf{01000000_{(2)}}$ zbir je $\mathbf{64_{(10)}}$

b) **Postupak isti kao pod a).** Rezultat je $\mathbf{64}_{(10)}$ (deveti, najznačajniji bit je odsečen) i pod a) i pod

Rezultat je tačan za označene brojeve, jer je $11111110_{(2)} = -2_{(10)}$, kao označen broj

C = 1, V = 0, N = 0 i pod a) i pod b).

Primer 21. Ilustrovati kako bi osmobitni računar binarno sabrao brojeve $34_{(10)}$ i $253_{(10)}$ kao:

a) Dva neoznačena binarna broja
b) Dva označena binarna broja

Koji decimalni broj je rezultat sabiranja pod a) i b)

U kom će stanju biti zastavice C, V i N posle sabiranja pod a) i b)

Rešenje:

a) $\quad 00100010_{(2)} = 34_{(10)}$

$\quad + \underline{11111101}_{(2)} = 253_{(10)}$

$\quad \cancel{1}\ \mathbf{00011111_{(2)}}$ zbir je $\mathbf{31_{(10)}}$

b) **Postupak isti kao pod a).** Rezultat je $\mathbf{31}_{(10)}$ (deveti bit je odsečen) i pod a) i pod b)

Rezultat je tačan za označene jer je $11111101_{(2)} = -3_{(10)}$ kao označen broj

C = 1, V = 0, N = 0 i pod a) i pod b).

Oduzimanje može da se svede na sabiranje:

$X - Y = X + (-Y)$

$(-Y)$ je negativna vrednost kodovana primenom prvog ili drugog komplementa.

Pravilo za oduzimanje primenom sabiranja i prvog komplementa

- Negativan broj se predstavlja kao prvi komplement binarnog zapisa.
- Kada se na MSB mestu pojavi prenos (C, carry) rezultat se koriguje dodavanjem prenosa na LSB.
- Razlikuju se pozitivna i negativna nula:
  $+0_{(10)} = 00000000_{(2)}$ i $-0_{(10)} = 11111111_{(2)}$

Pravilo za oduzimanje primenom drugog komplementa

- Negativan broj se predstavlja kao drugi komplement binarnog zapisa.
- Praktično jedini način predstavljanja negativnog broja u računarskom sistemu.
- Kada se na MSB mestu pojavi prenos ignoriše se (odseca se), ali ostaje zapisan u C flegu (carry flag).
- U drugom komplementu pozitivna i negativna nula su iste: $\pm 0_{(10)} = 00000000_{(2)}$.

Primer 22. Koristeći **8–bitne** binarne brojeve **sa znakom** izračunati primenom **prvog komplementa** razliku brojeva: $100_{(10)} - 56_{(10)} = ?$, a zatim prikazati rezultat u decimalnom brojnom sistemu.

Rešenje:

100 : 2 = 50	0
50 : 2 = 25	0
25 : 2 = 12	1
12 : 2 = 6	0
6 : 2 = 3	0
3 : 2 = 1	1
1 : 2 = 0	1

$100_{(10)} \rightarrow 01100100_{(2)}$

56 : 2 = 28	0
28 : 2 = 14	0
14 : 2 = 7	0
7 : 2 = 3	1
3 : 2 = 1	1
1 : 2 = 0	1

$56_{(10)} \rightarrow 00111000_{(2)}$

$-56_{(10)} \rightarrow 11000111_{(2)}$ 1.komplement

$100_{(10)} - 56_{(10)} = 100_{(10)} + (- 56_{(10)})$

$$
\begin{array}{r}
01100100_{(2)} \\
+\ \underline{11000111}_{(2)} \\
1\ 00101011_{(2)} \\
\underline{+1} \\
00101100_{(2)} = 2^5 + 2^3 + 2^2 = \mathbf{44_{(10)}}.
\end{array}
$$

Primer 23. Koristeći **8–bitne** binarne brojeve **sa znakom** izračunati primenom **prvog komplementa** razliku brojeva: $111_{(10)} - 87_{(10)} = ?$, a zatim prikazati rezultat u decimalnom brojnom sistemu.

Rešenje:

111 : 2 = 55	1
55 : 2 = 27	1
27 : 2 = 13	1
13 : 2 = 6	1
6 : 2 = 3	0
3 : 2 = 1	1
1 : 2 = 0	1

87 : 2 = 43	1
43 : 2 = 21	1
21 : 2 = 10	1
10 : 2 = 5	0
5 : 2 = 2	1
2: 2 = 1	0
1: 2 = 0	1

$111_{(10)} \rightarrow 01101111_{(2)}$

$87_{(10)} \rightarrow 01010111_{(2)}$

$-87_{(10)} \rightarrow 10101000_{(2)}$ 1.komplement

$111_{(10)} - 87_{(10)} = 111_{(10)} + (-87_{(10)})$

$$
\begin{array}{r}
01101111_{(2)} \\
+\underline{10101000}_{(2)} \\
100010111_{(2)} \\
\underline{+1} \\
00011000_{(2)} \rightarrow 2^4 + 2^3 = \mathbf{24_{(10)}}.
\end{array}
$$

Primer 24. Koristeći **8–bitne** binarne brojeve **sa znakom** izračunati primenom **prvog komplementa** razliku brojeva: $113_{(10)} - 66_{(10)} = ?$, a zatim prikazati rezultat u decimalnom brojnom sistemu.

Rešenje:

$$\begin{array}{r|l}
113 : 2 = 56 & 1 \\
56 : 2 = 28 & 0 \\
28 : 2 = 14 & 0 \\
14 : 2 = 7 & 0 \\
7 : 2 = 3 & 1 \\
3 : 2 = 1 & 1 \\
1 : 2 = 0 & 1
\end{array}\uparrow
\qquad\qquad
\begin{array}{r|l}
66 : 2 = 33 & 0 \\
33 : 2 = 16 & 1 \\
16 : 2 = 8 & 0 \\
8 : 2 = 4 & 0 \\
4 : 2 = 2 & 0 \\
2 : 2 = 1 & 0 \\
1 : 2 = 0 & 1
\end{array}\uparrow$$

$113_{(10)} \rightarrow 01110001_{(2)}$

$66_{(10)} \rightarrow 01000010_{(2)}$

$-66_{(10)} \rightarrow 10111101_{(2)}$ 1.komplement

$113_{(10)} - 66_{(10)} = 113_{(10)} + (-66_{(10)})$

$$\begin{array}{r}
01110001_{(2)} \\
+\ \underline{10111101}_{(2)} \\
1\ 00101110_{(2)} \\
\rightarrow\ +1 \\
\hline
00101111_{(2)}
\end{array}
\rightarrow 2^5 + 2^3 + 2^2 + 2^1 + 2^0 = \mathbf{47_{(10)}}.$$

Primer 25. Koristeći **8 – bitne** binarne brojeve **sa znakom** izračunati primenom **prvog komplementa** razliku brojeva: $91_{(10)} - 124_{(10)} = ?$, a zatim prikazati rezultat u decimalnom brojnom sistemu.

Rešenje:

$$\begin{array}{r|l}
124 : 2 = 62 & 0 \\
62 : 2 = 31 & 0 \\
31 : 2 = 15 & 1 \\
15 : 2 = 7 & 1 \\
7 : 2 = 3 & 1 \\
3 : 2 = 1 & 1 \\
1 : 2 = 0 & 1
\end{array}\uparrow
\qquad\qquad
\begin{array}{r|l}
91 : 2 = 45 & 1 \\
45 : 2 = 22 & 1 \\
22 : 2 = 11 & 0 \\
11 : 2 = 5 & 1 \\
5 : 2 = 2 & 1 \\
2 : 2 = 1 & 0 \\
1 : 2 = 0 & 1
\end{array}\uparrow$$

$124_{(10)} \rightarrow 01111100_{(2)}$

$-124_{(10)} \rightarrow 10000011_{(2)}$ 1.komplement

$91_{(10)} \rightarrow 01011011_{(2)}$

$91_{(10)} - 124_{(10)} = 91_{(10)} + (-124_{(10)})$

$$\begin{array}{r}
01011011_{(2)} \\
+\ \underline{10000011}_{(2)}
\end{array}$$

$11011110 \rightarrow 10100001$ (1. komplement) $= -(2^5 + 2^0) = -\mathbf{33_{(10)}}$.

Primer 26. Koristeći **8 – bitne** označene binarne brojeve izračunati primenom **drugog komplementa** razliku brojeva: $45_{(10)} - 73_{(10)} = ?$, a zatim prikazati rezultat u decimalnom brojnom sistemu.

Rešenje:

$$\begin{array}{r|l} 45 : 2 = 22 & 1 \\ 22 : 2 = 11 & 0 \\ 11 : 2 = 5 & 1 \\ 5 : 2 = 2 & 1 \\ 2 : 2 = 1 & 0 \\ 1 : 2 = 0 & 1 \end{array} \uparrow \qquad \begin{array}{r|l} 73 : 2 = 36 & 1 \\ 36 : 2 = 18 & 0 \\ 18 : 2 = 9 & 0 \\ 9 : 2 = 4 & 1 \\ 4 : 2 = 2 & 0 \\ 2 : 2 = 1 & 0 \\ 1 : 2 = 0 & 1 \end{array} \uparrow$$

$45_{(10)} \rightarrow 00101101_{(2)}$

$73_{(10)} \rightarrow 01001001_{(2)}$

$-73_{(10)} \rightarrow 10110110_{(2)}$ 1.komplement

$+1$

$10110111_{(2)}$ 2.komplement

$45_{(10)} - 73_{(10)} = 45_{(10)} + (-73_{(10)})$

$$\begin{array}{r} 00101101_{(2)} \\ +\ \underline{10110111_{(2)}} \\ \mathbf{11100100_{(2)}} \end{array} \rightarrow -2^7 + 2^6 + 2^5 + 2^2 = \mathbf{-28_{(10)}}$$

ili

$11100100_{(2)}$

$10011011_{(2)}$ (negativan broj 1.komplement)

$\underline{\quad +1}$ (korekcija)

$10011100_{(2)} \rightarrow -(2^4 + 2^3 + 2^2) = \mathbf{-28_{(10)}}$.

Primer 27. Koristeći **8 – bitne** označene binarne brojeve izračunati primenom **drugog komplementa** razliku brojeva: $59_{(10)} - 100_{(10)} = ?$, a zatim prikazati rezultat u decimalnom brojnom sistemu.

Rešenje

$$\begin{array}{r|l} 59 : 2 = 29 & 1 \\ 29 : 2 = 14 & 1 \\ 14 : 2 = 7 & 0 \\ 7 : 2 = 3 & 1 \\ 3 : 2 = 1 & 1 \\ 1 : 2 = 0 & 1 \end{array} \uparrow \qquad \begin{array}{r|l} 100 : 2 = 50 & 0 \\ 50 : 2 = 25 & 0 \\ 25 : 2 = 12 & 1 \\ 12 : 2 = 6 & 0 \\ 6 : 2 = 3 & 0 \\ 3 : 2 = 1 & 1 \\ 1 : 2 = 0 & 1 \end{array} \uparrow$$

$59_{(10)} \rightarrow 00111011_{(2)}$

$100_{(10)} \rightarrow 01100100_{(2)}$

$-100_{(10)} \rightarrow 10011011_{(2)}$ 1. komplement

$+1$

$10011100_{(2)}$ 2. komplement

$59_{(10)} - 100_{(10)} = 59_{(10)} + (-100_{(10)})$

$$\begin{array}{r} 00111011_{(2)} \\ +\ \underline{10011100_{(2)}} \\ \mathbf{11010111_{(2)}} \end{array} \rightarrow -2^7 + 2^6 + 2^4 + 2^2 + 2^1 + 2^0 = \mathbf{-41_{(10)}}$$

ili

$11010111_{(2)}$

$10101000_{(2)}$ (negativan broj 1.komplement)

$\underline{\quad +1}$ (korekcija)

$10101001_{(2)} \rightarrow -(2^5 + 2^3 + 2^0) = \mathbf{-41_{(10)}}$.

Primer 28. Koristeći **8 – bitne** binarne brojeve **sa znakom** izračunati primenom **drugog komplementa** razliku brojeva: $94_{(10)} - 122_{(10)} = ?$, a zatim prikazati rezultat u decimalnom brojnom sistemu.

Rešenje:

122 : 2 = 61	0
61 : 2 = 30	1
30 : 2 = 15	0
15 : 2 = 7	1
7 : 2 = 3	1
3 : 2 = 1	1
1 : 2 = 0	1

94 : 2 = 47	0
47 : 2 = 23	1
23 : 2 = 11	1
11 : 2 = 5	1
5 : 2 = 2	1
2 : 2 = 1	0
1 : 2 = 0	1

$122_{(10)} \rightarrow 01111010_{(2)}$

$-122_{(10)} \rightarrow 10000101_{(2)}$ 1. komplement

$+1$

$10000110_{(2)}$ 2. komplement

$94_{(10)} \rightarrow 01011110_{(2)}$

$94_{(10)} - 122_{(10)} = 94_{(10)} + (-122_{(10)})$

$$\begin{array}{r} 01011110_{(2)} \\ +\ \underline{10000110_{(2)}} \\ \mathbf{11100100_{(2)}} \end{array} \rightarrow -2^7 + 2^6 + 2^5 + 2^2 = \mathbf{-28_{(10)}}.$$

Primer 29. Koristeći **8 – bitne** binarne brojeve **sa znakom** izračunati primenom **drugog komplementa** razliku brojeva: $107_{(10)} - 63_{(10)} = ?$, a zatim prikazati rezultat u decimalnom brojnom sistemu.

Rešenje:

107 : 2 = 53	1
53 : 2 = 26	1
26 : 2 = 13	0
13 : 2 = 6	1
6 : 2 = 3	0
3 : 2 = 1	1
1 : 2 = 0	1

$107_{(10)} \rightarrow 01101011_{(2)}$

63 : 2 = 31	1
31 : 2 = 15	1
15 : 2 = 7	1
7 : 2 = 3	1
3 : 2 = 1	1
1 : 2 = 0	1

$63_{(10)} \rightarrow 00111111_{(2)}$

$-63_{(10)} \rightarrow 11000000_{(2)}$ 1. komplement

$+1$

$11000001_{(2)}$ 2. komplement

$107_{(10)} - 63_{(10)} = 107_{(10)} + (-63_{(10)})$

$$\begin{array}{r} 01101011_{(2)} \\ +\ \underline{11000001_{(2)}} \\ \cancel{1}00101100_{(2)} \end{array} = 2^5 + 2^3 + 2^2 = \mathbf{44_{(10)}}$$

Primer 30. Koristeći 8-bitne označene binarne brojeve izračunati primenom prvog i drugog komplementa razliku brojeva i prikazati rezultat u decimalnom brojnom sistemu.

a) $112_{(10)} - 31_{(10)}$	e) $105_{(10)} - 14_{(10)}$	i) $68_{(10)} - 29_{(10)}$	m) $73_{(10)} - 119_{(10)}$
b) $31_{(10)} - 112_{(10)}$	f) $14_{(10)} - 105_{(10)}$	j) $29_{(10)} - 68_{(10)}$	n) $119_{(10)} - 73_{(10)}$
c) $84_{(10)} - 16_{(10)}$	g) $121_{(10)} - 45_{(10)}$	k) $125_{(10)} - 55_{(10)}$	o) $88_{(10)} - 98_{(10)}$
d) $16_{(10)} - 84_{(10)}$	h) $45_{(10)} - 121_{(10)}$	l) $55_{(10)} - 125_{(10)}$	p) $98_{(10)} - 88_{(10)}$.

3. FORMATI ZAPISA BROJEVA U RAČUNARSKOM SISTEMU

Formatom zapisa brojeva u računarskom sistemu određen je broj cifara za čuvanje celobrojnog i razlomljenog dela brojne vrednosti iz skupa racionalnih brojeva.

- Broj cifara levo zareza definiše opseg brojeva koji mogu da se predstave tim formatom.
- Broj cifara desno od zareza definiše tačnost sa kojom se prikazuju brojevi.
- Istom binarnom zapisu u zavisnosti od formata mogu da odgovaraju različite brojne vrednosti
- Ista brojna vrednost može da bude zapisana u različitim formatima

3.1. Predstavljanje brojeva tehnikom fiksnog zareza

Svaki razlomljeni broj $\mathbf{b_p b_{p-1} ... b_2 b_1 b_0 , b_{-1} b_{-2} ... b_{-q}}$ u opštem slučaju sadrži tri informacije:

- znak (ukoliko je broj označen)
- cifre koje sadrži broj
- poziciju zareza u broju.

Tehnika zapisa brojeva pomoću fiksnog zareza podrazumeva zamenu razlomljenog broja celim brojem koji (najčešće) sadrži iste cifre kao razlomljeni, a poziciju zareza mora da pamti programer. Računske operacije se obavljaju nad celobrojnom zamenom, a programer na kraju koriguje rezultat na osnovu pozicije zareza koju je zapamtio. To omogućava korišćenje standardne, hardverski realizovane, aritmetičlo-logičke jedinice (ALU) računara koja može da radi samo sa celim brojevima. Međutim ova tehnika takođe nosi dosta poteškoća prilikom realizacije složenijih aritmetičkih operacija kao i rizik gubitka rezolucije ili prekoračenja makimalnih vrednosti celobrojne ALU.

Simbol za format predstavljanja **neoznačenih brojeva** tehnikom fiksnog zareza, ako je binarni zapis dužine **N** je:

$$\mathbf{Q_{N-R,R}}$$

N - R = broj cifara ispred fiksnog zareza
R = broj cifara iza fiksnog zareza (radix)

- Na osnovu poznatog formata $\mathbf{Q_{8-R,R}}$ za čuvanje **neoznačenog broja** u 8-bitnom registru, dobija se ekvivalentna vrednost u decimalnom brojnom sistemu primenom formule:

$$x_{(10)} = 2^{-R} \cdot (b_7 \cdot 2^7 + b_6 \cdot 2^6 + ... + b_1 \cdot 2^1 + b_0 \cdot 2^0)$$

Simbol za format predstavljanja **označenih brojeva** tehnikom fiksnog zareza, ako je binarni zapis dužine **N** je:

$$\mathbf{Q^S_{N-R,R}}$$

N - R = broj cifara ispred fiksnog zareza, uključujući 1 bit sa informacijom o znaku (MSB)

R = broj cifara iza fiksnog zareza (radix)

- Na osnovu poznatog formata $\mathbf{Q^S_{N-R,R}}$ za čuvanje **označenog broja** u 8-bitnom registru, dobija se ekvivalentna vrednost u decimalnom brojnom sistemu primenom formule:

$$x_{(10)} = 2^{-R} \cdot (-b_7 \cdot 2^7 + b_6 \cdot 2^6 + ... + b_1 \cdot 2^1 + b_0 \cdot 2^0)$$

Primer 1. Izvršiti konverziju 8-bitnog binarnog broja $x_{(2)} = 01001011_{(2)}$ u decimalan brojni sistem, $x_{(2)} \rightarrow x_{(10)}$, ukoliko je zapis broja:

a) neoznačen celi broj

b) u formatu $\mathbf{Q_{N-R,R}}$ pri čemu je N = 8, a R $\in$ {2, 3, 4 ,5}

c) u formatu $\mathbf{Q^{S}_{N-R,R}}$ pri čemu je N = 8, a R $\in$ {2, 3, 4 ,5}.

Rešenje:

a) Pošto je format zapisa celobrojan i bez znaka, u opštem slučaju binarni broj može da se prikaže kao $x_{(2)} = b_R\, b_{R-1} \ldots b_1\, b_0$, gde su b_R , b_{R-1} , ..., b_1 , $b_0 \in \{0, 1\}$ cifre u binarnom zapisu .
Za konverziju u decimalni brojni sistem $x_{(2)} \rightarrow x_{(10)}$, koristi se opšta formula

$$x_{(10)} = b_R{\cdot}2^R + b_{R-1}{\cdot}2^{R-1} + \ldots + b_1{\cdot}2^1 + b_0{\cdot}2^0$$

Primenom formule za konverziju $x_{(2)} \rightarrow x_{(10)}$ na zadati binarni broj $x_{(2)} = 01001011_{(2)}$, dobija se:

$$x_{(10)} = 0{\cdot}2^7 + 1{\cdot}2^6 + 0{\cdot}2^5 + 0{\cdot}2^4 + 1{\cdot}2^3 + 0{\cdot}2^2 + 1{\cdot}2^1 + 1{\cdot}2^0 = \mathbf{75_{(10)}}.$$

b) Ako je zapis broja u formatu $\mathbf{Q_{N-R,R}}$ pri čemu je:

1) **N = 8**, a **R =2**, to znači da 6 viših binarnih cifara datog broja $x_{(2)} = 01001011_{(2)}$ predstavlja celobrojni deo binarnog broja, dok dve najniže binarne cifre predstavljaju razlomljeni deo, odnosno da binarni broj može da se posmatra kao:

$$x_{(2)} = 010010{,}11_{(2)}$$

Za konverziju $x_{(2)} \rightarrow x_{(10)}$ može da se primeni opšta formula:

$$2^{-R}{\cdot}(\, b_7{\cdot}2^7 + b_6{\cdot}2^6 + \ldots + b_1{\cdot}2^1 + b_0{\cdot}2^0\,)$$

Pošto je R = 2,

$$x_{(10)} = 2^{-2} \cdot (0{\cdot}2^7 + 1{\cdot}2^6 + 0{\cdot}2^5 + 0{\cdot}2^4 + 1{\cdot}2^3 + 0{\cdot}2^2 + 1{\cdot}2^1 + 1{\cdot}2^0) = \mathbf{75/4} = \mathbf{18{,}75_{(10)}}$$

2) **N = 8**, a **R =3**, to znači da 5 viših binarnih cifara datog broja $x_{(2)} = 01001011_{(2)}$ predstavlja celobrojni deo binarnog broja, dok tri najniže binarne cifre predstavljaju razlomljeni deo, odnosno da binarni broj može da se posmatra kao:

$$x_{(2)} = 01001{,}011_{(2)}$$

Za konverziju $x_{(2)} \rightarrow x_{(10)}$ može da se primeni opšta formula:

$$2^{-R}{\cdot}(\, b_7{\cdot}2^7 + b_6{\cdot}2^6 + \ldots + b_1{\cdot}2^1 + b_0{\cdot}2^0\,)$$

Pošto je R = 3,

$$x_{(10)} = 2^{-3}{\cdot}(0{\cdot}2^7 + 1{\cdot}2^6 + 0{\cdot}2^5 + 0{\cdot}2^4 + 1{\cdot}2^3 + 0{\cdot}2^2 + 1{\cdot}2^1 + 1{\cdot}2^0) = \mathbf{75/8{=}\ 9{,}375_{(10)}}$$

Primeri 3) i 4) kada je R = 4 i R = 5 rade se na analogan način kao primeri 1) i 2).

c) Ako je zapis broja u formatu $\mathbf{Q^{S}_{N-R,R}}$ pri čemu je:

1) **N = 8**, a **R =2**, to znači da je na mestu bita najviše težine (MSB) u zapisu sačuvan znak broja (“0” = “+”, “1” = “-”), 5 viših binarnih cifara datog broja $x_{(2)} = 01001011_{(2)}$ predstavlja celobrojni deo binarnog broja, dok 2 najniže binarne cifre predstavljaju razlomljeni deo, odnosno binarni broj može da se posmatra kao:

$$x_{(2)} = 010010{,}11_{(2)} = +10010{,}11_{(2)}$$

Za konverziju $x_{(2)} \rightarrow x_{(10)}$ može da se primeni opšta formula:

$$2^{-R}{\cdot}(-b_7{\cdot}2^7 + b_6{\cdot}2^6 + \ldots + b_1{\cdot}2^1 + b_0{\cdot}2^0)$$

Pošto je R = 2,

$$x_{(10)} = 2^{-2}\cdot(-0\cdot2^7 + 1\cdot2^6 + 0\cdot2^5 + 0\cdot2^4 + 1\cdot2^3 + 0\cdot2^2 + 1\cdot2^1 + 1\cdot2^0) = \mathbf{+75/4 = +18{,}75_{(10)}}$$

2) **N = 8**, a **R =3**, to znači da je na mestu bita najviše težine (MSB) u zapisu sačuvan znak broja (“0” = “+”, “1” = “–“), 4 više binarne cifre datog broja $x_{(2)}$ = $01001011_{(2)}$ predstavljaju celobrojni deo binarnog broja, dok 3 najniže binarne cifre predstavljaju razlomljeni deo, odnosno da binarni broj može da se posmatra kao:

$$x_{(2)} = 01001{,}011_{(2)} = +1001{,}011_{(2)}$$

Za konverziju $x_{(2)} \rightarrow x_{(10)}$ može da se primeni opšta formula:

$$2^{-R}\cdot(-b_7\cdot2^7 + b_6\cdot2^6 + ... + b_1\cdot2^1 + b_0\cdot2^0)$$

Pošto je R = 3,

$$x_{(10)} = 2^{-3}\cdot(-0\cdot2^7 + 1\cdot2^6 + 0\cdot2^5 + 0\cdot2^4 + 1\cdot2^3 + 0\cdot2^2 + 1\cdot2^1 + 1\cdot2^0)= \mathbf{+75/8 = +9{,}375_{(10)}}$$

Primeri 3) i 4) kada je R = 4 i R = 5 rade se na analogan način kao primeri 1) i 2).

Primer 2. Pretvoriti 8-bitni binarni broj $x_{(2)}$ = $11001001_{(2)}$ u decimalan ako je primenjen format $Q_{5,3}$.

Rešenje:

U formatu $Q_{5,3}$ za celobrojni deo se koristi 5 binarnih cifara, a za decimalni deo 3 binarne cifre:

$$x_{(2)} = \underbrace{11001}_{\text{5 cifara}},\underbrace{001}_{\text{3 cifre}} = 2^4 + 2^3 + 2^0 + 2^{-3} = \mathbf{25{,}125_{(10)}}$$

Primer 3. Pretvoriti neoznačen 8-bitni binarni broj $x_{(2)}$ = $01101011_{(2)}$ u decimalan ako je primenjen format $Q_{6,2}$.

Rešenje:

U formatu $Q_{6,2}$ za celobrojni deo se korist 6 binarnih cifara, a za decimalni deo 2 binarne cifre:

$$x_{(2)} = \underbrace{011010}_{\text{6 cifara}},\underbrace{11}_{\text{2 cifre}} = 2^4 + 2^3 + 2^1 + 2^{-1} + 2^{-2} = \mathbf{26{,}75_{(10)}}$$

Primer 4. U memoriji se nalazi heksadecimalni broj $65_{(16)}$. O kom neoznačenom decimalnom broju se radi ako se zna da je zapisan u formatu $Q_{6,2}$.

Rešenje:

$65_{(16)} \rightarrow \mathbf{0110\ 0101_{(2)}}$; ako je format $\mathbf{Q_{6,2}} \Rightarrow \mathbf{011001{,}01_{(2)}} \rightarrow \mathbf{25{,}25_{(10)}}$

Primer 6. U memoriji se nalazi heksadecimalan broj $97_{(16)}$. O kom:

a) neoznačenom

b) označenom

decimalnom broju se radi ako se zna da je zapisan u formatu $Q_{7,1}$.

Rešenje:

a) $97_{(16)} \rightarrow \mathbf{1001\ 0111_{(2)}}$; ako je format $\mathbf{Q_{7,1}} \Rightarrow \mathbf{1001011{,}1_{(2)}} \rightarrow \mathbf{75{,}5_{(10)}}$

b) $97_{(16)} \rightarrow 1001\ 0111_{(2)}$; ako je format $Q^S_{7,1} \Rightarrow$ **$1001011{,}1_{(2)} \rightarrow -52{,}5_{(10)}$**

Primer 7. Koje decimalne vrednosti odgovaraju binarnom zapisu $0000\ 1011_{(2)}$, ako je format:

a) $Q_{8,0}$ d) $Q_{5,3}$

b) $Q_{7,1}$ e) $Q_{4,4}$

c) $Q_{6,2}$ f) $Q_{3,5}$

Rešenje:

a) $0000\ 1011_{(2)} \rightarrow 11_{(10)}$ d) $0000\ 1011_{(2)} \rightarrow 1{,}375_{(10)}$

b) $0000\ 1011_{(2)} \rightarrow 5{,}5_{(10)}$ e) $0000\ 1011_{(2)} \rightarrow 0{,}6875_{(10)}$

c) $0000\ 1011_{(2)} \rightarrow 2{,}75_{(10)}$ f) $0000\ 1011_{(2)} \rightarrow 0{,}34375_{(10)}$

Primer 8. Koje decimalne vrednosti odgovaraju binarnim zapisima:

a) 1100 1010 d) 0100 0110

b) 1010 1010 e) 0011 1010

c) 1001 0101 f) 0111 1001

ako je format $Q^S_{8,0}$; $Q_{5,3}$; $Q_{7,1}$; $Q_{4,4}$; $Q^S_{6,2}$; $Q_{3,5}$; $Q_{2,6}$ i $Q^S_{1,7}$.

Primer 9. Konvertovati 8-bitne brojeve $x_{(2)}$ iz binarnog u decimalan brojni sistem $x_{(2)} \rightarrow x_{(10)}$:

a) Ako su brojevi neoznačeni;
b) Ako je zapis u formatu $Q_{N-R,R}$ pri čemu je $N = 8$, a $R \in \{2, 3, 4, 5\}$;
c) Ako je zapis u formatu $Q^S_{N-R,R}$ pri čemu je $N = 8$, a $R \in \{2, 3, 4, 5\}$.

1) 1110 0011 5) 0001 1011 9) 0011 1110

2) 1000 1010 6) 1110 1111 10) 0101 0101

3) 0100 0111 7) 1100 1100 11) 0101 1100

4) 1111 0110 8) 0100 1101 12) 0011 0100

Pri određivanju **optimalnog formata** treba obezbediti:
- **Dovoljan broj bita** levo od decimalnog zareza za **celobrojni deo**
- **Maksimalan broj bita** desno decimalnog zareza za **razlomljeni deo**

Promenom formata menjaju se tačnost prikaza brojne vrednosti i opseg

Primer 10. Odrediti optimalan format $Q_{8-R,\,R}$ za čuvanje decimalnog broja $4{,}563_{(10)}$.

Rešenje:

U pitanju je format za zapis brojeva bez znaka, tako da u optimalnom formatu treba predvideti minimalan neophodan broj binarnih cifara za zapis celobrojnog dela i maksimalan broj binarnih cifara za zapis dela koji je desno od decimalne tačke:

$4_{(10)} = 100_{(2)}$

$0.563 \cdot 2 = 1{,}126$	1	↓
$0.126 \cdot 2 = 1{,}252$	0	
$0.252 \cdot 2 = 0{,}504$	0	
$0.504 \cdot 2 = 1{,}008$	1	
$0.008 \cdot 2 = 0{,}016$	0	

$4{,}563_{(10)} = 100{,}10010_{(2)}$

Iz prethodnog sledi da je optimalan format za čuvanje 8-bitnog zapisa broja $\mathbf{Q_{3,5}}$.

Primer 11. Odrediti optimalan format $Q_{8-R,\ R}$ za čuvanje decimalnog broja $2{,}872_{(10)}$.

Rešenje:

U pitanju je format za zapis brojeva bez znaka, tako da u optimalnom formatu treba predvideti neophodan broj binarnih cifara za zapis celobrojnog dela i maksimalan broj cifara za zapis dela koji je desno od decimalne tačke:

$2_{(10)} = 10_{(2)}$

$0.872 \cdot 2 = 1{,}744$	1	↓
$0.744 \cdot 2 = 1{,}488$	1	
$0.488 \cdot 2 = 0{,}976$	0	
$0.976 \cdot 2 = 1{,}952$	1	
$0.952 \cdot 2 = 1{,}904$	1	
$0.904 \cdot 2 = 1{,}808$	1	

$2{,}872_{(10)} = 10{,}110111_{(2)}$

Iz prethodnog sledi da je optimalan format za čuvanje 8-bitnog zapisa broja $\mathbf{Q_{2,6}}$.

Primer 12. Kako zapisati neoznačen, a kako označen decimalan broj $3{,}6_{(10)}$ u binarnom brojnom sistemu, $x_{(2)}$, u 8-bitnom registru u formatu fiksnog zareza? U oba slučaja napisati koji bi bio optimalni format? Da li postoji greška usled zaokruživanja i ako postoji da li je veća kada se broj čuva u optimalnom formatu označenog ili neoznačenog broja?

Rešenje:

Za ceo deo, odnosno za je 3 potrebno najmanje dva bita, pa za razlomljeni deo 0,6 ostaje 6 bita.

$$3{,}6_{(10)} \rightarrow 11{,}100110011001_{(2)}$$

Optimalni format bi bio $\mathbf{Q_{2,6}}$. Ekvivalentan binarni zapis broja u ovom formatu je:

$$3{,}6_{(10)} \rightarrow \mathbf{11100110_{(2)}}$$

Ako je u pitanju označen broj, mora da se predvidi jedan bit za znak u celobrojnom delu zapisa, tako da je optimalan format $\mathbf{Q^S_{3,5}}$. Ekvivalentan binarni zapis broja u ovom formatu je:

$$3{,}6_{(10)} \rightarrow \mathbf{01110011_{(2)}}$$

Odsecanje bita u razlomljenom delu binarnog zapisa dovodi do greške u zaokruživanju, koje se ogleda u odstupanju ekvivalentne decimalne vrednosti binarnog zapisa u 8-bitnom registru za izabrani optimalni format.

Ekvivalentna decimalna vrednost neoznačenog broja u optimalnom formatu je:

$$11100110_{(2)} = 2^{-6} \cdot (1 \cdot 2^7 + 1 \cdot 2^6 + 1 \cdot 2^5 + 0 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0) = 3{,}59375_{(10)}$$

Ekvivalentna decimalna vrednost označenog broja u optimalnom formatu je:

$01110011_{(2)} = 2^{-5} \cdot (-0 \cdot 2^7 + 1 \cdot 2^6 + 1 \cdot 2^5 + 1 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0) = 3{,}59375_{(10)}$

Kada se čuva u optimalnom formatu pozitivan označen broj, greška usled zaokruživanja je ista kao kada se u optimalnom formatu čuva neoznačen broj.

Primer 13. Kako zapisati decimalni broj $127{,}7_{(10)}$ u binarnom brojnom sistemu, $x_{(2)}$, u 8-bitnom registru u optimalnom formatu fiksnog zareza? Da li postoji greška usled zaokruživanja?

Rešenje:

Za ceo deo, odnosno za 127 potrebno je 7 bita, pa za razlomljeni deo, 7 ostaje 1 bit.

$$127{,}7_{(10)} \rightarrow 0\ 1111111{,}101\ldots_{(2)}$$

Optimalni format bi bio $\mathbf{Q_{7,1}}$. Ekvivalentan binarni zapis broja u ovom formatu je:

$$127{,}7_{(10)} \rightarrow \mathbf{11111111_{(2)}}$$

Ekvivalentna decimalna vrednost označenog broja u optimalnom formatu je:

$$11111111_{(2)} = 2^{-1} \cdot (1 \cdot 2^7 + 1 \cdot 2^6 + 1 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0) = 127{,}5_{(10)}$$

Primer 14. Odrediti optimalan format za čuvanje broja π ($3{,}1416_{(10)}$) u osmobitnom zapisu, ako je:

a) broj neoznačen

b) broj označen

Koji je optimalni format?

Rešenje:

a) $3{,}1416_{(10)}$ (neoznačen) → $\mathbf{11001001_{(2)}}$ u formatu $\mathbf{Q_{2,6}}$

b) $3{,}1416_{(10)}$ (označen) → $\mathbf{01100100_{(2)}}$ u formatu $\mathbf{Q^S_{3,5}}$.

Primer 15. Kako bi u memoriji osmobitnog računara izgledao optimalan zapis decimalnog broj $66{,}6_{(10)}$?

a) u sistemu neoznačenih brojeva

b) u sistemu označenih brojeva

Za oba slučaja naznačiti formate u kojima je zapisan broj.

Rešenje:

a) $66{,}6_{(10)}$ (neoznačen) → $\mathbf{10000101_{(2)}}$ u formatu $\mathbf{Q_{7,1}}$

b) $66{,}6_{(10)}$ (označen) → $\mathbf{01000010_{(2)}}$ u formatu $\mathbf{Q^S_{8,0}}$

Primer 16. Koji hekasadecimalan sadržaj predstavlja optimalni zapis decimalnog broja $9{,}87_{(10)}$ u osmobitnom sistemu:

a) neoznačenih brojeva

b) označenih brojeva

Naznačiti format u kome je broj zapisan.

Rešenje:

a) $9{,}87_{(10)}$ (neoznačeni) → $10011101_{(2)}$ = $\mathbf{9D_{(16)}}$ u formatu $\mathbf{Q_{4,4}}$

b) $9{,}87_{(10)}$ (označeni) → $01001110_{(2)}$ = $\mathbf{4E_{(16)}}$ u formatu $\mathbf{Q^S_{5,3}}$

Primer 17. Kako bi u memoriji osmobitnog računara izgledao optimalni zapis decimalnog broja $34{,}6_{(10)}$ u sistemu:

a) neoznačenih brojeva

b) označenih brojeva

Naznačiti format u kome je broj zapisan.

Rešenje:

a) $34{,}6_{(10)}$ (neoznačeni) → $10001010_{(2)}$ u formatu $\mathbf{Q_{6,2}}$

b) $34{,}6_{(10)}$ (označeni) → $01000101_{(2)}$ u formatu $\mathbf{Q^S_{7,1}}$

Primer 18. Koji hekasadecimalan sadržaj predstavlja optimalni zapis decimalnog broja $6{,}283_{(10)}$ u 8-bitnom sistemu:

a) neoznačenih brojeva

b) označenih brojeva

Naznačiti format u kome je broj zapisan.

Rešenje:

a) $6{,}283_{(10)}$ (neoznačeni) → $11001001_{(2)}$ = $\mathbf{C9_{(16)}}$ u formatu $\mathbf{Q_{3,5}}$

b) $6{,}283_{(10)}$ (označeni) → $01100100_{(2)}$ = $\mathbf{64_{(16)}}$ u formatu $\mathbf{Q^S_{4,4}}$

Primer 19. U šesnaestobitnom zapisu sa fiksnim zarezom, koristeći format koji je optimalan sa gledišta tačnosti zapisa, predstaviti decimalni broj $1025{,}608_{(10)}$ kao:

a) neoznačen broj

b) označen broj.

Odgovore dati kao dva šesnaestobitna (četvorocifrena) heksadecimalna broja.

Rešenje:

Pretvaranjem u binarni sistem celobrojnog dela dobijamo

$1025 = 10000000001_{(2)}$ (zapis dužine 11 bita)

Pretvaranjem razlomljenog dela (uzastopnim množenjem razlomljenog dela sa 2) dobijamo

$0{,}608 = 0{,}100110111010\ldots_{(2)}$ (zapis beskonačne dužine)

Ispostaviće se da nam toliki broj decimala nije bio neophodan ali je dat u rešenju samo radi kontrole.

Dakle, binarni zapis zadatog razlomljenog decimalnog broja je:

$$1025{,}608_{(10)} = 10000000001{,}100110111010\ldots_{(2)}$$

a) Za zapis celobrojnog dela ovog broja (kada se on tretira kao neoznačeni broj) treba uzeti najmanje 11 bita. Tolika je, naime, dužina binarnog zapisa celobrojnog dela i to je neophodan i minimalan broj bita koji se mora "potrošiti" da bi se ovaj broj prikazao, bez obzira na željenu tačnost. Ako bi pokušali da zapišemo ovaj broj sa manje od 11 bita, dobili bi broj koji ne odgovara originalu. Kako smo na zapis celobrojnog dela "potrošili" 11 bita, a radimo u šesnaestobitnom formatu, ostatak do 16 (pet bita) treba upotrebiti za zapis binarnih cifara levo od zareza. Sa gledišta tačnosti zapisa, potrebno je uzeti što veći broj binarnih cifara levo od zareza, a u ovom slučaju je to 5 jer je toliko ostalo. Ako bi celobrojni deo zapisali sa više od 11 bita (što je moguće) ostalo bi nam manje bita za zapis razlomljenog dela što bi umanjilo tačnost zapisa. Na osnovu ovoga, šesnaest bita koja treba uzeti za zapis sa maksimalnom mogućom tačnošću su biti koji su podebljani:

$1025{,}608_{(10)} = \ldots00000\mathbf{10000000001{,}10011}0111010\ldots_{(2)}$

Za bite koji nisu podebljani iza zareza, na žalost, nema mesta u 16-bitnim formatima. Tačnost bi bila veća kada bi uzeli više njih ali to ne smemo da uradimo na račun celobrojnog dela. Nule koje su dopisane spreda ne utiču na vrednost pozitivnog broja, a dopisane radi ilustracije izbora optimalnog formata. Dakle, ovaj broj treba zapisati u formatu $\mathbf{Q_{11,\,5}}$ (jedanaest bita ispred i pet iza zareza):

$1025{,}608_{(10)} = 1000\ 0000\ 0011\ 0011_{(2)} = \mathbf{8033_{(16)}}$ u formatu $\mathbf{Q_{11,5}}$

Da se podsetimo: *U formatu fiksnog zareza, zarez se ne pojavljuje u zapisu broja, već format nosi informaciju o njegovom položaju.*

Ako bi broj bita u zapisu razlomljenog dela bio konačan i manji od 5, tačnost zapisa bi bila maksimalna (potpuna) i sa manjim brojem bita. Na primer, za decimalni broj 0,5 potrebna je samo jedna binarna cifra za zapis razlomljenog dela ($0{,}5_{(10)} = 0{,}1_{(2)}$) i taj jedan bit iza zareza obezbeđuje potpunu tačnost. Kada je broj bita u binarnom zapisu iza zareza veći od raspoloživog broja, potrebno je celobrojni deo zapisati sa najmanjim mogućim brojem bita.

b) Da bi se broj 1025,608 predstavio kao označen treba uzeti najmanje 12 bita ispred zareza. Sada više nije dovoljan jedanaestobitni zapis celobrojnog dela iako je upravo toliki broj bita koji dobijamo prevođenjem $1025_{(10)}$ u binarni sistem. Broj $1025_{(10)}$ je pozitivan, a ako bi bio zapisan u jedanaestobitnom formatu, bit najveće pozicione vrednost bi bio jedinica što bi ukazivalo da je broj negativan. I zaista, sa 11 bita u sistemu označenih brojeva moguće je prikazati brojeve od -2^{10} do $2^{10}-1$, dakle od $-1024_{(10)}$ do $+1023_{(10)}$, a zadati broj $1025_{(10)}$ ne spada u taj opseg.

Zapis broja predstavljen sa 12 bita je:

$1025_{(10)} = 0100\ 0000\ 0001_{(2)}$

U bitu najveće težine sačuvana informacija o znaku (bit 0 ukazuje da je broj pozitivan).

Sada od binarnog kôda razlomljenog dela možemo uzeti samo 4 bita jer je toliko preostalo:

$0{,}608_{(10)} = 0{,}\underbrace{\mathbf{1001}}10111010\ldots$

četiri bita koja mogu da uđu u zapis (12 je neophodno za zapis) celobrojnog

Ako ponovo posmatramo traženi broj pretvoren u binarni kôd, videćemo u čemu se razlikuje optimalni zapis označenog od optimalnog zapisa neoznačenog broja.

$1025{,}608_{(10)} = \ldots0000\mathbf{010000000001{,}1001}10111010\ldots_{(2)}$

Prema tome $1025{,}608_{(10)}$ kao označen broj treba zapisati u formatu $\mathbf{Q^S_{12,4}}$:

$\mathbf{1025{,}608_{(10)} = 0100\ 0000\ 0001\ 1001_{(2)} = 4019_{(16)}}$ **u formatu** $\mathbf{Q^S_{12,4}}$.

Primer 20. U šesnaestobitnom zapisu sa fiksnim zarezom, koristeći format koji je optimalan sa gledišta tačnosti zapisa, predstaviti decimalni broj $-71{,}75_{(10)}$. Odgovor dati kao šesnaestobitni (četvorocifreni) heksadecimalan broj.

Rešenje:

Broj je negativan pa se samim tim može predstaviti jedino kao označen broj. Kao i kod celih negativnih brojeva i ovde treba poći od broja +71,75 pa njegovom binarnom kôdu odgovarajuće dužine promeniti znak tako što se pronađe drugi komplement tog zapisa. Binarni kôd broja +71,75 se dobija tako što se prvo pronađe binarni kôd broja 71 uzastopnim deljenjem sa 2, a zatim binarni kôd broja 0,75 uzastopnim množenjem. Broj bita iza zareza je konačan i prevođenjem na binarni zapis se ne gubi tačnost.

$$71_{(10)} = 1000111_{(2)}$$

$$0{,}75_{(10)} = 0{,}11_{(2)}$$

Dakle, broj $\mathbf{+71{,}75_{(10)}} = \mathbf{1000111{,}11_{(2)}}$

Kako je broj binarnih cifara iza zareza konačan (postoje samo dve binarne cifre), to postoji više šesnaestobitnih formata pomoću kojih se može zapisati ovaj broj sa potpunom tačnošću. To može biti format $Q^S_{14,\,2}$ sa 14 bita za celobrojni deo (neophodno je bar 8) i 2 bita iza zareza (tačno toliko je i potrebno). Podjednako dobar je i format $Q^S_{13,\,3}$ kao i $Q^S_{12,\,4}$ i tako dalje sve do formata $Q^S_{8,\,8}$.

> *Da se podsetimo:*
> *I ovde je nužno broj predstaviti kao označen jer ćemo mu kasnije menjati znak, a operacija promene znaka (drugi komplement) nema smisla nad neoznačenim brojevima. To je i razlog što je potrebno najmanje osam bita za zapis broja +71. Obratite pažnju da ovaj broj kada se pretvori u binarni ima sedam bita koliko je potrebno kada bi ga tretirali kao neoznačen, ali za zapis označenog broja +71 nužna su osam bita.*

Ako usvojimo format $Q^S_{14,\,2}$

$+71{,}75_{(10)} = 0000\ 0001\ 0001\ 1111_{(2)}$ u formatu $Q^S_{14,2}$

Negativan broj se dobija kao drugi komplement pozitivnog broja iste apsolutne vrednosti.

$-71{,}75_{(10)} = 1111\ 1110\ 1110\ 0001_{(2)} = \mathbf{FEE1_{(16)}}$ $(\mathbf{Q^S_{14,2}})$

Podjednako dobro rešenje je i, na primer:

$-71{,}75_{(10)} = 1011\ 1000\ 0100\ 0000_{(2)} = \mathbf{B840_{(16)}}$ $(\mathbf{Q^S_{8,8}})$

kao i svi formati između ova dva.

> **! Česta greška:**
> Pogrešno je najpre pronaći kôd za broj $-71_{(10)}$ pa tom kôdu dodati kôd za $0{,}75_{(10)}$.
>
> $+71_{(10)} = 0000\ 0000\ 0100\ 0111_{(2)}$ pa je $-71_{(10)} = 1111\ 1111\ 1011\ 1001_{(2)}$ (drugi komplement)
> $0{,}75_{(10)} = 0{,}11_{(2)}$
>
> Kombinacijom ova dva dolazi se do pogrešnog rezultata:
>
> $-71{,}75_{(10)} = 11\ 1111\ 1011\ 1001{,}11_{(2)}$
>
> Broj $-71{,}75_{(10)} = -(71+0{,}75)_{(10)}$, dakle promena znaka se mora uraditi **posle** dodavanja 0,75.
>
> Moguće je broju –71 dodati broj –0,75 (dakle, drugi komplement zapisa broja 0,75, a ne sam taj broj) i dobio bi se tačan rezultat. Ovo sabiranje zbog neminovno različitih formata traži konverziju formata i znatno je složenije, ali daje tačan rezultat.

Primer 21. Koji se heksadecimalan broj nalazi u memoriji šesnaestobitnog računara kao zapis decimalnog broja:

a) $0{,}875_{(10)}$

b) $-0{,}875_{(10)}$,

ako se zna da je zapisan kao označeni broj u formatu $Q^S_{1,15}$?

Rešenje:

a) $0{,}875_{(10)} \rightarrow 0{,}111_{(2)}$

$0{,}875_{(10)} \rightarrow$ **0111 0000 0000 0000**$_{(2)}$ u formatu $Q^S_{1,15} \rightarrow$ **7000**$_{(16)}$

b) $-0{,}875_{(10)} \Rightarrow 0{,}875_{(10)} \rightarrow 0111\ 0000\ 0000\ 0000_{(2)}$

$$\begin{array}{rl} 1000\ 1111\ 1111\ 1111_{(2)} & \text{1. kom} \\ +1 & \\ \hline 1001\ 0000\ 0000\ 0000_{(2)} & \text{2. kom} \end{array}$$

$-0{,}875_{(10)} \rightarrow$ **1001 0000 0000 0000**$_{(2)}$ u formatu $Q^S_{1,15} \rightarrow$ **9000**$_{(16)}$.

Primer 22. U šesnaestobitnom zapisu sa fiksnim zarezom, koristeći format koji je optimalan sa gledišta tačnosti zapisa, predstaviti decimalan broj -1,5. Odgovor dati kao šesnaestobitni (četvorocifreni) heksadecimalni broj.

Rešenje:

$+1{,}5_{(10)} = 01{,}1_{(2)}$ ili $0000\ 0000\ 0000\ 0011_{(2)}$ u formatu $Q^S_{15,1}$ primenom drugog komplementa:

$-1{,}5_{(10)} = 1111\ 1111\ 1111\ 1101_{(2)}$ = **FFFD**$_{(16)}$ u formatu $\mathbf{Q^S_{15,1}}$.

Podjednako dobra rešenja se mogu dati i u drugim formatima: $Q^S_{14,2}$, $Q^S_{13,3}$ i tako dalje, sve do $Q^S_{2,14}$.

Primer 23. Koji je najmanji, a koji najveći pozitivan broj koji se može prikazati u formatu $Q^S_{6,2}$?

Rešenje:

Najmanji pozitivan broj u formatu $Q^S_{6,2}$ je **000000,01**$_{(2)}$ što iznosi $\mathbf{2^{-2} = 0{,}25_{(10)}}$.

Najveći pozitivan broj u formatu $Q^S_{6,2}$ je **011111,11**$_{(2)}$ što iznosi **31,75**$_{(10)}$.

Da se podsetimo:
Bit najveće pozicione vrednosti pozitivnog broja mora biti nula.

Primer 24. Koji je najmanji po apsolutnoj vrednosti, a koji najveći po apsolutnoj vrednosti (najnegativniji) negativan broj koji se može prikazati u formatu $Q^S_{6,2}$?

Rešenje:

Najmanji negativan broj u formatu $Q^S_{6,2}$ je **111111,11**$_{(2)}$ što iznosi **–0,25**$_{(10)}$

Najveći negativan broj u formatu $Q^S_{6,2}$ je **100000,00**$_{(2)}$ što iznosi **–32**$_{(10)}$

Brojevi mogu da se sabiraju ili oduzimaju samo ako su u istom formatu!

Primer 25. Opisati binarno sabiranje brojeva $64{,}87_{(10)}$ i $-3{,}14_{(10)}$ u osmobitnoj aritmetici fiksnog zareza. Koji decimalni broj je rezultat ovakvog sabiranja?

Rešenje:

$64{,}87_{(10)}$ se mora predstaviti u označenom formatu (jer treba da se sabira sa negativnim brojem). Stoga je nužno celobrojni deo predstaviti sa minimalno 8 bita (sa 7 bita, opseg označenih brojeva je od $-63_{(10)}$ do $+63_{(10)}$, što nije dovoljno) dakle jedini osmobitni format koji dolazi u obzir je $Q^s_{8,0}$.

Predstavljajući $64{,}87_{(10)}$ u ovom formatu gubimo informaciju o razlomljenom delu, ali u osmobitnoj aritmetici to je najbolje što može da se uradi.

$64{,}87_{(10)}$ je $\mathbf{01000000_{(2)}}$ u formatu $\mathbf{Q^S_{8,0}}$. Da bi mogao da se sabira sa njim, i $-3{,}14_{(10)}$ se mora predstaviti u istom formatu ($Q^s_{8,0}$).

$+3{,}14_{(10)} = 011{,}00100\ldots_{(2)}$ što je $00000011_{(2)}$ u formatu $Q^S_{8,0}$.

$-3{,}14_{(10)}$ se dobija kao drugi komplement ovog zapisa:

$-3{,}14_{(10)} = \mathbf{11111101_{(2)}}$ formatu $\mathbf{Q^S_{8,0}}$

$$\begin{array}{rll} & 01000000 = 64{,}87_{(10)} & (Q^S_{8,0}) \\ + & \underline{11111101} = -3{,}14_{(10)} & (Q^S_{8,0}) \\ \cancel{1} & \mathbf{00111101} = 61_{(10)} & (Q^S_{8,0}) \end{array}$$

Dakle, rezultat je $\mathbf{61_{(10)}}$

Primer 26. Opisati binarno sabiranje brojeva $32{,}17_{(10)}$ i $-9{,}81_{(10)}$ u osmobitnoj aritmetici fiksnog zareza. Koji decimalni broj je rezultat ovakvog sabiranja?

Rešenje:

$32{,}17_{(10)}$ je $\mathbf{01000000_{(2)}}$ u formatu $\mathbf{Q^s_{7,1}}$ zbog toga se i $-9{,}81_{(10)}$ mora predstaviti u istom formatu.

$+9{,}81_{(10)} = 1001{,}1101\ldots_{(2)}$ što je $00010011_{(2)}$ formatu $\mathbf{Q^s_{7,1}}$.

$-9{,}81_{(10)}$ se dobija kao 2. komplement ovog zapisa:

$-9{,}81_{(10)} = \mathbf{11101101_{(2)}}$ formatu $\mathbf{Q^s_{7,1}}$

$$\begin{array}{rll} & 01000000 = 32{,}17_{(10)} & (Q^S_{7,1}) \\ + & \underline{11101101} = -9{,}81_{(10)} & (Q^S_{7,1}) \\ \cancel{1} & \mathbf{00101101} = 22{,}5_{(10)} & (Q^S_{7,1}) \end{array}$$

Dakle, rezultat je $\mathbf{22{,}5_{(10)}}$.

3.2. Predstavljanje brojeva tehnikom pokretnog zareza (*floating point*)

Eksponencijalno predstavljanje brojeva je nastalo iz potrebe zapisa jako velikih brojeva i brojeva koji su po apsolutnoj vrednosti jako mali (bliski nuli). U opštem slučaju, eksponencijalni zapis razlomljenog broja $\boldsymbol{x}$ iz brojnog sistema sa osnovom $\boldsymbol{S}$ može da se zapiše u obliku:

$$x_{(S)} = \boldsymbol{Z}\ \boldsymbol{M} \cdot \boldsymbol{S}^{\boldsymbol{E}}$$

Ovaj zapis sadrži tri informacije o razlomljenom binarnom broju:

- znak mantise (***Z***) -
- mantisu (***M***)
- eksponent (***E***)

Znak mantise (***Z***) se posmatra odvojeno od apsolutne vrednosti i definisan je kao **0** za pozitivne i **1** za negativne brojeve.

Cifre koje čine broj nalaze u okviru mantise ***M***. Mantisa ***M*** se često predstavlja u **normalizovanom** obliku. Normalizacija podrazumeva da je cifra mantise na poziciji –1 (prva cifra desno od zareza) različita od nule, a sve cifre na pozicijama levo od zareza su nule, odnosno, da je mantisa oblika **0**,***d***… $_{(S)}$ gde je ***d*** cifra **različita** od **nule**. Mantisa normalizovanog **decimalnog** broja je uvek u opsegu: $\mathbf{0{,}1} \le \boldsymbol{M} < \mathbf{1}$

Eksponent ***E*** sadrži informaciju o položaju decimalnog zareza. Eksponent može biti i pozitivan i negativan broj.

Napomena:
Normalizacija je stvar dogovora (standarda) i moguće je definisati različito. U delu literature koristi se normalizacija kod koje je cifra mantise na poziciji 1 (prva levo od zareza) različita od nule. Drugačije definisana normalizacija povlači za sobom različite vrednosti i mantise i eksponenta.

Primer 27. Broj $0{,}0000000123_{(10)}$ predstaviti u normalizovanom eksponencijalnom obliku.

Rešenje:

Broj $\mathbf{0{,}123_{(10)} \cdot 10^{-7}}$ predstavlja **normalizovani eksponencijalni** oblik zadatog broja, u kome je:

- znak ***Z*** **= 0**
- normalizovana mantisa ***M*** = $\mathbf{0{,}123_{(10)}}$
- osnova brojnog sistema ***S*** = **10**
- eksponent ***E*** = $\mathbf{-7_{(10)}}$

Primer 28. Broj $205641{,}45_{(10)}$ predstaviti u normalizovanom eksponencijalnom obliku.

Rešenje:

Broj $\mathbf{0{,}20564145_{(10)} \cdot 10^{6}}$ predstavlja **normalizovani** eksponencijalni oblik zadatog broja, u kome je:

- znak ***Z*** **= 0**
- normalizovana mantisa ***M*** **=** $\mathbf{0{,}20564145_{(10)}}$
- osnova brojnog sistema ***S*** = **10**
- eksponent ***E*** **=** $\mathbf{6_{(10)}}$

Primer 29. Broj $0{,}000325_{(10)}$ predstaviti u normalizovanom eksponencijalnom obliku.

Rešenje:

Broj $\mathbf{0{,}325_{(10)}\cdot 10^{-3}}$ predstavlja **normalizovani** eksponencijalni oblik zadatog broja, u kome je:

- znak $\boldsymbol{Z} = \mathbf{0}$
- normalizovana mantisa $\boldsymbol{M} = \mathbf{0{,}325_{(10)}}$
- osnova brojnog sistema $\boldsymbol{S} = \mathbf{10}$
- eksponent $\boldsymbol{E} = \mathbf{-3_{(10)}}$

Primer 30. Broj $-325_{(10)}$ predstaviti u normalizovanom eksponencijalnom obliku.

Rešenje:

Broj $-\,\mathbf{0{,}325_{(10)}\cdot 10^{3}}$ predstavlja **normalizovani** eksponencijalni oblik zadatog broja, u kome je:

- znak $\boldsymbol{Z} = \mathbf{1}$
- normalizovana mantisa $\boldsymbol{M} = \mathbf{0{,}325_{(10)}}$
- osnova brojnog sistema $\boldsymbol{S} = \mathbf{10}$
- eksponent $\boldsymbol{E} = \mathbf{3_{(10)}}$

U slučaju binarnih brojeva prva cifra mantise desno od zareza je uvek 1 (jedina cifra različita od nule u binarnom sistemu) odnosno, mantisa je oblika $\mathbf{0{,}1\ldots}_{(2)}$. Pod tim uslovima, mantisa (***m***) normalizovanog binarnog broja je uvek $\mathbf{0{,}5 \le \boldsymbol{M} < 1}$. Binarni broj se pomera nalevo ili nadesno sve dok prva cifra iza zareza ne bude 1.

Pomeranje nalevo ili nadesno predstavlja množenje sa 2^{E} ili deljenje sa 2^{E} gde ***E*** odgovara broju pomeranja.

Napomena: *U delu literature broj se svodi na mantisu oblika* (1,....) *odnosno, na opseg između* 1 *i* 2 *(*$1 \le$ *mantisa* < 2*).*

Primer 31. Broj $100110{,}01_{(2)}$ predstaviti u normalizovanom eksponencijalnom obliku.

Rešenje:

Broj $\mathbf{0{,}10011001}_{(2)}\cdot\mathbf{2^{6}}$ predstavlja **normalizovani eksponencijalni** oblik zadatog broja, u kome je:

- znak $\boldsymbol{Z} = \mathbf{0}$
- normalizovana mantisa $\boldsymbol{M} = \mathbf{0{,}10011001_{(2)}}$
- osnova brojnog sistema $\boldsymbol{S} = \mathbf{2}$
- eksponent $\boldsymbol{E} = \mathbf{6_{(10)}}$

Primer 32. Broj $0{,}0001001_{(2)}$ predstaviti u normalizovanom eksponencijalnom obliku.

Rešenje:

Broj $\mathbf{0{,}1001_{(2)}\cdot 2^{-3}}$ predstavlja **normalizovani eksponencijalni** oblik zadatog broja, u kome je:

- znak $\boldsymbol{Z} = \mathbf{0}$
- normalizovana mantisa $\boldsymbol{M} = \mathbf{0{,}1001_{(2)}}$
- osnova brojnog sistema $\boldsymbol{S} = \mathbf{2}$
- eksponent $\boldsymbol{E} = \mathbf{-3_{(10)}}$

Primer 33. Decimalni broj **0,09375**$_{(10)}$ predstaviti kao binarni u eksponencijalnom obliku ($\boldsymbol{M \cdot 2^{E}}$) sa normalizovanom mantisom ($\boldsymbol{M}$). Odrediti mantisu ($\boldsymbol{M}$) u formatu $Q_{0,24}$ i eksponent ($\boldsymbol{E}$) kao decimalni broj.

Rešenje:

$\mathbf{0{,}09375_{(10)}} = \mathbf{0{,}00011_{(2)}} = \mathbf{0{,}11_{(2)} \cdot 2^{-3}}$ predstavlja **normalizovani** eksponencijalni oblik zadatog broja, u kome je:

- znak $\boldsymbol{Z} = \mathbf{0}$
- normalizovana mantisa $\boldsymbol{M} = \mathbf{1100\ 0000\ 0000\ 0000\ 0000\ 0000_{(2)}}$ u formatu $\mathbf{Q_{0,24}}$
- osnova brojnog sistema $\boldsymbol{S} = \mathbf{2}$
- $\boldsymbol{E} = \mathbf{-3_{(10)}}$

Po standardu **1-7-24** razlomljeni binarni broj se upisuje u računar kao 32-bitni podatak. Binarni broj se najpre predstavi u eksponencijalnoj normalizovanoj formi, na osnovu čega može da se odredi vrednost znaka, mantise i eksponenta.

Tridesetdvobitni zapis po standardu **1-7-24** sadrži:

- **Polje znaka** (***Z***) (bit **31**) sadrži 0 ako je broj pozitivan ili sadrži 1 ako je broj negativan.
- **Polje uvećanog eksponenta (*E*+64)** (sedam bita, od bita **30** do bita **24**) sadrži eksponent ***E*** uvećan za 64. Eksponent je uvećan za iznos koji se naziva „**višak**“, u ovom standardu 64, da bi broj koji se upisuje u ovo polje uvek bio pozitivan. Vrednost viška je uvek definisana standardom. Polje uvećanog eksponenta se često označava sa ***b*** (od *biased exponent*).
- **Polje mantise (mantisa)** (od bita **23** do bita **0**) sadrži mantisu predstavljenu u formatu $Q_{(0,24)}$, odnosno 24 cifre mantise. Na mestu bita 23 je prva cifra mantise desno od zareza (cifra mantise na poziciji –1) i tako dalje. Ukoliko mantisa nema 24 cifre, na mestima odgovarajućih bitova zapisa nalaze se nule.

31	30 29 28 27 26 25 24	23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
Z	***E*+64**	**mantisa**

Format 1-7-24 koristi višak 64 što čini opseg dozvoljenih vrednosti eksponenta od –64 do +63 (opseg je isti kao opseg označenih 7-bitnih brojeva). Višak se bira tako da za bilo koji eksponent (koji može biti i pozitivan i negativan) polje uvećanog eksponenta bude uvek pozitivno. Polje uvećanog eksponenta se često označava sa ***b*** (od *biased exponent*).

- Najveći broj koji može da se upiše po standardu **1-7-24** ima eksponent 63 i mantisu blisku jedinici, što je približno 2^{63}, odnosno oko $9{,}2 \cdot 10^{18}$.
- Najmanji broj (po apsolutnoj vrednost) koji može da se upiše po standardu **1-7-24** je onaj čiji eksponent nije manji od –64. S obzirom da je najmanja mantisa 0,5, taj najmanji broj je $0{,}5 \cdot 2^{-63}$, što je oko $5{,}4 \cdot 10^{-20}$.

Primer 34. Broj $13{,}25_{(10)}$ predstaviti u formatu **1-7-24**. Kom heksadecimalnom broju odgovara ekvivalentan binarni zapis ovog broja u navedenom formatu?

Rešenje:

$\mathbf{13{,}25_{(10)}} = 1101{,}01_{(2)} = \mathbf{0{,}110101_{(2)}} \cdot 2^{4}$ predstavlja **normalizovani** eksponencijalni oblik zadatog broja, u kome je:

- $Z = 0$ — bit **31** je **nula**, jer je broj pozitivan
- $E = 4 \Rightarrow b = E + 64 = 68_{(10)} = 1000100_{(2)}$ — biti **30** do **24** sadrže uvećani eksponent ***b***
- $M = 1101\ 0100\ 0000\ 0000\ 0000\ 0000_{(2)}$ — biti **23** do **0** sadrže cifre mantise

31	30 29 28 27 26 25 24	23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
0	**1000100**	**1101 0100 0000 0000 0000 0000**

32-bitni zapis: $\mathbf{13{,}25_{(10)} = 0100\ 0100\ 1101\ 0100\ 0000\ 0000\ 0000\ 0000_{(2)} = 44D40000_{(16)}}$

Primer 35. Odrediti 32-bitni binarni zapis broja $-61{,}6875_{(10)}$ u formatu pokretnog zareza **1-7-24**.

Rešenje:

$\mathbf{-61{,}6875_{(10)}} = -111101{,}1011_{(2)} = \mathbf{-0{,}1111011011_{(2)}} \cdot 2^6$ predstavlja **normalizovani** eksponencijalni oblik zadatog broja, u kome je:

- $Z = 1$ — bit **31** je **jedan**, jer je broj **negativan**
- $E = 6 \Rightarrow b = E + 64 = 70_{(10)} = 1000110_{(2)}$ — biti **30** do **24** sadrže uvećani eksponent ***b***
- $M = 1111\ 0110\ 1100\ 0000\ 0000\ 0000_{(2)}$ — biti **23** do **0** sadrže cifre mantise

32-bitni zapis: $\mathbf{-61{,}6875_{(10)} = 1100\ 0110\ 1111\ 0110\ 1100\ 0000\ 0000\ 0000_{(2)}}$

Primer 36. Odrediti 32-bitni binarni zapis broja $+65{,}75_{(10)}$ u formatu pokretnog zareza **1-7-24**.

Rešenje:

$\mathbf{+65{,}75_{(10)}} = 1000001{,}11_{(2)} = \mathbf{0{,}100000111_{(2)}} \cdot 2^7$ predstavlja **normalizovani** eksponencijalni oblik zadatog broja, u kome je:

- $Z = 0$ — bit **31** je **nula**, jer je broj **pozitivan**
- $E = 7 \Rightarrow b = E + 64 = 71_{(10)} = 1000111_{(2)}$ — biti 30 do 24 sadrže uvećani eksponent ***b***
- $M = 1000\ 0011\ 1000\ 0000\ 0000\ 0000_{(2)}$ — biti 23 do 0 sadrže cifre mantise

32-bitni zapis: $\mathbf{+65{,}75_{(10)} = 0100\ 0111\ 1000\ 0011\ 1000\ 0000\ 0000\ 0000_{(2)}}$

Primer 37. Odrediti 32-bitni binarni zapis broja $+75_{(10)}$ u formatu pokretnog zareza **1-7-24**.

Rešenje:

$\mathbf{+75_{(10)}} = 1001011_{(2)} = \mathbf{0{,}1001011_{(2)}} \cdot 2^7$ predstavlja **normalizovani** eksponencijalni oblik zadatog broja, u kome je:

- $Z = 0$ — bit **31** je **nula**, jer je broj **pozitivan**
- $E = 7 \Rightarrow b = E + 64 = 7_{(10)}1 = 1000111_{(2)}$ — biti **30** do **24** sadrže uvećani eksponent ***b***
- $M = 1001\ 0110\ 0000\ 0000\ 0000\ 0000_{(2)}$ — biti **23** do **0** sadrže cifre mantise

32-bitni zapis: $\mathbf{+75_{(10)} = 0100\ 0111\ 1001\ 0110\ 0000\ 0000\ 0000\ 0000_{(2)}}$

Primer 38. Odrediti 32-bitni binarni zapis broja $-75{,}5_{(10)}$ u formatu pokretnog zareza **1-7-24**.

Rešenje:

$\mathbf{-75{,}5_{(10)}} = -1001011{,}1_{(2)} = \mathbf{-0{,}10010111_{(2)}} \cdot 2^7$ predstavlja **normalizovani** eksponencijalni oblik zadatog broja, u kome je:

- ***Z*** = **1** — bit **31** je **jedan**, jer je broj **negativan**
- ***E*** = 7 $\Rightarrow$ ***b* = *E***+64 **= 71**$_{\mathbf{(10)}}$ = **1000111**$_{\mathbf{(2)}}$ — biti **30** do **24** sadrže uvećani eksponent ***b***
- ***M*** = **1001 0111 0000 0000 0000 0000**$_{\mathbf{(2)}}$ — biti **23** do **0** sadrže cifre mantise

32-bitni zapis: **-75,5**$_{\mathbf{(10)}}$ = **1100 0111 1001 0111 0000 0000 0000 0000**$_{\mathbf{(2)}}$

Primer 39. Odrediti 32-bitni binarni zapis broja -95,5$_{(10)}$ u formatu pokretnog zareza **1-7-24**.

Rešenje:

-95,5$_{\mathbf{(10)}}$ = -1011111,1$_{(2)}$ = **-0,10111111**$_{\mathbf{(2)}}\cdot 2^7$ predstavlja **normalizovani** eksponencijalni oblik zadatog broja, u kome je:

- ***Z*** = **1** — bit **31** je **jedan**, jer je broj **negativan**
- ***E*** = 7 $\Rightarrow$ ***b* = *E***+64 **= 71**$_{\mathbf{(10)}}$ = **1000111**$_{\mathbf{(2)}}$ — biti **30** do **24** sadrže uvećani eksponent ***b***
- ***M*** = **1011 1111 0000 0000 0000 0000**$_{\mathbf{(2)}}$ — biti **23** do **0** sadrže cifre mantise

32-bitni zapis: **-95,5**$_{\mathbf{(10)}}$ = **1100 0111 1011 1111 0000 0000 0000 0000**$_{\mathbf{(2)}}$

Primer 40. Odrediti 32-bitni binarni zapis broja +12,75$_{(10)}$ u formatu pokretnog zareza **1-7-24**.

Rešenje:

+12,75$_{\mathbf{(10)}}$ = 1100,11$_{(2)}$ = **0,110011**$_{\mathbf{(2)}}\cdot 2^4$ predstavlja **normalizovani** eksponencijalni oblik zadatog broja, u kome je:

- ***Z*** = **0** — bit **31** je **nula**, jer je broj **pozitivan**
- ***E*** = 4 $\Rightarrow$ ***b* = *E***+64 **= 68**$_{\mathbf{(10)}}$ = **1000100**$_{\mathbf{(2)}}$ — biti **30** do **24** sadrže uvećani eksponent ***b***
- ***M*** = **1100 1100 0000 0000 0000 0000**$_{\mathbf{(2)}}$ — biti **23** do **0** sadrže cifre mantise

32-bitni zapis: **+12,75**$_{\mathbf{(10)}}$ = **0100 0100 1100 1100 0000 0000 0000 0000** $_{\mathbf{(2)}}$

Primer 41. Koji je decimalan broj zapisan u 32-bitnom registru, čiji sadržaj odgovara heksadecimalnom zapisu 47 97 00 00$_{(16)}$ u formatu pokretnog zareza **1-7-24**?

Rešenje:

+75,5$_{\mathbf{(10)}}$

Primer 42. Koji je decimalan broj zapisan u 32-bitnom registru, čiji sadržaj odgovara heksadecimalnom zapisu 47 BF 00 00$_{(16)}$ u formatu pokretnog zareza **1-7-24**?

Rešenje:

+95,5$_{\mathbf{(10)}}$

Primer 43. Koji je decimalan broj zapisan u 32-bitnom registru, čiji sadržaj odgovara heksadecimalnom zapisu C4 CC 00 00$_{(16)}$, ako se zna da je za čuvanje broja primenjen format **1-7-24**.

Rešenje:

-12,75$_{\mathbf{(10)}}$

Primer 44. Pronaći kako bi u memoriji izgledao zapis broja $–0{,}6875_{(10)}$:

a) Kao broj u pokretnom zarezu u 32-bitnom formatu **1-7-24** bita.

b) Kao broj u zapisu sa fiksnim zarezom u formatu $Q^S_{1,7}$.

c) U kojim sve formatima fiksnog zareza može da se napiše ovaj broj? Obrazložiti.

d) Napisati broj u jednom od formata iz tačke c), po sopstvenom izboru.

Rešenje:

a) $\mathbf{– 0{,}6875_{(10)}} = -0{,}1011_{(2)} = \mathbf{–0{,}1011_{(2)}\cdot 2^0}$ predstavlja **normalizovani** eksponencijalni oblik zadatog broja, u kome je:

- $\boldsymbol{Z} = \mathbf{1}$ — bit **31** je **jedan**, jer je broj **negativan**
- $\boldsymbol{E} = 0 \Rightarrow \boldsymbol{b} = \boldsymbol{0}+64 = \mathbf{64_{(10)}} = \mathbf{1000000_{(10)}}$ — biti **30** do **24** sadrže uvećani eksponent ***b***
- $\boldsymbol{M} = \mathbf{101100000000000000000000_{(2)}}$ — biti **23** do **0** sadrže cifre mantise

32-bitni zapis: $\mathbf{-0{,}6875_{(10)}} = \mathbf{1100\ 0000\ 1011\ 0000\ 0000\ 0000\ 0000\ 0000_{(2)}}$

b) $+0{,}6875_{(10)} = 0{,}1011_{(2)} = 01011000_{(2)}$ ($Q^S_{1,7}$) pa je zapis $–0{,}6875_{(10)}$ drugi komplement ovog binarnog broja, odnosno, $\mathbf{–0{,}6875_{(10)} = 10101000_{(2)}}$ ($\mathbf{Q^S_{1,7}}$)

c) $+0{,}6875_{(10)} = 0{,}1011_{(2)}$ bez gubitka tačnosti može još da se zapiše u formatima $\mathbf{Q^S_{2,6}}$, $\mathbf{Q^S_{3,5}}$ i $\mathbf{Q^S_{4,4}}$ pa su to i formati u kojima bi broj $– 0{,}6875_{(10)}$ bio zapisan bez gubitka tačnosti.

d) $\mathbf{–0{,}6875_{(10)}} = \mathbf{11010100_{(2)}}$ ($\mathbf{Q^S_{2,6}}$) $= \mathbf{11101010_{(2)}}$ ($\mathbf{Q^S_{3,5}}$) $= \mathbf{11110101_{(2)}}$ ($\mathbf{Q^S_{4,4}}$)

Tridesetdvobitni zapis po standardu **IEEE 754** sadrži sledeća polja:

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

Z	***E*+126**	**mantisa bez jedinice na mestu MSB**

- **Polje znaka (*Z*)** (bit **31**) sadrži 0 ako je broj pozitivan ili sadrži 1 ako je broj negativan.
- **Polje uvećanog eksponenta (*E*+126)** (osam bita, od bita **30** do bita **23**) sadrži eksponent ***E*** uvećan za 126.
- **Polje mantise (mantisa)** (od bita **22** do bita **0**) sadrži mantisu predstavljenu u formatu $Q_{(0,24)}$, bez jedinice na mestu MSB

Ovim izmenama je značajno proširen opseg brojeva u odnosu na standard 1-7-24, a polje mantise je idalje ostalo 24-bitno, samo se njen MSB bit ne upisuje zato što je uvek 1. Kako broj bita zapisa mantise određuje rezoluciju brojeva koji se mogu zapisati standardom, ta je rezolucija ostala ista u odnosu na prethodni standard, a opseg brojeva se udvostručio. Uporedni pregled karakteristika formata 1-7-24 i IEEE 754 prikazan je u tabeli:

	Z	Polje ***b***	Polje mantise
1-7-24	Pozitivan: **0** Negativan: **1**	Sedmobitno, višak **64**	24-bitno **mantisa** u formatu $Q_{0,24}$
IEEE754	Pozitivan: **0** Negativan: **1**	Osmobitno, višak **126**	23-bitno **mantisa_1:** mantisa u formatu $Q_{0,24}$ bez jedinice na mestu MSB

32 bitni zapis u formatu **IEEE 754** predstavlja ***float*** tip u većini implementacija C jezika.

Standard **IEEE 754** definiše i i prošireni, 64-bitni zapis koji predstavlja tip ***double*** u C-jeziku i koristi: **1** bit za znak, **11 bita** za polje ***b***, sa viškom **1024**, i **52** bita za **mantisu_1**. Mantisa je 53-bitna ali se jedinica na mestu MSB ne beleži.

- Najveći (po apsolutnoj vrednosti) broj po IEEE 754 standardu u 32-bitnom zapisu iznosi oko $\pm 3{,}4 \cdot 10^{38}$
- Najmanji (po apsolutnoj vrednosti) broj po IEEE 754 standardu u 32-bitnom zapisu iznosi oko $\pm 1{,}2 \cdot 10^{-38}$
- Najveći (po apsolutnoj vrednosti) broj po IEEE 754 standardu u 64-bitnom zapisu iznosi oko $\pm 2{,}2 \cdot 10^{308}$
- Najmanji (po apsolutnoj vrednosti) broj po IEEE 754 standardu u 64-bitnom zapisu iznosi oko $\pm 1{,}8 \cdot 10^{-308}$

Ako je broj koji se dobije kao rezultat neke operacije, po apsolutnoj vrednosti veći od najvećeg to se naziva **prekoračenje**. Postoji mogućnost i da rezultat po apsolutnoj vrednosti bude manji od najmanjeg broja koji se može prikazati u usvojenom formatu tada se to naziva **potkoračenje**.

Specijalni slučajevi brojnih vrednosti u zapisu sa pokretnim decimalnim zarezom:

- nula, kada je i polje ***b*** i ***mantisa_1*** nula;
- beskonačnost, kada polje ***b*** isadrži sve jedinice, a ***mantisa_1*** je nula;
- NaN (*not a number*) kada polje ***b*** isadrži sve jedinice, a ***mantisa_1*** je različita od nule; Ovakav rezultat se dobija, na primer, deljenjem nule nulom, ili množenjem nule i beskonačnosti, ili kao rezultat neke operacije koja nema smisla (koren ili logaritam negativnog broja).

Primer 45. Broj $13{,}25_{(10)}$ predstaviti u formatu **IEEE 754**. Kom heksadecimalnom broju odgovara binarni zapis ovog broja u navedenom formatu?

Rešenje:

$\mathbf{13{,}25_{(10)}} = 1101{,}01_{(2)} = \mathbf{0{,}110101 \cdot 2^4}$ predstavlja **normalizovani** eksponencijalni oblik zadatog broja, u kome je:

- $Z = 0$ — bit **31** je **nula**, jer je broj **pozitivan**
- $b = 4+126 = 130_{(10)} = 1000\ 0010_{(2)}$ — biti **30** do **23** sadrže **uvećani eksponent *b***
- $M = 1101\ 0100\ 0000\ 0000\ 0000\ 0000_{(2)}$ — biti **22** do **0** sadrže **cifre mantise**

32-bitni zapis: $\mathbf{13{,}25_{(10)}} = \mathbf{0100\ 0001\ 0101\ 0100\ 0000\ 0000\ 0000\ 0000_{(2)}} = \mathbf{41540000_{(16)}}$

NAPOMENA:

*U delu literature vezane za opis tehnike upisivanja brojeva u pokretnom zarezu, broj se svodi na mantisu oblika (1,....), odnosno, na opseg između 1 i 2, ($1 \leq$ mantisa < 2). Sa tako definisanom mantisom, eksponent je za 1 manji u odnosu na eksponent u sistemu normalizacije koja se koristi u ovom poglavlju. Dakle, ako se koristi mantisa oblika (1,...), eksponent ja za 1 manji i višak iznosi 127. Na primer, za broj 13,25, tako definisani eksponent bi bio 3, a mantisa bi bila $1{,}10101_2$. Naravno broj u polju **b** bi ponovo bio 3+127=130, a u polju **mantisa_1** bi bili bitovi mantise iza zareza. Binarni zapis broja je isti, sva tri polja sadrže iste binarne vrednosti, samo se pojam mantise i pojam eksponenta razlikuju u odnosu na pojmove korišćene u ovom poglavlju!*

Primer 46. Za broj $65{,}75_{(10)}$ odrediti zapis u memoriji računara ako se zna da je za čuvanje broja primenjen **IEEE 754** standard.

Rešenje:

$\mathbf{+65{,}75_{(10)}} = 1000001{,}11_{(2)} = \mathbf{0{,}100000111_{(2)}} \cdot 2^7$ predstavlja **normalizovani** eksponencijalni oblik zadatog broja, u kome je:

- $\boldsymbol{Z} = \mathbf{0}$ bit **31** je **nula**, jer je broj **pozitivan**
- $\boldsymbol{E} = 7 \Rightarrow \boldsymbol{b} = \boldsymbol{E}+126 = \mathbf{133_{(10)}} = \mathbf{10000101_{(2)}}$ biti **30** do **23** sadrže **uvećani eksponent *b***
- $\boldsymbol{M} = \mathbf{000\ 0011\ 1000\ 0000\ 0000\ 0000_{(2)}}$ biti **22** do **0** sadrže **cifre mantise**

32-bitni zapis: $\mathbf{+65{,}75_{(10)}} = \mathbf{0100\ 0010\ 1000\ 0011\ 1000\ 0000\ 0000\ 0000_{(2)}}$

Primer 47. Za broj $-61{,}6875_{(10)}$ odrediti zapis u memoriji računara ako se zna da je za čuvanje broja primenjen **IEEE 754** standard.

Rešenje:

$\mathbf{-61{,}6875_{(10)}} = -111101{,}1011_{(2)} = \mathbf{-0{,}1111011011_{(2)}} \cdot 2^6$ predstavlja **normalizovani** eksponencijalni oblik zadatog broja, u kome je:

- $\boldsymbol{Z} = \mathbf{1}$ bit 31 je **jedan**, jer je broj **negativan**
- $\boldsymbol{E} = 6 \Rightarrow \boldsymbol{b} = \boldsymbol{E}+126 = \mathbf{132_{(10)}} = \mathbf{1000\ 0100}_{(2)}$ biti **30** do **23** sadrže **uvećani eksponent *b***
- $\boldsymbol{M} = \mathbf{111\ 0110\ 1100\ 0000\ 0000\ 0000_{(2)}}$ biti **22** do **0** sadrže **cifre mantise**

32-bitni zapis: $\mathbf{-61{,}6875_{(10)}} = \mathbf{1100\ 0010\ 0111\ 0110\ 1100\ 0000\ 0000\ 0000_{(2)}}$

Primer 48. Za broj $+15{,}25_{(10)}$ odrediti zapis u memoriji računara ako se zna da je za čuvanje broja primenjen **IEEE 754** standard.

Rešenje:

$\mathbf{+15{,}25_{(10)}} = 1111{,}01_{(2)} = \mathbf{0{,}111101_{(2)}} \cdot 2^4$ predstavlja **normalizovani** eksponencijalni oblik zadatog broja, u kome je:

- $\boldsymbol{Z} = \mathbf{0}$ bit **31** je **nula**, jer je broj **pozitivan**
- $\boldsymbol{E} = 4 \Rightarrow \boldsymbol{b} = \boldsymbol{E}+126 = \mathbf{130_{(10)}} = \mathbf{1000\ 0010}_{(2)}$ biti **30** do **23** sadrže **uvećani eksponent *b***
- $\boldsymbol{M} = \mathbf{111\ 0100\ 0000\ 0000\ 0000\ 0000_{(2)}}$ biti **22** do **0** sadrže **cifre mantise**

32-bitni zapis: $\mathbf{+15{,}25_{(10)}} = \mathbf{0100\ 0001\ 0111\ 0100\ 0000\ 0000\ 0000\ 0000_{(2)}}$

Primer 49. Koji je decimalan broj zapisan u 32-bitnom registru, čiji sadržaj odgovara heksadecimalnom zapisu 3F 80 00 00, ako se zna da je za čuvanje broja primenjen **IEEE 754** standard?

Rešenje:

$\mathbf{+1{,}0_{(10)}}$

Primer 50. Koji je decimalan broj zapisan u 32-bitnom registru, čiji sadržaj odgovara heksadecimalnom zapisu $40\ 40\ 00\ 00_{(16)}$, ako se zna da je za čuvanje broja primenjen **IEEE 754** standard?

Rešenje:

+3,0$_{(10)}$

Primer 51. Koji je decimalan broj zapisan u 32-bitnom registru, čiji sadržaj odgovara heksadecimalnom zapisu 3F 00 00 00$_{(16)}$, ako se zna da je za čuvanje broja primenjen **IEEE 754** standard?

Rešenje:

+0,5$_{(10)}$

Primer 52. Koji je decimalan broj zapisan u 32-bitnom registru, čiji sadržaj odgovara heksadecimalnom zapisu C0 40 00 00$_{(16)}$, ako se zna da je za čuvanje broja primenjen **IEEE 754** standard?

Rešenje:

-3$_{(10)}$

Primer 53. Koji je decimalan broj zapisan u 32-bitnom registru, čiji sadržaj odgovara heksadecimalnom zapisu C2 C8 00 00$_{(16)}$, ako se zna da je za čuvanje broja primenjen **IEEE 754** standard?

Rešenje:

-100$_{(10)}$

Primer 54. Koji je decimalan broj zapisan u 32-bitnom registru prikazanom na slici, ako se zna da je za čuvanje broja primenjen **IEEE 754** standard?

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

Rešenje:

+9,6875$_{(10)}$

Primer 55. Koji je decimalan broj zapisan u 32-bitnom registru prikazanom na slici, ako se zna da je za čuvanje broja primenjen **IEEE 754** standard?

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

Rešenje:

+10$_{(10)}$

Primer 56. Koji je decimalan broj zapisan u 32-bitnom registru prikazanom na slici, ako se zna da je za čuvanje broja primenjen **IEEE 754** standard?

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

Rešenje:

+2,0$_{(10)}$

Primer 57. Koji je decimalan broj zapisan u 32-bitnom registru prikazanom na slici, ako se zna da je za čuvanje broja primenjen **IEEE 754** standard?

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

Rešenje:

+3,0$_{(10)}$

Primer 58. Kako bi u memoriji izgledao zapis broja – 263,6875$_{(10)}$:

a) Kao broj u pokretnom zarezu u 32-bitnom formatu **1-7-24** bita.

b) Opisati po čemu bi se razlikovao zapis ovog broja po **IEEE 754** standardu.

c) U kojim sve formatima fiksnog zareza može da se napiše ovaj broj? Obrazložiti.

d) Napisati broj u binarnom obliku u jednom od formata iz tačke c), po sopstvenom izboru.

Rešenje:

a) **–263,6875$_{(10)}$** = –100000111,1011$_{(2)}$ = **–0,1000001111011·2^9** predstavlja **normalizovani** eksponencijalni oblik zadatog broja, u kome je:

- ***Z*** = 1 — bit **31** je **jedan**, jer je broj **negativan**
- ***E*** = 9 ⇒***b*** **=** ***E***+64 **= 73$_{(10)}$ = 1001001$_{(2)}$** — biti **30** do **24** sadrže **uvećani eksponent *b***
- ***M*** **= 1000001111011**00000000000$_{(2)}$ (Q$_{0,24}$) — biti **23** do **0** sadrže **cifre mantise**

32-bitni zapis: **–263,6875$_{(10)}$** = **1100 1001 1000 0011 1101 1000 0000 0000$_{(2)}$**

b) Mantisa ***M*** ne bi imala jedinicu na mestu MSB (**mantisa_1** = **000 0011 1101 1000 0000 0000$_{(2)}$**), dok je ***b*** = **9**+**126**= **135$_{(10)}$** = **1000 0111$_{(2)}$.**

IEEE 754 zapis bi bio: **1100 0011 1000 0011 1101 1000 0000 0000$_{(2)}$**

c) +263,6875$_{(10)}$ = 0100000111,1011$_{(2)}$ bez gubitka tačnosti može da se zapiše u formatima **Q$^S_{12,4}$**, **Q$^S_{11,5}$**, i **Q$^S_{10,6}$** pa su to i formati u kojima bi broj – 263,6875$_{(10)}$ bio zapisan bez gubitka tačnosti.

d) **–263,6875$_{(10)}$** = **1110111110000101$_{(2)}$ (Q$^S_{12,4}$) = 1101111100001010$_{(2)}$ (Q$^S_{11,5}$) ...**

Primer 59. Kako bi u memoriji izgledao zapis broja –519,6875$_{(10)}$:

a) Kao broj u pokretnom zarezu u 32-bitnom formatu **1-7-24**.

b) Opisati po čemu bi se razlikovao zapis ovog broja po **IEEE 754** standardu

c) U kojim sve formatima fiksnog zareza može da se napiše ovaj broj bez gubitka tačnosti zapisa? Obrazložiti.

d) Napisati broj (binarni oblik) u jednom od formata iz tačke c) , po sopstvenom izboru.

Rešenje:

a) **–519,6875$_{(10)}$** = –110100011,1011$_{(2)}$ = **–0,1101000111011·2^9** predstavlja **normalizovani** eksponencijalni oblik zadatog broja, u kome je:

- ***Z*** = 1 — bit **31** je **jedan**, jer je broj **negativan**
- ***E*** = 9 ⇒***b*** **=** ***E*****+64 = 73$_{(10)}$ = 1001001$_{(2)}$** — biti **30** do **24** sadrže **uvećani eksponent *b***
- ***M*** **= 1101000111011**00000000000**$_{(2)}$** ($Q_{0,24}$) — biti **23** do **0** sadrže **cifre mantise**

32-bitni zapis: **– 519,6875$_{(10)}$** = **1100 1001 1101 0001 1101 1000 0000 0000$_{(2)}$**

b) Mantisa ***M*** ne bi imala jedinicu na mestu MSB (**mantisa_1**=**100 0001 1101 1000 0000 0000$_{(2)}$**), dok je ***b*** = **9+126** = **135$_{(10)}$** = **1000 0111$_{(2)}$,**

IEEE 754 zapis bi bio: **1100 0011 1101 0001 1101 1000 0000 0000$_{(2)}$**

c) +519,6875$_{(10)}$ = 110100011,1011$_{(2)}$ bez gubitka tačnosti može da se zapiše u formatima **$Q^S_{12,4}$**, **$Q^S_{11,5}$** i **$Q^S_{10,6}$** pa su to i formati u kojima bi broj – 519,6875$_{(10)}$ bio zapisan bez gubitka tačnosti.

d) **–519,6875$_{(10)}$** = **1110 0101 1100 0101$_{(2)}$ ($Q^S_{12,4}$) = 1100 1011 1000 1010$_{(2)}$ ($Q^S_{11,5}$)**

Primer 60. Zapis razlomljenog broja po **1-7-24** standardu (sa viškom 64) je BEC0 0000$_{(16)}$

a) O kom decimalnom broju se radi?

b) Predstaviti ovaj broj po **IEEE 754** standardu.

c) Šta određuje broj bita u polju mantise, a šta broj bita u polju b (za uvećani eksponent)?

Rešenje:

a) BEC0 0000$_{(16)}$ = **1011 1110 1100 0000 0000 0000 0000 0000$_{(2)}$**

Polje znaka je **1**, pa je broj negativan.

Polje ***b*** je **0111110$_{(2)}$** = 62$_{(10)}$ pa, pošto je ***b*** = ***E***+64, eksponent iznosi ***E*** **= –2**.

Mantisa u formatu $Q_{0,24}$ iznosi **1100 0000 0000 0000 0000 0000$_{(2)}$**, odnosno, mantisa je ***M*** = **0,11$_{(2)}$**

Traženi broj je |*x*| = **0,11$_{(2)}$·2^{-2}** = 0,75/4 = 0,1875$_{(10)}$, odnosno, ***x*** **= –0,1875$_{(10)}$**

b) Mantisa ne bi imala jedinicu na mestu MSB, **mantisa_1** =**100 0000 0000 0000 0000 0000$_{(2)}$**

Polje ***b*** bi bilo: ***b*** **=** ***E***+126= 124$_{(10)}$ = **0111 1100$_{(2)}$.**

IEEE 754 zapis bi bio: **1011 1110 0100 0000 0000 0000 0000 0000$_2$= BE40 0000$_{(16)}$**

c) Broj bita u polju mantise određuje tačnost sa kojom se brojevi mogu zapisati, odnosno, minimalni korak između dva susedna broja koji se mogu zapisati.

Veličina polja za eksponent (uz standardom usvojenu vrednost viška) definiše opseg eksponenata. Minimalna vrednost eksponenta određuje najmanji broj različit od nule, a maksimalni eksponent određuje najveći boj koji se može upisati po tom standardu.

3.3. Predstavljanje brojeva tehnikom binarnog kodovanja

Binarnim kodovanjem cifara decimalnog brojnog sistema (BCD) svaka cifra decimalnog brojnog sistema se zamenjuje ekvivalentnom binarnom vrednošću koja se sastoji od četiri binarne cifre. Najčešće se primenjuju kodovi **8421**, kod "**više 3**" i **Grejov** kod.

Kod 8421 je težinski kod koji se dobija direktnom konverzijom vrednosti svake cifre iz decimalnog brojnog sistema ekvivalentnom vrednošću u binarnom brojnom sistemu.

Kod "više 3" nije težinski kod. Ovaj kod ima osobinu autokomplementiranja koju ne poseduje kod 8421. Na primer, prvi komplement koda "više 3" za cifru 0 je 1100, što predstavlja istovremeno kod "više 3" za cifru 9, dok prvi komplement koda "više 3" za cifru 9 ima istu vrednost kao kod "više 3" za cifru 0, odnosno 0011.

Grejov kod nije težinski kod. Dva susedna binarna koda u Gray-ovom kodu razlikuju se u vrednosti samo jednog bita koda.

Decimalna cifra	**8421** BCD kod	**"više 3"** BCD kod	**Grejov** BCD kod
0	0000	0011	0000
1	0001	0100	0001
2	0010	0101	0011
3	0011	0110	0010
4	0100	0111	0110
5	0101	1000	0111
6	0110	1001	0101
7	0111	1010	0100
8	1000	1011	1100
9	1001	1100	1000

Binarnim kodovanjem cifara decimalnog brojnog sistema, obezbeđuje se tačno prikazivanje racionalnih decimalnih brojeva, bez greške usled zokruživanja koja nastaje pri klasičnoj konverziji brojnih vrednosti iz decimalnog u binarni brojni sistem.

Primer 61. Prikazati decimalan broj $15{,}147_{(10)}$ u kodu 8421, kodu "više 3", i Grejovom kodu.

Rešenje:

Svaka cifra zadatog decimalnog broja se koduje sa četiri binarne cifre odgovarajućeg BCD koda:

$15{,}147_{(10)} = 0001\ 0101{,}0001\ 0100\ 0111_{(8421)}$

$15{,}147_{(10)} = 0100\ 1000{,}0100\ 0111\ 1010_{(\text{više 3})}$

$15{,}147_{(10)} = 0001\ 0111{,}0001\ 0110\ 0100_{(\text{Grej})}$

Primer 62. Brojeve predstavljene u BCD kodovima predstaviti u decimalnom brojnom sistemu.

a) $100100{,}00111_{(8421)}$

b) $111011{,}01111_{(\text{“ više 3 ”})}$

c) $1100{,}100011_{(\text{Grej})}$

Rešenje:

Treba odvojiti po četiri binarne cifre levo i desno od decimalnog zareza. Ukoliko je neophodno, četiri bita na krajnjoj levoj i desnoj poziciji se formiraju dodavanjem nula.

a) $100100{,}00111_{(8421)} = 0010\ 0100{,}0011\ 1000_{(8421)} = \mathbf{24{,}38_{(10)}}$

b) $111011{,}01111_{(\text{„ više 3 ”})} = 0011\ 1011,\ 0111\ 1000_{(\text{„ više 3 ”})} = \mathbf{8{,}45_{(10)}}$

c) $10101{,}0100011_{(\text{Grej})} = 0001\ 0101{,}0100\ 0110_{(\text{Gray})} = \mathbf{6{,}74_{(10)}}$

<u>Primer 63.</u> U memoriji šesnaestobitnog računara nalazi se podatak $8C38_{(16)}$. O kom decimalnom broju je reč ako se zna da je zapisan kao:

a) Neoznačen ceo broj

b) Označen ceo broj

c) Četvorocifreni broj u Grejovom kodu (viši bajt je prvi)

d) Neoznačeni broj u formatu $Q_{12,4}$

e) Dva označena osmobitna broja u formatu $Q_{4,4}$

f) Dva neoznačena osmobitna broja u formatu $Q_{2,6}$

Rešenje:

a) $2^{15} + 2^{11} + 2^{10} + 2^5 + 2^4 + 2^3 = \mathbf{35896_{(10)}}$

b) $-2^{15} + 2^{11} + 2^{10} + 2^5 + 2^4 + 2^3 = \mathbf{-29640_{(10)}}$

c) $\mathbf{9829_{(10)}}$

d) *Prvi način*:

U neoznačenom formatu $Q_{12,4}$ dvanaest bita se nalazi ispred, a četiri bita iza zareza. Težinski koeficijenti za svaki bit rastu ulevo od zareza od 2^0 do 2^{11}, a desno od zareza rastu u negativnu stranu od 2^{-1} do 2^{-4}.

$$8C38_{(16)}\ (Q_{12,4}) = \underset{2^{11}}{1}\,0\,0\,0\ \ \underset{2^7}{1}\,\underset{2^6}{1}\,0\,0\ \ 0\,0\,\underset{2^1}{1}\,\underset{2^0}{1},\ \underset{2^{-1}}{1}\,0\,0\,0_{\ (2)}$$

Traženi decimalni broj se dobija sabiranjem ovih težinskih koeficijenata:

$8C38_{(16)} = 2^{11} + 2^7 + 2^6 + 2^1 + 2^0 + 2^{-1} = \mathbf{2243{,}5_{(10)}}$

Drugi način:

$8C38_{(16)}$ se direktno pretvara u decimalni kôd, a dobijeni rezultat treba da se podeli sa 2^4 (odnosno da se pomnoži sa $1/2^4$):

$8C38_{16)} = 2^{-4}(2^{15} + 2^{11} + 2^{10} + 2^5 + 2^4 + 2^3) = 35896/16 = \mathbf{2243{,}5_{(10)}}$

Da se podsetimo:
Celobrojna vrednost u drugom načinu konverzije se u opštem slučaju deli sa 2^R, gde je R broj bita iza zareza, odnosno broj bita u razlomljenom delu (u prethodnom primeru R=4).

e) *Prvi način*:

$8C_{(16)}\ (Q^S_{4,4}) = 1000{,}1100_{(2)} = -2^3 + 2^{-1} + 2^{-2} = \mathbf{-7{,}25_{(10)}}$

$38_{(16)}\ (Q^S_{4,4}) = 0011,\ 1100_{(2)} = 2^1 + 2^0 + 2^{-1} + 2^{-2} = \mathbf{3{,}5_{(10)}}$

Drugi način:

$8C_{(16)} = 2^{-4}(-2^7 + 2^3 + 2^2) = -116/16 = \mathbf{-7{,}25_{(10)}}$

$38_{(16)} = 2^{-4}(2^5 + 2^4 + 2^3) = 56/16 = \mathbf{3{,}5_{(10)}}$

f) *Prvi način*:

$8C_{(16)}\ (Q_{2,6}) = 10{,}001100_{(2)} = 2^1 + 2^{-3} + 2^{-4} = \mathbf{2{,}1875_{(10)}}$

$38_{(16)}\ (Q_{2,6}) = 00{,}111100_{(2)} = 2^{-1} + 2^{-2} + 2^{-3} + 2^{-4} = \mathbf{0{,}875_{(10)}}$

Drugi način:

$8C_{(16)} = 2^{-6}(2^7+2^3+2^2) = 140/64 = \mathbf{2{,}1875_{(10)}}$

$38_{(16)} = 2^{-6}(2^5+2^4+2^3) = 56/64 = \mathbf{0{,}875_{(10)}}$

Primer 64. U memoriji šesnaestobitnog računara nalazi se podatak $9840_{(16)}$. O kom decimalnom broju je reč ako se zna da je zapisan kao:

a) Neoznačen ceo broj

b) Označen ceo broj

c) Četvorocifreni broj u kodu **8421** (viši bajt je prvi)

d) Neoznačeni broj u formatu $Q_{12,4}$

e) Dva označena osmobitna broja u formatu $Q_{4,4}$

f) Dva neoznačena osmobitna broja u formatu $Q_{3,5}$

Rešenje:

a) $\mathbf{38976_{(10)}}$

b) $\mathbf{-26560_{(10)}}$

c) $\mathbf{9840_{(10)}}$

d) $\mathbf{2436{,}0_{(10)}}$

e) $\mathbf{-6{,}5_{(10)}}$ i $\mathbf{4{,}0_{(10)}}$

f) $\mathbf{4{,}75_{(10)}}$ i $\mathbf{2{,}0_{(10)}}$

Primer 65. U dve uzastopne lokacije osmobitnog računara nalaze se podaci $01101001_{(2)}$ i $10001010_{(2)}$, pri čemu je prvi podatak na nižoj adresi. O kojim decimalnim brojevima je reč ako se zna da su zapisani kao:

a) Dva neoznačena binarna broja

b) Dva označena binarna broja

c) Dvocifreni broj u kodu **8421** (u podatku na nižoj adresi) i dvocifreni broj u kodu **"više 3"** (u podatku na višoj adresi)

d) Neoznačeni brojevi u formatu $Q_{3,5}$

e) Neoznačeni brojevi u formatu $Q_{7,1}$

Rešenje:

a) $\mathbf{105_{(10)}}$ i $\mathbf{138_{(10)}}$

b) $\mathbf{105_{(10)}}$ i $\mathbf{-118_{(10)}}$

c) $\mathbf{69_{(10)}}$ i $\mathbf{57_{(10)}}$

d) $\mathbf{3{,}28125_{(10)}}$ i $\mathbf{4{,}3125_{(10)}}$

e) $\mathbf{52{,}5_{(10)}}$ i $\mathbf{69{,}0_{(10)}}$

Primer 66. U dve uzastopne lokacije osmobitnog računara nalaze se podaci $10100011_{(2)}$ i $00111010_{(2)}$, pri čemu je prvi podatak na nižoj adresi. O kojim decimalnim brojevima je reč ako se zna da su zapisani kao:

a) Dva neoznačena binarna broja

b) Dva označena binarna broja

c) Jedan četvorocifreni broj u kodu **"više 3"** (viši bajt je na nižoj adresi)

d) Neoznačeni brojevi u formatu $Q_{4,4}$

e) Neoznačeni brojevi u formatu $Q_{7,1}$

Rešenje:

a) $\mathbf{163_{(10)}}$ i $\mathbf{58_{(10)}}$

b) $\mathbf{-93_{(10)}}$ i $\mathbf{58_{(10)}}$

c) $\mathbf{7007_{(10)}}$

d) $\mathbf{10{,}1875_{(10)}}$ i $\mathbf{3{,}625_{(10)}}$

e) $\mathbf{81{,}5_{(10)}}$ i $\mathbf{29{,}0_{(10)}}$

---

4. SABIRANJE BROJEVA KOJI SU PREDSTAVLJENI KODOVIMA 8421 I "VIŠE 3"

Sabiranje brojeva koji su predstavljeni kodovima 8421 i "više 3" ne može da se izvrši direktnom primenom pravila binarne aritmetike, već je potrebna korekcija.

Koraci sabiranja u kodu 8421:

Prvi korak**:**

- BCD brojevi se sabiraju bit po bit prema pravilima binarne aritmetike.

Drugi korak:

- Ako nema prenosa u sledeću tetradu i ako je broj $\leq 1001_{(2)}$ nema korekcije.
- Ako je broj $> 1001_{(2)}$ i nema prenosa u sledeću tetradu, dodaje se $6_{(10)}$(**$0110_{(2)}$**).
- Ako sa neke tetrade postoji prenos u sledeću, toj tetradi se dodaje $6_{(10)}$ (**$0110_{(2)}$**).

Koraci sabiranja u kodu "više 3":

Prvi korak:

- BCD brojevi se sabiraju bit po bit prema pravilima binarne aritmetike.

Drugi korak:

- Ako sa neke tetrade postoji prenos u sledeću, toj tetradi se dodaje $3_{(10)}$ (**$0011_{(2)}$**).
- Ako nema prenosa u sledeću tetradu od te tetrade se oduzima $3_{(10)}$ (tetradi se dodaje drugi komplement broja $-3_{(10)}$, koji u binarnom obliku iznosi **$1101_{(2)}$**).

Primer 1. Prikazati u kodu **8421** brojeve $6187_{(10)}$ i $2495_{(10)}$, a zatim izračunati njihov zbir i vratiti rezultat u decimalan brojni sistem.

Rešenje:

$$
\begin{array}{lrl}
6187_{(10)} & & 0110\ 0001\ 1000\ 0111_{(8421)} \\
2495_{(10)} & + & \underline{0010\ 0100\ 1001\ 0101_{(8421)}} \\
 & & 1000\ 0110\ 0001\ 1100 \\
 & + & \underline{0110\ 0110} \quad \text{1. korekcija} \\
 & & \mathbf{1000\ 0110\ 1000\ 0010 = 8682_{(10)}}
\end{array}
$$

Primer 2. Prikazati u kodu **8421** brojeve $7531_{(10)}$ i $1484_{(10)}$, a zatim izračunati njihov zbir i vratiti rezultat u decimalan brojni sistem.

Rešenje:

$$
\begin{array}{lrl}
7531_{(10)} & & 0111\ 0101\ 0011\ 0001_{(8421)} \\
1484_{(10)} & + & \underline{0001\ 0100\ 1000\ 0100_{(8421)}} \\
 & & 1000\ 1001\ 1011\ 0101 \\
 & + & \underline{0110} \quad \text{1. korekcija} \\
 & & 1000\ 1010\ 0001\ 0101 \\
 & + & \underline{0110} \quad \text{2. korekcija} \\
 & & \mathbf{1001\ 0000\ 0001\ 0101_{(8421)}} = 9015_{(10)}
\end{array}
$$

Primer 3. Prikazati u kodu **8421** brojeve $6187_{(10)}$ i $2495_{(10)}$, a zatim izračunati njihov zbir i vratiti rezultat u decimalan brojni sistem.

Rešenje:

$$
\begin{array}{llr}
6187_{(10)} & & 0110\ 0001\ 1000\ 0111_{(8421)} \\
2495_{(10)} & + & \underline{0010\ 0100\ 1001\ 0101}_{(8421)} \\
 & & 1000\ 0110\ 0001\ 1100 \\
 & + & \underline{\ 0110\ 0110} \ \text{1. korekcija} \\
 & & \mathbf{1000\ 0110\ 1000\ 0010 = 8682_{(10)}}
\end{array}
$$

Primer 4. Prikazati u kodu **8421** brojeve $7531_{(10)}$ i $1484_{(10)}$, a zatim izračunati njihov zbir i vratiti rezultat u decimalan brojni sistem.

Rešenje:

$$
\begin{array}{llr}
7531_{(10)} & & 0111\ 0101\ 0011\ 0001_{(8421)} \\
1484_{(10)} & + & \underline{0001\ 0100\ 1000\ 0100}_{(8421)} \\
 & & 1000\ 1001\ 1011\ 0101 \\
 & + & \underline{\ 0110\ } \ \text{1. korekcija} \\
 & & 1000\ 1010\ 0001\ 0101 \\
 & + & \underline{\ 0110\ } \ \text{2. korekcija} \\
 & & \mathbf{1001\ 0000\ 0001\ 0101_{(8421)} = 9015_{(10)}}
\end{array}
$$

Primer 5. Prikazati u kodu **8421** brojeve $5324_{(10)}$ i $1768_{(10)}$, a zatim izračunati njihov zbir i vratiti rezultat u decimalan brojni sistem.

Rešenje:

$$
\begin{array}{llr}
5324_{(10)} & & 0101\ 0011\ 0010\ 0100_{(8421)} \\
1768_{(10)} & + & \underline{0001\ 0111\ 0110\ 1000}_{(8421)} \\
 & & 0110\ 1010\ 1000\ 1100 \\
 & + & \underline{\ 0110\ \ 0110} \ \text{korekcija} \\
 & & \mathbf{0111\ 0000\ 1001\ 0010_{(8421)} = 7092_{(10)}}
\end{array}
$$

Primer 6. Prikazati u kodu **8421** brojeve $3712_{(10)}$ i $1456_{(10)}$, a zatim izračunati njihov zbir i vratiti rezultat u decimalan brojni sistem.

Rešenje:

$$
\begin{array}{llr}
3712_{(10)} & & 0011\ 0111\ 0001\ 0010_{(8421)} \\
1456_{(10)} & + & \underline{0001\ 0100\ 0101\ 0110}_{(8421)} \\
 & & 0100\ 1011\ 0110\ 1000 \\
 & + & \underline{\ 0110\ } \ \text{korekcija} \\
 & & \mathbf{0101\ 0001\ 0110\ 1000_{(8421)} = 5168_{(10)}}.
\end{array}
$$

Primer 7. Prikazati u kodu **8421** brojeve $1357_{(10)}$ i $5468_{(10)}$, a zatim izračunati njihov zbir i vratiti rezultat u decimalan brojni sistem.

Rešenje:

$$
\begin{array}{llr}
1357_{(10)} & & 0001\ 0011\ 0101\ 0111_{(8421)} \\
5468_{(10)} & + & \underline{0101\ 0100\ 0110\ 1000}_{(8421)} \\
 & & 0110\ 0111\ 1011\ 1111 \\
 & + & \underline{\ 0110\ 0110} \ \text{korekcija} \\
 & & \mathbf{0110\ 1000\ 0010\ 0101_{(8421)} = 6825_{(10)}}.
\end{array}
$$

Primer 8. Prikazati u kodu **8421** brojeve $2875_{(10)}$ i $6943_{(10)}$, a zatim izračunati njihov zbir i vratiti rezultat u decimalan brojni sistem.

Rešenje:

```
2875(10)     0010 1000 0111 0101(8421)
6943(10)   + 0110 1001 0100 0011(8421)
             1001 0001 1011 1000
           +      0110 0110              korekcija
             1001 1000 0001 1000(8421) = 9818(10).
```

Primer 9. Prikazati u kodu **8421** brojeve $1337_{(10)}$ i $2468_{(10)}$, a zatim izračunati njihov zbir i vratiti rezultat u decimalan brojni sistem.

Rešenje:

```
1337(10)     0001 0011 0011 0111(8421)
2468(10)   + 0010 0100 0110 1000(8421)
             0011 0111 1001 1111
           +                0110         1. korekcija
             0011 0111 1010 0101
           +           0110              2. korekcija
             0011 1000 0000 0101(8421) = 3805(10).
```

Primer 10. Prikazati u kodu **8421** brojeve $2075_{(10)}$ i $5943_{(10)}$, a zatim izračunati njihov zbir i vratiti rezultat u decimalan brojni sistem.

Rešenje:

```
2075(10)     0010 0000 0111 0101(8421)
5943(10)   + 0101 1001 0100 0011(8421)
             0111 1001 1011 1000
           +           0110              1. korekcija
             0111 1010  0001 1000
           +      0110                   2. korekcija
             1000 0000  0001 1000(8421) = 8018(10).
```

Primer 11. Prikazati u kodu **8421** brojeve $2369_{(10)}$ i $1653_{(10)}$, a zatim izračunati njihov zbir i vratiti rezultat u decimalan brojni sistem.

Rešenje:

```
2369(10)     0010 0011 0110 1001(8421)
1653(10)    +0001 0110 0101 0011(8421)
             0011 1001 1011 1100
           +           0110 0110         1. korekcija
             0011 1010 0010 0010
           +      0110                   2. korekcija
             0100 0000 0010 0010(8421) = 4012(10).
```

Primer 12. Prikazati u kodu **"više 3"** brojeve $8153_{(10)}$ i $1298_{(10)}$, a zatim izračunati njihov zbir i vratiti rezultat u decimalni brojni sistem.

Rešenje:

$8153_{(10)}$ = 1011 0100 1000 0110$_{(\text{“ više 3 ”})}$
$1298_{(10)}$ = + 0100 0101 1100 1011$_{(\text{“ više 3 ”})}$
1111 1010 0101 0001
+ 1101 1101 0011 0011 korekcija
1100 0111 1000 0100$_{(\text{" više 3"})}$ **=** $\mathbf{9451_{(10)}}$

Primer 13. Prikazati u kodu **"više 3"** brojeve $5947_{(10)}$ i $3106_{(10)}$, a zatim izračunati njihov zbir i vratiti rezultat u decimalni brojni sistem.

Rešenje:

$5947_{(10)}$ 1000 1100 0111 1010$_{(\text{“ više 3 ”})}$
$3106_{(10)}$ + 0110 0100 0011 1001$_{(\text{“ više 3 ”})}$
1111 0000 1011 0011
+ 1101 0011 1011 0011 korekcija
1100 0011 1000 0110$_{(\text{" više 3"})}$ **=** $\mathbf{9053_{(10)}}$

Primer 14. Prikazati u kodu **"više 3"** brojeve $2875_{(10)}$ i $6943_{(10)}$, a zatim izračunati njihov zbir i vratiti rezultat u decimalni brojni sistem.

Rešenje:

$2875_{(10)}$ 0101 1011 10101000 $_{(\text{“ više 3 ”})}$
$6943_{(10)}$ + 1001 1100 0111 0110 $_{(\text{“ više 3 ”})}$
1111 1000 0001 1110
+ 1101 0011 0011 1101 korekcija
1100 1011 0100 1011$_{(\text{"više 3"})}$ **= 9818** $_{\mathbf{(10)}}$.

Primer 15. Prikazati u kodu **"više 3"** brojeve $5324_{(10)}$ i $1768_{(10)}$, a zatim izračunati njihov zbir i vratiti rezultat u decimalni brojni sistem.

Rešenje:

$5324_{(10)}$ 1000 0110 0101 0111 $_{(\text{“ više 3 ”})}$
$1768_{(10)}$ + 0100 1010 1001 1011 $_{(\text{“ više 3 ”})}$
1101 0000 1111 0010
+ 1101 0011 1101 0011 korekcija
1010 0011 1100 0101$_{(\text{"više 3"})}$ **=** $\mathbf{7092_{(10)}}$.

Primer 16. Prikazati u kodu **"više 3"** brojeve $3712_{(10)}$ i $1456_{(10)}$, a zatim izračunati njihov zbir i vratiti rezultat u decimalan brojni sistem.

Rešenje:

$3712_{(10)}$ 0110 1010 0100 0101 $_{(\text{“ više 3 ”})}$
$1456_{(10)}$ + 0100 0111 1000 1001 $_{(\text{“ više 3 ”})}$
1011 0001 1100 1110
+ 1101 0011 1101 1101 korekcija
1000 0100 1001 1011$_{(\text{"više 3"})}$ **= 5168** $_{\mathbf{(10)}}$.

Primer 17. Prikazati u kodu **"više 3"** brojeve $5318_{(10)}$ i $2471_{(10)}$, a zatim izračunati njihov zbir i vratiti rezultat u decimalan brojni sistem.

Rešenje:

$$
\begin{array}{lrl}
5318_{(10)} & & 1000\ 0110\ 0100\ 1011\ _{(\text{“ više 3 ”})} \\
2471_{(10)} & + & \underline{0101\ 0111\ 1010\ 0100}\ _{(\text{“ više 3 ”})} \\
 & & 1101\ 1101\ 1110\ 1111 \\
 & + & \underline{1101\ 1101\ 1101\ 1101}\ \text{korekcija} \\
 & & \mathbf{1010\ 1010\ 1011\ 1100}_{(\text{" više 3"})} = \mathbf{7789}_{(10)}.
\end{array}
$$

Primer 18. Prikazati navedene brojeve u kodu:

a) **8421**
b) **"više 3"**
a zatim izračunati njihov zbir i vratiti rezultat u decimalan brojni sistem:

1) $1111_{(10)} + 2567_{(10)}$	4) $4545_{(10)} + 4444_{(10)}$	7) $1234_{(10)} + 8765_{(10)}$
2) $2345_{(10)} + 2876_{(10)}$	5) $2323_{(10)} + 7654_{(10)}$	8) $2033_{(10)} + 6754_{(10)}$
3) $3987_{(10)} + 5555_{(10)}$	6) $6789_{(10)} + 1459_{(10)}$	9) $1099_{(10)} + 8188_{(10)}$.

Primer 19. Prikazati navedene brojeve u kodu:

a) **8421**
b) **"više 3"**
a zatim izračunati njihov zbir i vratiti rezultat u decimalan brojni sistem:

1) $3333_{(10)} + 4343_{(10)}$	4) $465_{(10)} + 6235_{(10)}$	7) $919_{(10)} + 135_{(10)}$
2) $1234_{(10)} + 3629_{(10)}$	5) $218_{(10)} + 1743_{(10)}$	8) $371_{(10)} + 817_{(10)}$
3) $1756_{(10)} + 3013_{(10)}$	6) $572_{(10)} + 9006_{(10)}$	9) $666_{(10)} + 444_{(10)}$

5. ANALIZA I SINTEZA LOGIČKIH FUNKCIJA

Aksiome i teoreme Bulove algebre

Neka je dat skup **S** = {**x, y, z,** ...} koji sadrži najmanje dva različita elementa, i neka su na ovom skupu definisana dva binarna operanda: **+** (logičko sabiranje, ILI**)** i **·** (logičko množenje, I), i jedan unarni operand $\bar{\ }$ (negacija, NE)**.**

Bulova algebra sadrži dva specijalna elementa **0** i **1**, takva da sve promenljive **x, y, z, ...** uzimaju vrednost iz skupa {**0, 1**}. Da bi ovaj skup **S,** i operacije +, · i $\bar{\ }$ sačinjavali Bulovu algebru, neophodno je da budu zadovoljene **aksiome Hantingtona**:

A-1: Binarne operacije **+** i **·** su komutativne na skupu **S,** i međusobno su distributivne tako da za svako **x**, **y**, **z**, koji pripadaju skupu **S**, važi:

$$x + y = y + x \qquad\qquad x \cdot y = y \cdot x$$

A-2: Binarne operacije **+** i na skupu **S** poseduju neutralne elemente **1** i **0,** tako da za svako **x** koje pripada skupu **S**, postoje elementi **1** i **0**, koji takođe pripadaju skupu **S**, tako da važi:

$$x + 0 = 0 + x = x \qquad\qquad x \cdot 1 = 1 \cdot x = x.$$

A-3: Na skupu **S**, za svako **x** koje pripada skupu **S**, postoji jedinstven inverzni element $\bar{x}$, koji takođe pripada skupu **S**, takav da je :

$$x + \bar{x} = 1 \qquad\qquad x \cdot \bar{x} = 0.$$

Osim aksioma, primenjuju se i teoreme, koje mogu da se dokažu:

T-1 Teorema idempotentnosti: $x + x = x$, $x \cdot x = x.$

T-2 Teorema o nultim elementima: $x + 1 = 1$, $x \cdot 0 = 0.$

T-3 Teorema o involuciji: $\bar{\bar{x}} = x$

T-4 Teorema o apsorpciji: $x + x \cdot y = x$, $x \cdot (x + y) = x.$

T-5 Teorema o asocijativnosti: $x + (y + z) = (x + y) + z$

T-6 De-Morganove teoreme: $\overline{(x + y)} = \bar{x} \cdot \bar{y}$, $\overline{(x \cdot y)} = \bar{x} + \bar{y}.$

Primer 1. Dokazati identitet: $(A + B)\cdot(A + C) = A + B \cdot C$

Rešenje:

$$\begin{aligned}(A+B)\cdot(A+C) &= A\cdot A + A\cdot C + A\cdot B + B\cdot C = A + A\cdot C + A\cdot B + B\cdot C = \\ &= A\cdot(1+C) + A\cdot B + B\cdot C = A + A\cdot B + B\cdot C = \\ &= A\cdot(1+B) + B\cdot C \\ &= A + B\cdot C\end{aligned}$$

Primer 2. Dokazati identitet: $(A+C)\cdot(A+D)\cdot(B+C)\cdot(B+D)=A\cdot B+C\cdot D$

Rešenje:

Na osnovu identiteta iz Primera 1, može da se izvrši direktno uprošćavanje logičkih proizvoda:

$$\begin{aligned}(A+C)\cdot(A+D)\cdot(B+C)\cdot(B+D) &= (A+C\cdot D)\cdot(B+C\cdot D)\\ &= A\cdot B+C\cdot D\cdot(A+B+1)\\ &= A\cdot B+C\cdot D\end{aligned}$$

Primer 3. Dokazati identitet: $A+\overline{A}\cdot B=A+B$

Rešenje:

$$\begin{aligned}A+\overline{A}\cdot B = A\cdot(1+\overline{A}+B)+\overline{A}\cdot B = A+A\cdot\overline{A}+A\cdot B+\overline{A}\cdot B &=\\ = A\cdot A+A\cdot\overline{A}+A\cdot B+\overline{A}\cdot B &=\\ = A\cdot(A+B)+\overline{A}\cdot(A+B) &=\\ = (A+B)\cdot(A+\overline{A}) &= A+B\end{aligned}$$

Primer 4. Dokazati identitet: $A+B+\overline{A}\cdot\overline{B}=1$

Rešenje:

$$A+B+\overline{A}\cdot\overline{B}=(A+B)+(\overline{A+B})=1$$

Primer 5. Dokazati identitet: $(A+B)\cdot(A+\overline{B})=A$

Rešenje:

$$\begin{aligned}(A+B)\cdot(A+\overline{B}) &= A\cdot A+A\cdot\overline{B}+A\cdot B+B\cdot\overline{B}\\ &= A+A\cdot(B+\overline{B})=A+A=A\end{aligned}$$

Primer 6. Uprostiti zadatu logičku funkciju:

$$F(A,B,C)=A\cdot\overline{B}\cdot\overline{C}+A\cdot\overline{B}\cdot C+A\cdot B\cdot\overline{C}+A\cdot B\cdot C$$

Rešenje:

$$F(A,B,C)=A\cdot\overline{C}\cdot(B+\overline{B})+A\cdot C\cdot(B+\overline{B})=A\cdot\overline{C}+A\cdot C=A\cdot(C+\overline{C})=A$$

Primer 7. Uprošćavanjem izraza za zadatu logičku funkciju

$$F=A\cdot(B\cdot C+D)+A\cdot C$$

dokazati da vrednost funkcije F ne zavisi od vrednosti promenljive B:

Rešenje:

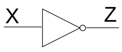
$$F = A \cdot (C + D)$$

Osnovne logičke operacije nad binarnim ciframa

Postoje dve vrste logičkih operacija, zavisno od broja operanada koje u njima učestvuju:

- **unarne**, logičke operacije nad jednim operandom (negacija)
- **binarne**, logičke operacije nad dva operanda (sve druge operacije)

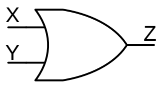
Negacija (NE, NOT) je najprostija logička operacija koja se obavlja nad jednim operandom.Zove se još inverzija ili komplementiranje. Operacija NE konvertuje vrednost 1 u vrednost 0 i obrnuto. U tabeli istinitosti za NE operaciju X je ulazna veličina (operand), a Z je izlazna veličina (rezultat).



X	Z
0	1
1	0

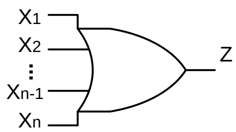
$Z=\overline{X}$

ILI operacija (**OR**) se vrši nad dve ili više ulaznih vrednosti, a naziva se još i logičko sabiranje (disjunkcija). Da bi rezultat **ILI** operacije imao vrednost **1** mora bar jedna ulazna veličina da ima vrednost **1**. U tabeli istinitosti za ILI operaciju nad dve ulazne vrednosti X i Y, kao i tabeli istinitosti za **n** ulaznih vrednosti X_1 ,..., X_n dobija se rezultat Z = 1 kada su jedna ili više ulaznih vrednosti istovremeno jednake 1.



X	Y	Z
0	0	0
0	1	1
1	0	1
1	1	1

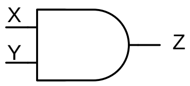
Z=X+Y



X_1	X_2	...	X_{n-1}	X_n	Z
0	0	...	0	0	0
0	0	...	0	1	1
0	0	...	1	0	1
0	0	...	1	1	1
...	...	...	...	...	...
1	1	...	1	0	1
1	1	...	1	1	1

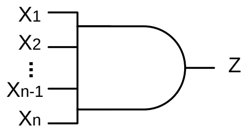
$Z=X_1+X_2+\ldots+X_{n-1}+X_n$

Operacija I (AND) se vrši nad dve ili više ulaznih vrednosti, a naziva se još i logičko množenje (konjunkcija). Rezultat **I** operacije je jednak **0**, ako je bar jedna ulazna vrednost jednaka nuli. Da bi rezultat I operacije bio Z = 1, moraju svi ulazni signali istovremeno da budu jednaki jedinici: X = Y=**1** tj. $X_1 = X_2 = ... = X_n =$**1**.



X	Y	Z
0	0	0
0	1	0
1	0	0
1	1	1

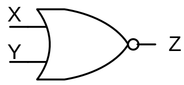
$Z=X \cdot Y$



X_1	X_2	...	X_{n-1}	X_n	Z
0	0	...	0	0	0
0	0	...	0	1	0
0	0	...	1	0	0
0	0	...	1	1	0
...	...	...	...	...	...
1	1	...	1	0	0
1	1	...	1	1	1

$Z=X_1 \cdot X_2 \cdot \ldots \cdot X_{n-1} \cdot X_n$

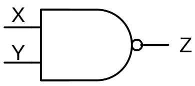
NILI operacija (NOR) daje rezultat **1** samo kada se na svim ulazima nalazi vrednost **0**.



X	Y	Z
0	0	1
0	1	0
1	0	0
1	1	0

$Z=\overline{X+Y}$

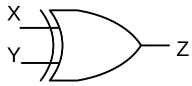
NI operacija (NAND) daje rezultat **0** samo ako se na svim ulazima nalazi vrednost **1**.



X	Y	Z
0	0	1
0	1	1
1	0	1
1	1	0

$Z=\overline{X \cdot Y}$

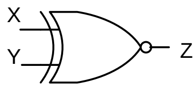
Ekskluzivno ILI (**XOR**) se naziva još i isključivo ILI. Rezultat operacije je **1**, ako je jedna i **samo** jedna od ulaznih veličina jednaka 1, odnosno ako su vrednosti na ulazima uzajamno različite. Rezultat ove operacije odgovara zbiru binarnih cifara (ne uzimajući u obzir prenos), pa se zato ova operacija naziva i sabiranje po modulu dva**.**



X	Y	Z
0	0	0
0	1	1
1	0	1
1	1	0

$Z=X\oplus Y$

Ekskluzivno NILI (**XNOR**) se naziva još isključivo NILI. Rezultat operacije je **1** ako i samo ako su obe ulazne veličine jednake (obe imaju vrednost 0 ili 1). Ovo kolo se zbog toga zove i jednobitni komparator.



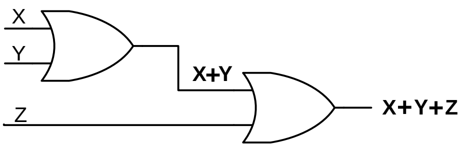
X	Y	Z
0	0	1
0	1	0
1	0	0
1	1	1

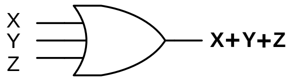
$Z=\overline{X\oplus Y}$

Višeulazna logička kola mogu da se realizuju primenom **dvoulaznih** kola

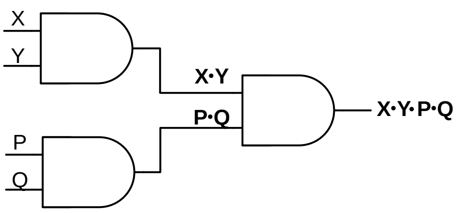
Primer:

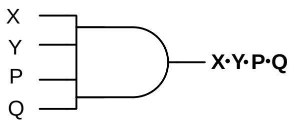
- **ILI** kolo sa tri ulaza (troulazno kolo) može da se realizuje primenom dva dvoulazna ILI kola





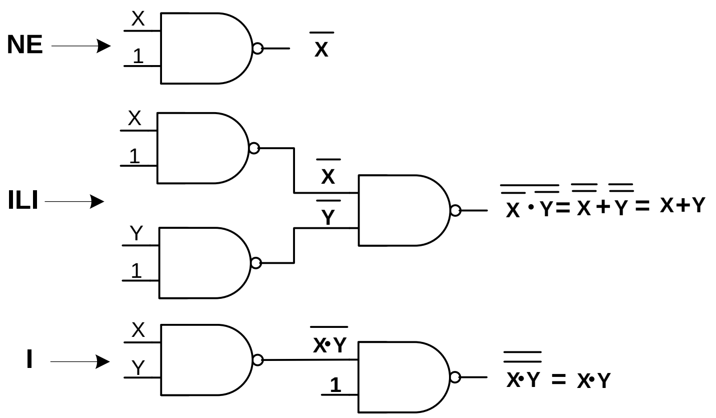
- **I** kolo sa četiri ulaza (četvoroulazno kolo) može da se realizuje primenom tri dvoulazna I kola



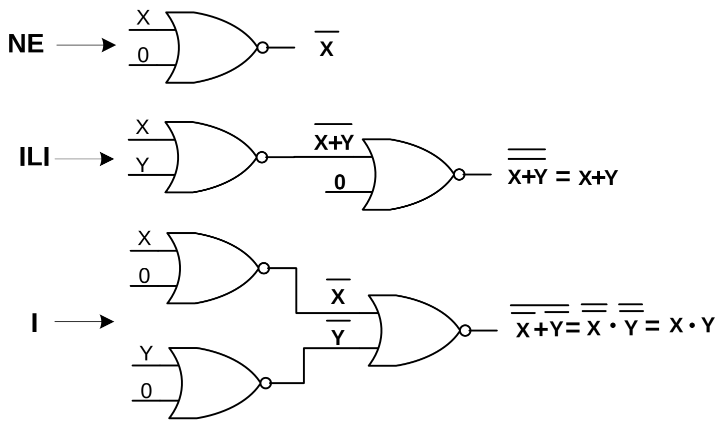


Osnovne logičke operacije mogu da se realizuju primenom samo **jednog tipa** logičkih kola

- Realizacija NE, ILI i I logičke operacije primenom **NI** kola



- Realizacija NE, ILI i I logičke operacije primenom **NILI** kola



<u>Primer 8.</u> Nacrtati šemu logičke mreže kojom se data funkcija

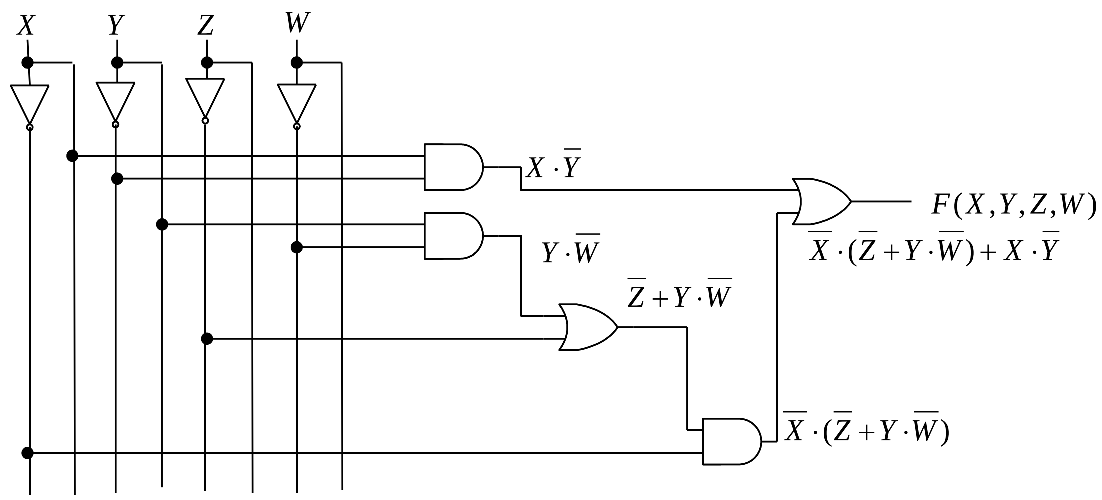
$$F(X,Y,Z,W) = \overline{X}\cdot\overline{Z} + X\cdot\overline{Y} + \overline{X}\cdot Y\cdot\overline{W}$$

realizuje pomoću **NE** kola i dvoulaznih **I** i **ILI** kola.

Rešenje:

Da bi logička mreža mogla da se realizuje samo primenom dvoulaznih logičkih kola, može se izvršiti odgovarajuće grupisanje promenljivih u okviru izraza logičke funkcije:

$$F(X,Y,Z,W) = \overline{X}\cdot\overline{Z} + X\cdot\overline{Y} + \overline{X}\cdot Y\cdot\overline{W} = \overline{X}\cdot(\overline{Z} + Y\overline{W}) + X\cdot\overline{Y}$$

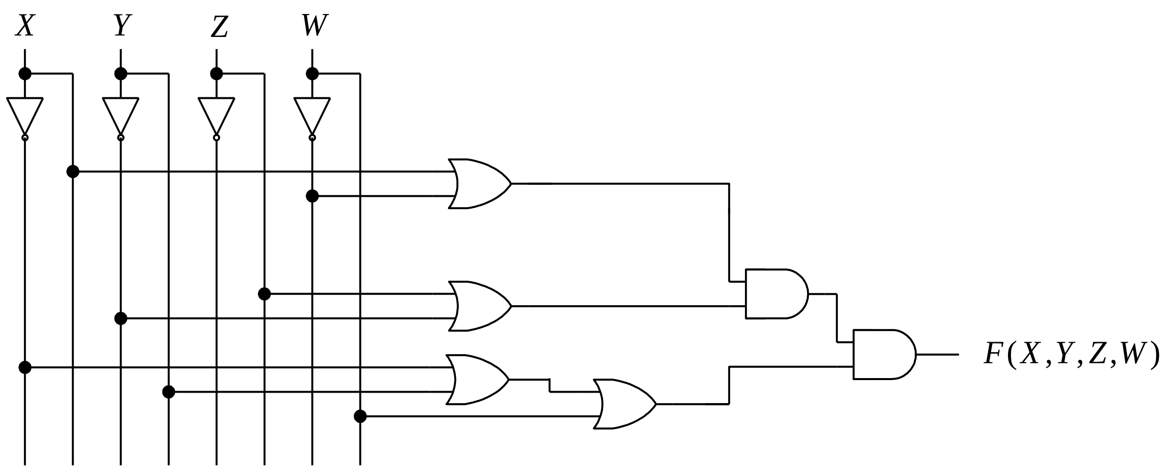


Primer 9. Nacrtati šemu logičke mreže kojom se data funkcija:

$$F(X,Y,Z,W) = (X + \overline{W}) \cdot (Z + \overline{Y}) \cdot (\overline{X} + Y + W)$$

realizuje pomoću **NE** i dvoulaznih **I** i **ILI** kola.

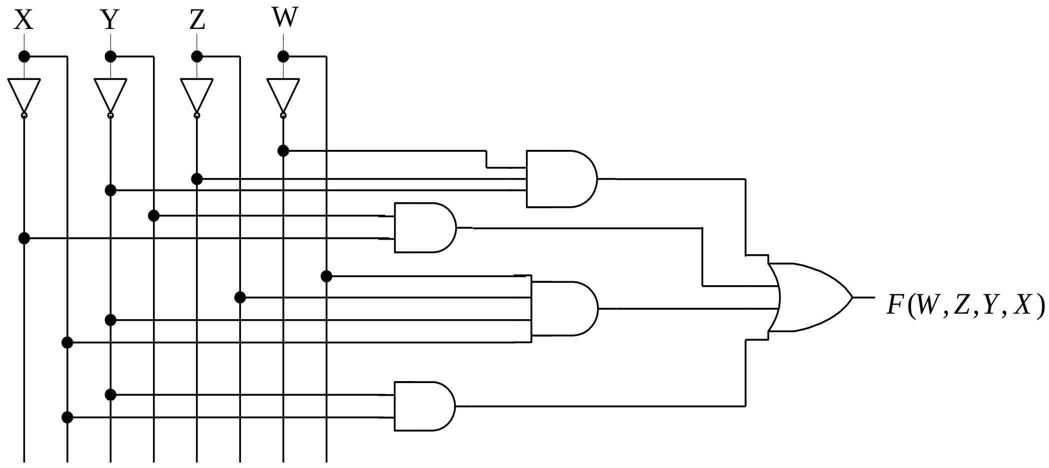
Rešenje:



Primer 10. Primenom **NE** kola i višeulaznih **ILI** i **I** logičkih kola nacrtati šemu logičke mreže kojom se realizuje funkcija F data izrazom:

$$F(W,Z,Y,X) = \overline{W} \cdot \overline{Z} \cdot \overline{Y} + Y \cdot \overline{X} + W \cdot Z \cdot \overline{Y} \cdot X + \overline{Y} \cdot X$$

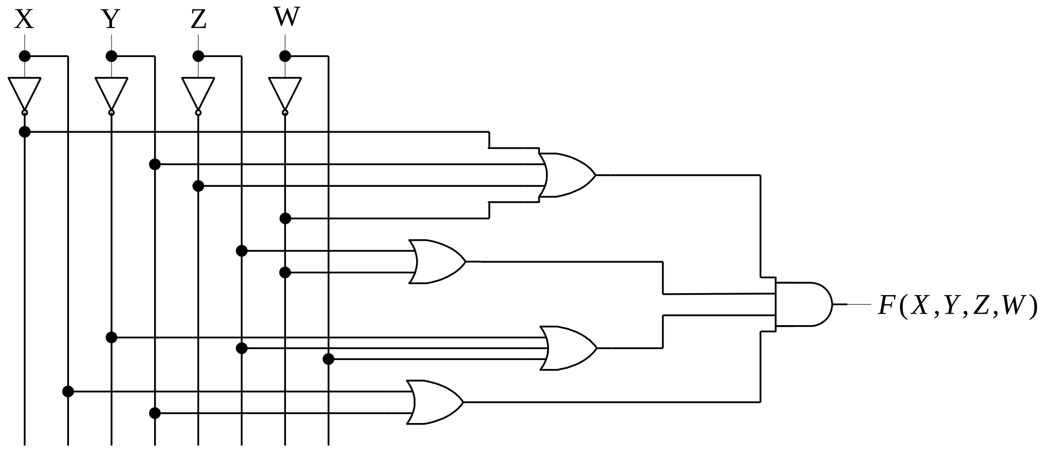
Rešenje:



Primer 11. Primenom **NE** kola i višeulaznih **ILI** i **I** logičkih kola nacrtati šemu logičke mreže kojom se realizuje funkcija F data izrazom:

$$F(X,Y,Z,W) = (\overline{X}+Y+\overline{Z}+\overline{W})\cdot(Z+\overline{W})\cdot(\overline{Y}+Z+W)\cdot(X+Y)$$

Rešenje:



Primer 12. Nacrtati šemu logičke mreže kojom se realizuje funkcija data izrazom:

$$F(A,B,C,D) = (A+\overline{B}+C)\cdot(B+\overline{C}+D)\cdot(\overline{A}+\overline{C}+\overline{D})$$

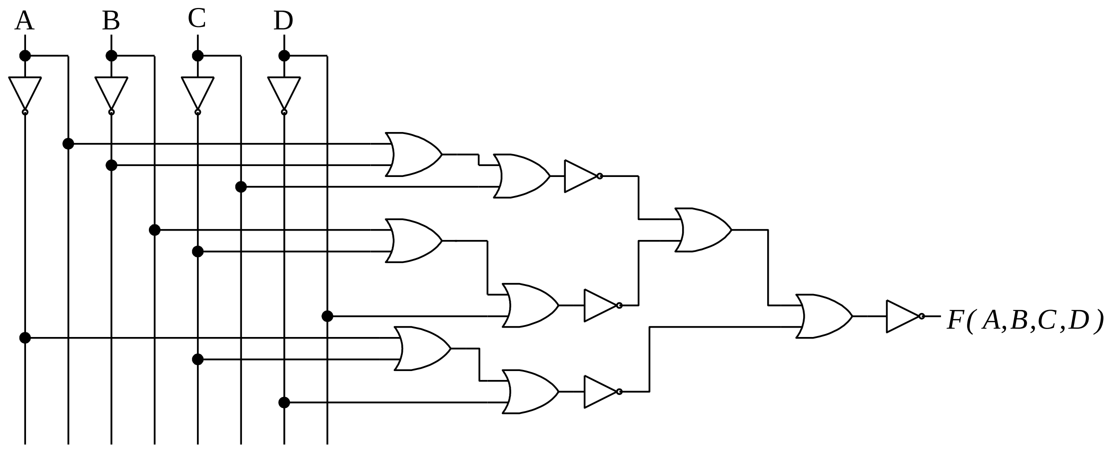
primenom **NE** kola i dvoulaznih **ILI** kola.

Rešenje:

$$F(A,B,C,D) = \overline{\overline{(A+\overline{B}+C)\cdot(B+\overline{C}+D)\cdot(\overline{A}+\overline{C}+\overline{D})}}$$

$$F(A,B,C,D) = \overline{\overline{(A+\overline{B}+C)\cdot(B+\overline{C}+D)}+\overline{(\overline{A}+\overline{C}+\overline{D})}}$$

$$F(A,B,C,D) = \overline{\overline{(A+\overline{B}+C)}+\overline{(B+\overline{C}+D)}+\overline{(\overline{A}+\overline{C}+\overline{D})}}$$



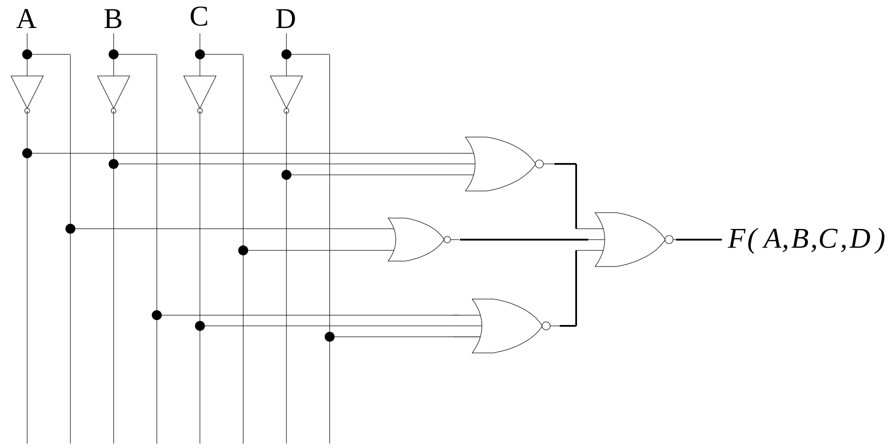
Primer 13. Primenom **NE** kola i višeulaznih **NILI** kola nacrtati šemu logičke mreže kojom se realizuje funkcija F data izrazom:

$$F(A,B,C,D) = (\overline{A}+\overline{B}+\overline{D})\cdot(A+C)\cdot(B+\overline{C}+D)$$

Rešenje:

$$F(A,B,C,D) = \overline{\overline{(\overline{A}+\overline{B}+\overline{D})\cdot(A+C)\cdot(B+\overline{C}+D)}}$$

$$F(A,B,C,D) = \overline{\overline{(\overline{A}+\overline{B}+\overline{D})}+\overline{(A+C)}+\overline{(B+\overline{C}+D)}}$$

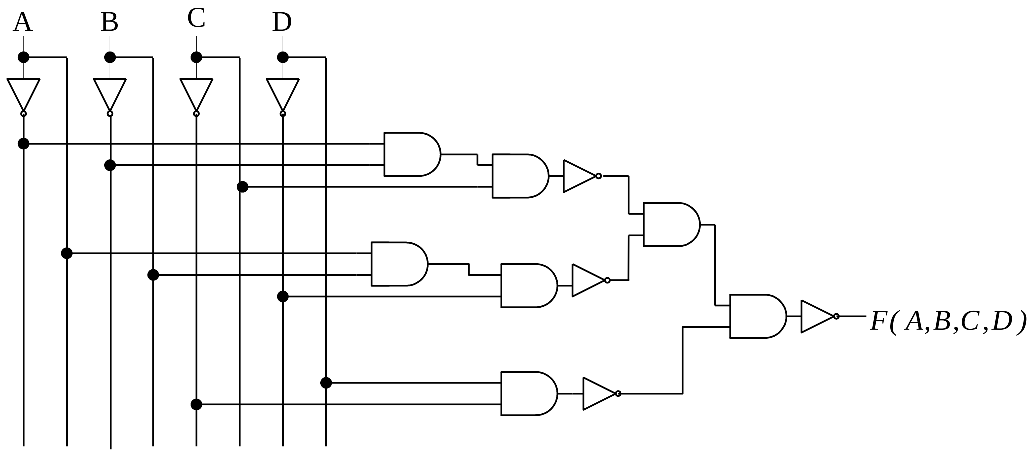


Primer 14. Primenom **NE** kola i dvoulaznih **I** kola nacrtati šemu logičke mreže kojom se realizuje funkcija F data izrazom:

$$F(A,B,C,D) = (\overline{A}\cdot\overline{B}\cdot C)+(A\cdot B\cdot\overline{D})+(D\cdot\overline{C})$$

Rešenje:

$$F(A,B,C,D)=\overline{\overline{\overline{(\overline{A}\cdot\overline{B}\cdot C)}\cdot\overline{(A\cdot B\cdot\overline{D})}\cdot\overline{(D\cdot\overline{C})}}}$$



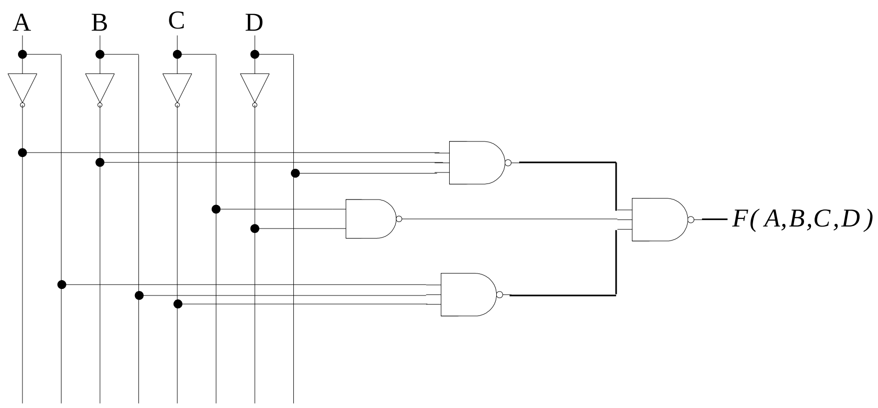
Primer 15. Primenom **NE** kola i višeulaznih **NI** kola nacrtati šemu logičke mreže kojom se realizuje funkcija F data izrazom:

$$F(A,B,C,D)=(\overline{A}\cdot\overline{B}\cdot D)+(C\cdot\overline{D})+(A\cdot B\cdot\overline{C})$$

Rešenje:

$$F(A,B,C,D)=\overline{\overline{(\overline{A}\cdot\overline{B}\cdot D)+(C\cdot\overline{D})+(A\cdot B\cdot\overline{C})}}$$

$$F(A,B,C,D)=\overline{\overline{(\overline{A}\cdot\overline{B}\cdot D)}\cdot\overline{(C\cdot\overline{D})}\cdot\overline{(A\cdot B\cdot\overline{C})}}$$



Primer 16. Logička funkcija je zadata izrazom:

$$F(A,B,C,D)=A\cdot D+\overline{B}\cdot C+\overline{A}\cdot B\cdot\overline{C}\cdot D+B\cdot C$$

a) Odrediti vrednost funkcije ako se na ulaz A dovede niz logičkih nivoa 10100011, na ulaz B: 11010101, na ulaz C: 10101000 i na ulaz D: 10101011.

b) Nacrtati šemu logičke mreže kojom se realizuje data funkcija, koristeći **NE** i dvoulazna **I** kola.

Rešenje:

a) Da bi se lakše odredila vrednost funkcije, može se primeniti analitičko uprošćavanje datog izraza primenom identiteta iz Primera 3.

$$F(A,B,C,D) = A\cdot D + \overline{B}\cdot C + \overline{A}\cdot B\cdot \overline{C}\cdot D + B\cdot C = D\cdot\left(A + \overline{A}\cdot B\cdot \overline{C}\right) + C\cdot\left(B + \overline{B}\right) =$$

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

$$= A\cdot D + C + B\cdot D = C + D\cdot(A + B)$$

A	*B*	*C*	*D*	*A*+*B*	*D*(*A*+*B*)	*F*(*A*,*B*,*C*,*D*)=*C*+*D*(*A*+*B*)
1	1	1	1	1	1	**1**
0	1	0	0	1	0	**0**
1	0	1	1	1	1	**1**
0	1	0	0	1	0	**0**
0	0	1	1	0	0	**1**
0	1	0	0	1	0	**0**
1	0	0	1	1	1	**1**
1	1	0	1	1	1	**1**

F(*A*,*B*,*C*,*D*)=**10101011**

b) Šema logičke mreže realizuje se na osnovu analitičke transformacije izraza za funkciju na sličan način kao što je objašnjeno u Primeru 12.

Primer 17. Odrediti vrednosti logičke funkcije:

$$F(X,Y,Z,W) = (\overline{Y} + \overline{Z})\cdot(X + W)\cdot(\overline{Y} + Z + \overline{W})\cdot(X + W)$$

ako su vrednosti ulaznih promenljivih:

a) $X = 0$, $Y = 1$, $Z = 1$, $W = 1$

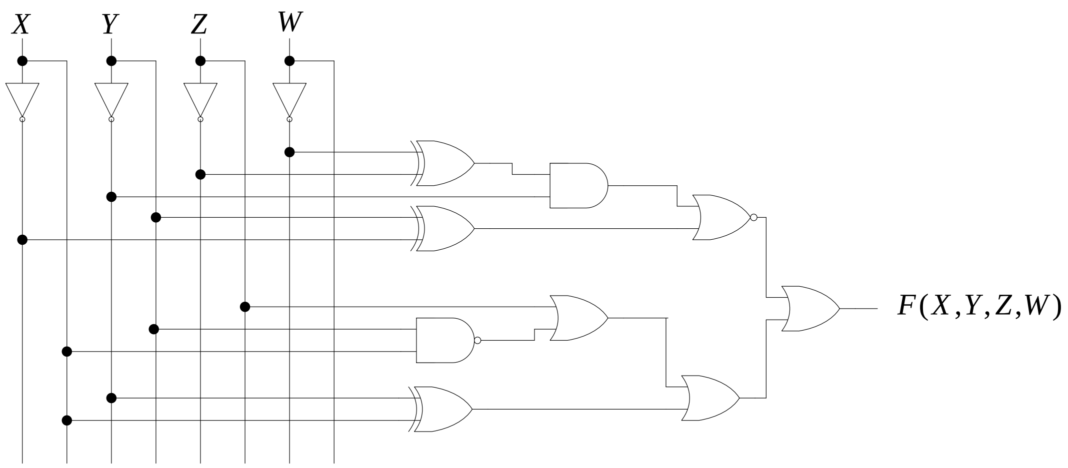
b) $X = 1$, $Y = 0$, $Z = 0$, $W = 0$

Rešenje:

a) $F(X,Y,Z,W) = F(0,1,1,1) = (\overline{1} + \overline{1})\cdot(0+1)\cdot(\overline{1} + 1 + \overline{1})\cdot(0+1) = 0\cdot 1\cdot 1\cdot 1 = 0$

b) $F(X,Y,Z,W) = F(1,0,0,0) = (\overline{0} + \overline{0})\cdot(1+0)\cdot(\overline{0} + 0 + \overline{0})\cdot(1+0) = 1\cdot 1\cdot 1\cdot 1 = 1$

Primer 18. a) Odrediti izraz za funkciju *F*(*X*, *Y*, *Z*, *W*) koja opisuje rad kombinacione mreže prikazane na slici:



b) Odrediti vrednost funkcije F ako su vrednosti promenljivih

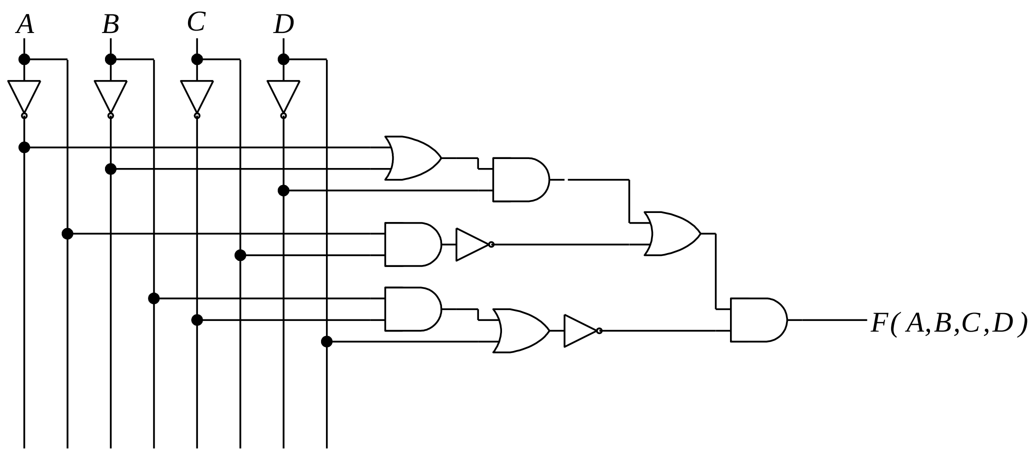
$X = 1,\ Y = 1,\ Z = 0,\ W = 0$

Rešenje:

a) $F(X,Y,Z,W) = \overline{(\overline{Z} \oplus \overline{W})\cdot\overline{Y} + (\overline{X} \oplus Y)} + \overline{X \cdot Y} + Z + X \oplus \overline{Y}$

b) $F(X,Y,Z,W) = F(1,1,0,0) = 1$

Primer 19. a) Odrediti izraz za funkciju F(A, B, C, D) koja opisuje rad kombinacione mreže prikazane na slici:



b) Odrediti vrednost funkcije F ako su vrednosti promenljivih

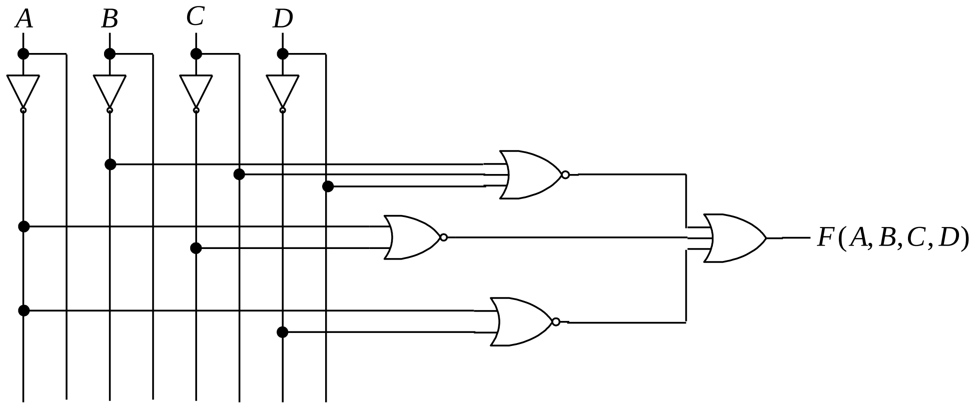
$A = 0,\ B = 0,\ C = 1,\ D = 1$

Rešenje:

a) $F(A,B,C,D) = \left((\overline{A} + \overline{B}) \cdot \overline{D} + \overline{A \cdot C}\right) \cdot \left(\overline{B \cdot \overline{C} + D}\right)$

b) $F(A,B,C,D) = F(0,0,1,1) = 0$

Primer 20. a) Odrediti izraz za funkciju F(A, B, C, D) prikazanu na slici.



b) Primenom analitičke transformacije izraza za logičku funkciju dokazati da funkcija može da se realizuje primenom NE i višeulaznih NI logičkih kola.

Rešenje:

a) $F(A,B,C,D)=\overline{\overline{B}+C+D}+\overline{\overline{A}+\overline{C}}+\overline{\overline{A}+\overline{D}}$

b) Primenom De Morganove teoreme u izrazu dobijenom u a) i dvostrukom negacijom izraza, dobija se:

$$F(A,B,C,D)=\overline{\overline{B\cdot\overline{C}\cdot\overline{D}}\cdot\overline{A\cdot C}\cdot\overline{A\cdot D}}$$

na osnovu koga je moguća implementacija funkcije primenom NE i višeulaznih NI kola.

Primer 21. Logička funkcija je zadata izrazom:

$$F(A,B,C,D)=\overline{A}\cdot B\cdot D+\overline{A}\cdot C+\overline{A}\cdot B\cdot C\cdot\overline{D}$$

a) Izvršiti analitičko uprošćavanje izraza logičke funkcije

b) Nacrtati šemu logičke mreže na osnovu uprošćenog logičkog izraza primenom NE i višeulaznih NI logičkih kola.

c) Odrediti vrednost funkcije, ako se na ulaze dovodi niz logičkih nivoa

A: 10100011, B: 01010001, C: 01110011 i D: 01011100.

Rešenje:

a) Analitičko uprošćavanje izraza logičke funkcije može da se uradi grupisanjem ulaznih promenljivih i primenom identiteta dokazanog u Primeru 3:

$$\begin{aligned}F(A,B,C,D)&=\overline{A}\cdot B\cdot D+\overline{A}\cdot C+\overline{A}\cdot B\cdot C\cdot\overline{D}=\\&=\overline{A}\cdot B\cdot(D+C\cdot\overline{D})+\overline{A}\cdot C=\\&=\overline{A}\cdot B\cdot(D+C)+\overline{A}\cdot C=\\&=\overline{A}\cdot B\cdot D+\overline{A}\cdot C\cdot(B+1)=\overline{A}\cdot(B\cdot D+C)\end{aligned}$$

b) Videti Primer 15.

c) $F(A, B, C, D)$ = 01010000

Primer 22. Data je funkcija:

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

a) Realizovati ovu funkciju pomoću jednog višeulaznog logičkih kola.

b) Ako se na ulaz A dovede niz logičkih nivoa 011001, Na ulaz B niz 001010 i na ulaz C niz 101010, kakav će se niz dobiti na izlazu.

Rešenje:

a) $F(A,B,C)=\overline{\overline{A}\cdot\overline{B}}+C=A+B+C$, potrebno je jedno troulazno ILI kolo.

b) $F(A,B,C)=111011$

Primer 23. Odrediti vrednosti logičke funkcije

$$F(A,B,C,D)=\overline{A}\cdot B\cdot\overline{C}+\overline{B}\cdot\overline{D}+A\cdot C\cdot D+B\cdot C$$

ako su vrednosti ulaznih promenljivih:

a) A = 1, B = 0, C = 1, D = 1

b) A = 1, B = 1, C = 0, D = 0

Rešenje:

a) $F(A,B,C,D)=1$

b) $F(A,B,C,D)=0$

Primer 24. Nacrtati šemu kombinacione mreže kojom se realizuje funkcija data izrazom:

$$F(A,B,C,D)=(A+\overline{C}+D)\cdot(\overline{B}+C+\overline{D})\cdot(\overline{A}+B+D)\cdot(C+\overline{D})$$

a) Primenom **NE** kola i dvoulaznih **I** i **ILI** logičkih kola.

b) Primenom **NE** kola i dvoulaznih **ILI** kola.

Primer 25. Nacrtati šemu kombinacione mreže kojom se realizuje funkcija data izrazom:

$$F(X,Y,Z,W)=(Y+W)\cdot(X+\overline{Z}+\overline{W})\cdot(\overline{X}+\overline{Y}+Z)\cdot(Z+\overline{W})$$

a) Primenom **NE** kola i višeulaznih **I** i **ILI** logičkih kola.

b) Primenom **NE** kola i višeulaznih **NILI** kola.

Primer 26. Nacrtati šemu kombinacione mreže kojom se realizuje funkcija data izrazom:

$$F(A,B,C,D)=B\cdot C\cdot D+\overline{A}\cdot\overline{C}\cdot D+A\cdot B\cdot\overline{D}+A\cdot\overline{B}\cdot\overline{D}$$

a) Primenom **NE** kola i dvoulaznih **I** i **ILI** logičkih kola.

b) Primenom **NE** kola i dvoulaznih **I** kola.

Primer 27. Nacrtati šemu kombinacione mreže kojom se realizuje funkcija data izrazom:

$$F(X,Y,Z,W) = A \cdot B \cdot D + A \cdot C \cdot \overline{D} + B \cdot \overline{C} + \overline{C} \cdot \overline{D}$$

a) Primenom **NE** kola i višeulaznih **I** i **ILI** logičkih kola.

b) Primenom **NE** kola i višeulaznih **NI** kola.

Primer 28. Nacrtati šemu kombinacione mreže kojom se realizuje funkcija data izrazom:

$$F(X,Y,Z,W) = (\overline{X} + Y) \cdot (Y + \overline{Z}) \cdot (X + Z + W) \cdot (X + \overline{Y} + Z)$$

a) Primenom **NE** kola i višeulaznih **ILI** i **I** logičkih kola.

b) Primenom **NE** kola i višeulaznih **ILI** kola.

Primer 29. Nacrtati šemu kombinacione mreže kojom se realizuje funkcija data izrazom:

$$F(A,B,C,D) = \overline{B} \cdot \overline{C} \cdot D + A \cdot B \cdot C + B \cdot \overline{C} \cdot \overline{D} + A \cdot D$$

a) Primenom **NE** kola i višeulaznih **ILI** i **I** logičkih kola.

b) Primenom **NE** kola i višeulaznih **I** kola.

6. MINIMIZACIJA LOGIČKIH FUNKCIJA PRIMENOM KARNOOVIH MAPA

Određivanje minimalne disjunktivne forme (MDF)

Svako polje Karnoove mape je rezervisano za tačno jednu moguću konjunkciju promenljivih ili njihovih negacija. Ukoliko se određena konjunkcija pojavljuje u funkciji, onda se u dato polje zapisuje 1, a ako se ne pojavljuje, ne zapisuje se ništa. Na taj način se jednoznačno preslikava tabelarni prikaz ili logički izraz funkcija dve, tri ili četiri promenljive na odgovarajuću Karnoovu mapu. Broj polja u Karnoovoj mapi za funkciju N promenljivih je 2^N. Mesto konjunkcije u Karnoovoj mapi, kao i šema popunjavanja Karnoove mape zavise od izabranog tipa mape.

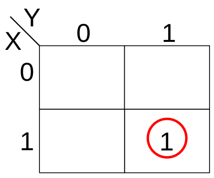
- **Karnoova mapa za funkciju dve promenljive X i Y** sastoji se od 4 polja, jer od dve promenljive mogu da formiraju maksimalno 4 potpuna logička proizvoda, odnosno konjunkcije. Ako je data konjunkcija $F(X,Y) = X \cdot Y$, njeno mesto u Tabeli 1 određeno je kombinacijom vrednosti ulazanih promenljivih **X = 1** i **Y = 1**.

$$F(X,Y) = (X \cdot Y)$$

i	X	Y	F
0	0	0	0
1	0	1	0
2	1	0	0
3	1	1	1

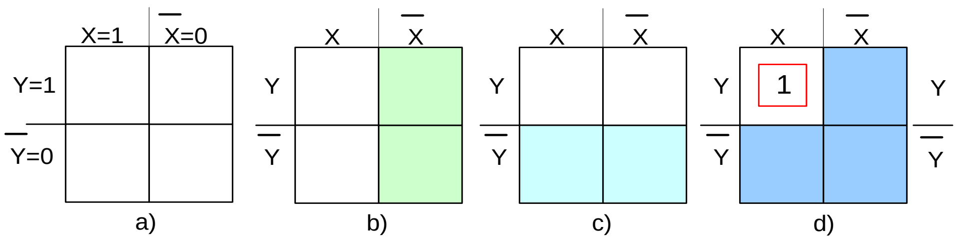
Tabela 1. Tabela istinitosti za funkciju dve promenljive

  - U Karnoovoj mapi na Slici 1, mesto koje pripada zadatoj konjunkciji mora da se nalazi u preseku reda u kome je **X = 1** i kolone u kojoj je **Y = 1,** jer su u zadatoj konjunkciji X i Y bez negacije.



Slika 1. Popunjavanje Karnoove mape prvog tipa za funkciju dve promenljive

  - U Karnoovoj mapi na Slici 2a, do mesta koje pripada zadatoj konjunkciji, u kojoj su X i Y bez negacije dolazi se eliminacijom polja koja ne zadovoljavaju kriterijum da je **X = 1** i **Y = 1**. Iz tog razloga je na Slici 2b osenčena oblast u kojoj je **X = 0**, dok je na Slici 2c) osenčena oblast u kojoj je **Y = 0**. Mesto za konjunkcija $X \cdot Y$ se nalazi u preseku ove dve osenčene oblasti i u to polje se upisuje **1** (Slika 2d).



Slika 2. Popunjavanje Karnoove mape drugog tipa za funkciju dve promenljive

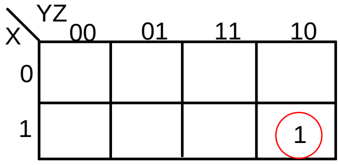
- **Karnoova mapa za funkciju tri promenljive X**, **Y** i **Z**, sastoji se od 8 polja, jer od tri promenljive može da se formira maksimalno 8 potpunih logičkih proizvoda, odnosno konjunkcija. Ako je data konjunkcija $F(X, Y, Z) = X \cdot Y \cdot \overline{Z}$, njeno mesto u Tabeli 2 određeno je kombinacijom ulaznih vrednosti promenljivih **X = 1**, **Y = 1** i **Z = 0**.

$$F(X, Y, Z) = X \cdot Y \cdot \overline{Z}$$

i	X	Y	Z	F
0	0	0	0	0
1	0	0	1	0
2	0	1	0	0
3	0	1	1	0
4	1	0	0	0
5	1	0	1	0
6	1	1	0	**1**
7	1	1	1	0

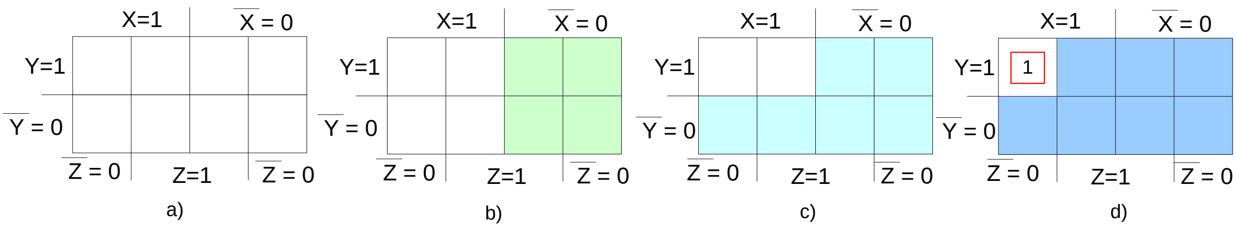
Tabela 2. Tabela istinitosti za funkciju tri promenljive

  - U Karnoovoj mapi na Slici 3, mesto zadate konjunkcije nalazi se u preseku reda u kome je **X = 1** (X bez negacije) i kolone u kojoj su **Y = 1** (Y bez negacije) i **Z = 0** (Z bez negacije).



Slika 3. Popunjavanje Karnoove mape prvog tipa za funkciju tri promenljive

  - U Karnoovoj mapi na Slici 4a, pošto je u zadatom izrazu promenljiva **X** bez negacije, u obzir dolaze 4 leva polja. Desna polovina mape se osenči, jer ta polja ne dolaze u obzir (Slika 4b). Druga promenljiva je **Y**, pa se 4 leva polja smanjuju samo na 2, i to u gornjoj vrsti, a donja vrsta se osenči (Slika 4c). Konačno, treći element je $\overline{Z}$, pa treba osenčiti polja u sredini u kojima je **Z = 1**. Preostalo, neosenčeno polje određuje polje datog izraza i u njega upisujemo **1** (Slika 4d). Analognim postupkom može se naći odgovarajuće polje za bilo koju konjunkciju 3 promenljive.



Slika 4. Popunjavanje Karnoova mape drugog tipa za funkciju tri promenljive

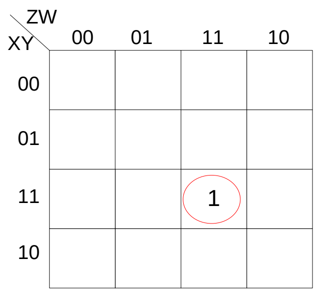
  - **Karnoova mapa za funkciju četiri promenljive X**, **Y**, **Z** i **W**, sastoji se od 16 polja, jer od četiti promenljive može da se formira maksimalno 16 potpunih logičkih proizvoda, odnosno konjunkcija. Ako je data konjunkcija F(X,Y,Z,W)=X·Y·Z·W, njeno mesto u Tabeli 3 određeno je kombinacijom ulaznih vrednosti promenljivih **X = 1**, **Y = 1**, **Z = 1** i **W = 1**.

$$F(X, Y, Z, W) = X \cdot Y \cdot Z \cdot W$$

i	X	Y	Z	W	F
0	0	0	0	0	0
1	0	0	0	1	0
2	0	0	1	0	0
3	0	0	1	1	0
4	0	1	0	0	0
5	0	1	0	1	0
6	0	1	1	0	0
7	0	1	1	1	0
8	1	0	0	0	0
9	1	0	0	1	0
10	1	0	1	0	0
11	1	0	1	1	0
12	1	1	0	0	0
13	1	1	0	1	0
14	1	1	1	0	0
15	1	1	1	1	1

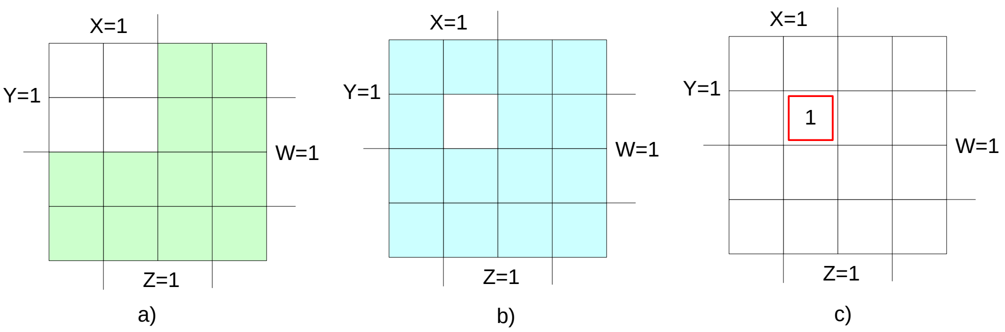
Tabela 3. Karnoova mapa za funkciju četiri promenljive

- U Karnoovoj mapi na Slici 5, mesto zadate konjunkcije nalazi se u preseku reda u kome su **X = 1** (X bez negacije), **Y = 1** (Y bez negacije) i kolone u kojoj su i **Z = 1** (Z bez negacije) i **W = 1** (W bez negacije)**.**



Slika 5. Popunjavanje Karnoova mape prvog tipa za funkciju četiri promenljive

- U Karnoovoj mapi na Slici 6a, pošto Pošto su sve promenljive bez negacije, treba osenčiti polja u kojima promenljive uzimaju vrednost 0 (Slike 6a i 6b), a u preostalo neosenčeno polje se upisuje 1 koja odgovara ovoj konjunkciji (Slika 6c).



Slika 6. Popunjavanje Karnoove mape drugog tipa za funkciju četiri promenljive

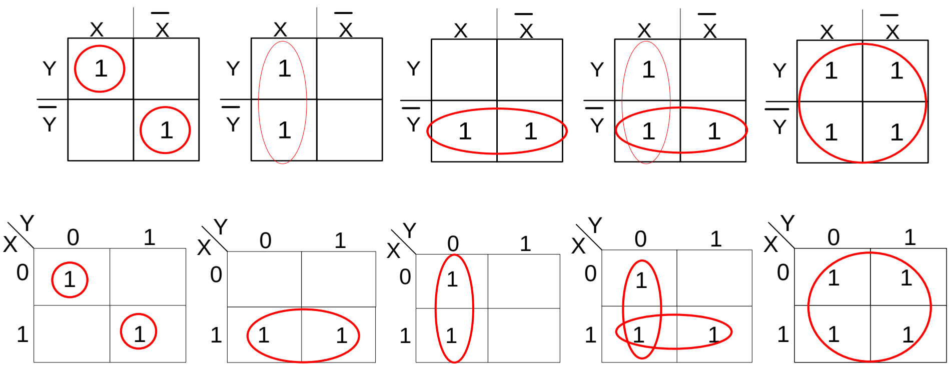
Proces minimizacije u cilju dobijanja analitičkog izraza u **minimalnoj disjunktivnoj formi (MDF)** sastoji se u grupisanju jedinica u okviru mape u veće celine. Pri tome se nastoji da ove celine budu što veće, ali one ne smeju da sadrže polja koja nemaju 1. Pri grupisanju polja treba primeniti sledeća opšta pravila:

- Broj grupisanih jedinica mora da bude 2^N (N = 0, 1, 2, 3 ili 4).
- Treba formirati minimalan broj što većih grupa tako da se obuhvate sve jedinice.
- Preklapanje grupa ne smeta ako se time prave veće grupe.
- Moraju se grupisati **sve jedinice.**

Pomenuta pravila garantuju minimalnu funkciju u disjunktivnoj formi sa gledišta potrebnog broja logičkih kola da bi se funkcija realizovala ali to ne znači da ne može da postoji druga funkcija koja bi bila bolja ili jednostavnija za konkretnu primenu. Takođe ne postoji garancija da je disjunktivna forma optimalni izbor uvek.

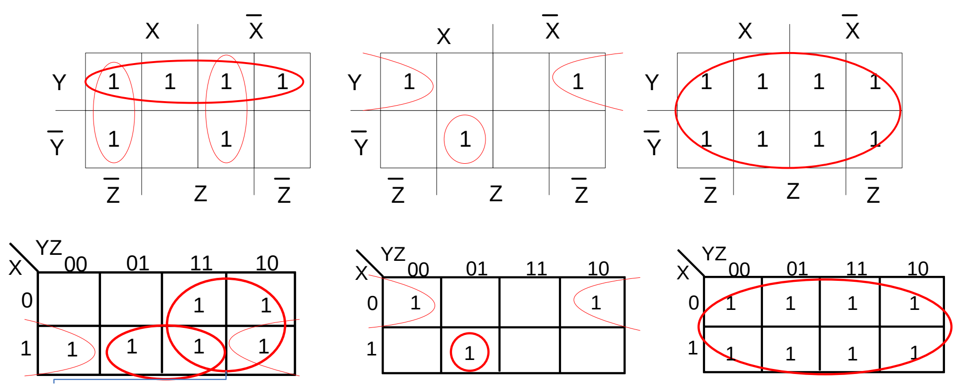
Minimalna disjunktivna forma ima onoliko članova koliko je grupa napravljeno. Svaki član je logički proizvod (logička I funkcija) do četiri ulaza (za funkciju 4 promenljive) ili njihovih invertovanih vrednosti. Ako grupu čine dve jedinice, član ima tri elementa; ako grupu čine četiri jedinice, član ima dva elementa, a samo jedan element ako smo grupisali osam jedinica. Članovi su međusobno povezani logičkom ILI funkcijom.

- Primeri grupisanja polja u **Karnoovim mapama sa 2 promenljive** dati su na slici 7.



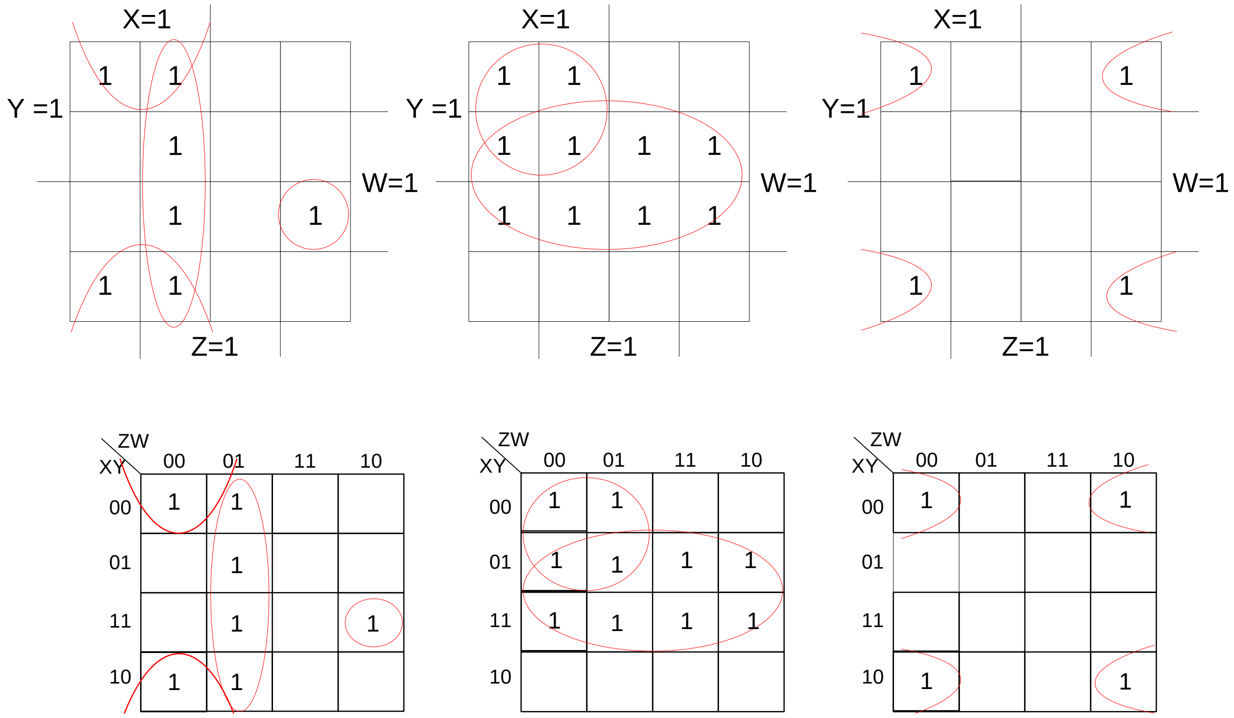
Slika 7. Grupisanje jedinica u Karnoovoj mapi sa dve promenljive

Primeri grupisanja polja u u **Karnoovojim mapama sa 3 promenljive** prikazani su na Slici 8. Ovaj tip mapa može se zamisliti kao da tabela nacrtana na valjku, čiji je omotač po visini razvijen u ravan. Zbog toga je moguće grupisanje polja uz levu i desnu ivicu zajedničkom konturom.



Slika 8. Grupisanje jedinica u Karnoovoj mapi sa tri promenljive

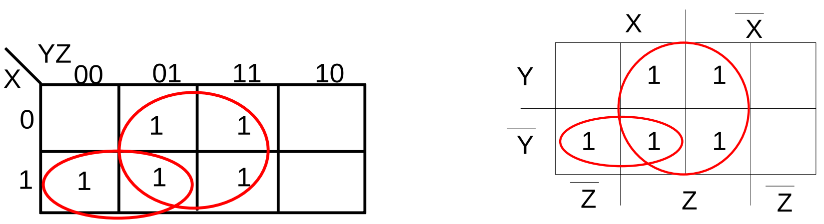
Primeri pravilnog grupisanja polja u **Karnoovim mapama sa 4 promenljive** dati su na Slici 9. Kao i u prethodnim primerima, primenjeno je pravilo da treba formirati minimalan broj grupa tako da budu obuhvaćene sve jedinice u Karnoovoj mapi.



Slika 9. Grupisanje jedinica u Karnoovoj mapi sa četiri promenljive

Određivanje analitičkog oblika minimalne disjunktivne forme logičke funkcije (**MDF**) na osnovu formiranih grupa polja sa **jedinicama** sastoji se u sledećem:

- Za svaku formiranu grupu definiše se konjunkcija promenljivih ili njihovih negacija, ali samo onih koje su za celu grupu nepromenjene.
- Kada su formirane konjunkcije za sve grupe, konačna funkcija dobija se kao disjunkcija ovih pojedinačnih članova. Primeri mapa sa 3 promenljive, u kojima su već formirane grupe prikazan je na slikama 10a i 10b.



a) b)

Slika 10. Određivanje analitičkog izraza za grupu jedinica

- Prvo definišemo odgovarajuću konjunkciju za grupu od 4 srednje jedinice. U ovom slučaju grupa ima jedinice u poljima X (**X = 1**) i $\overline{X}$ (**X = 0**), pa ta varijabla neće figurisati u rezultujućem izrazu. U datoj grupi ima polja i sa Y (**Y = 1)** i sa $\overline{Y}$ (**Y = 0),** pa ni ova varijabla ne utiče na rezultujuću konjunkciju. Grupa je definisana u svim poljima sa **Z = 1** (oblast **Z**). Odavde sledi da je prvi faktor minimalne logičke funkcije **Z** (samo jedna promenljiva).

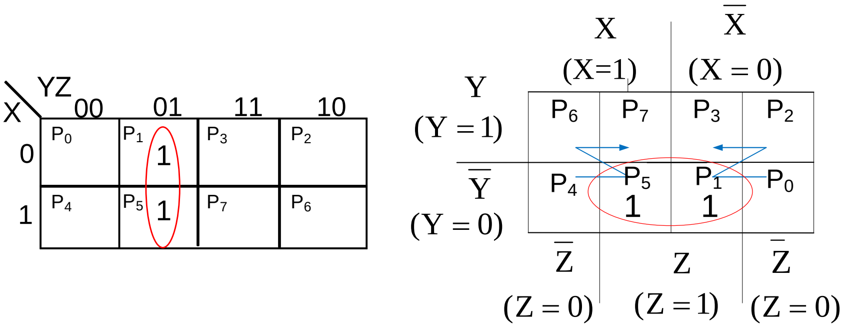
- Druga grupa sastoji se od dva polja u donjem levom uglu obe mape. Vrednost varijable **X** je jedinstvena u ovoj grupi (**X = 1**), pa ulazi u konjunkciju. Zatim, u grupi postoji polje sa varijablom Z, ali i sa $\overline{Z}$, pa, prema tome, ova varijabla ne ulazi u konačni izraz za ovu grupu. Kako je za ovu grupu i varijabla $\overline{Y}$ jedinstvena, to i ona ulazi u izraz, pa je konačan oblik konjunkcije za ovu grupu: $X \cdot \overline{Y}$. Minimalni oblik ove funkcije dobija se povezivanjem konjunkcija za pojedine grupe operatorom disjunkcije: $F(X, Y, Z) = Z + X \cdot \overline{Y}$

Da bi se tabela istinitosti preslikala na Karnoovu mapu, posmatraju se svi redovi u kojima logička funkcija ima vrednost **1** za odgovarajuće kombinacije vrednosti ulaznih promenljivih. Svaka kombinacija vrednosti ulaznih promenljivih definiše jedno određeno polje u mapi

- Kada promenljiva ima vrednost 1, posmatra se u Karnoovoj mapi oblast u kojoj je ta promenljiva jednaka jedinici, a ako ima vrednost 0, posmatra se njena negacija.
- Karnoova mapa može popuniti po šablonu, jer svakoj vrsti iz tabele uvek odgovara tačno određeno mesto u Karnoovoj mapi. Kako je položaj konjunkcija fiksiran, to se samo umesto P_0,.., P_7 upišu jedinice ili nule
    - Primer preslikavanja Tabele 4 u Karnoove mape po šablonu prikazan je na Slikama 11a i 11b.

0	0	0	0	P_0	0
1	0	0	1	P_1	1
2	0	1	0	P_2	0
3	0	1	1	P_3	0
4	1	0	0	P_4	0
5	1	0	1	P_5	1
6	1	1	0	P_6	0
7	1	1	1	P_7	0

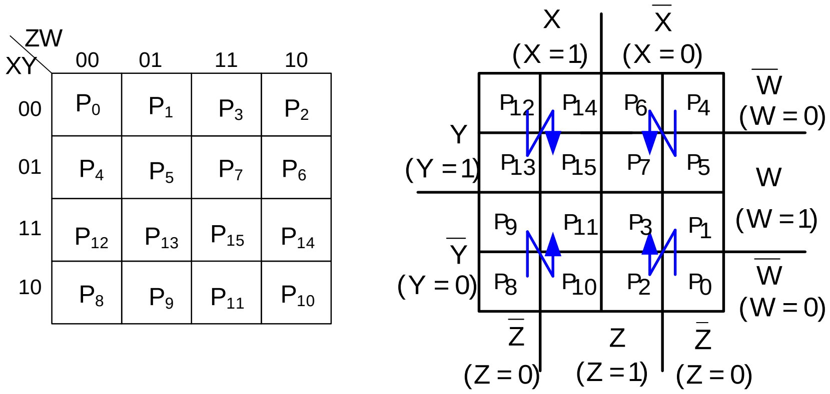
Tabela 4. Karnoova mapa za funkciju četiri promenljive



a) b)

Slika 11. Popunjavanje Karnoove mape i minimizacija funkcije **F**

Raspored polja u Karnoovim mapama za četiri ulazne promenljive prikazan je na Slikama 12a i 12b. Ove Karnoove mape treba zamisliti kao da su nacrtana na torusu, pa razvijene u ravan. Drugim rečima, leva ivica mape se direktno naslanja na desnu, a donja ivica na gornju. Ovo treba imati u vidu prilikom zaokruživanja susednih članova.



a) b)

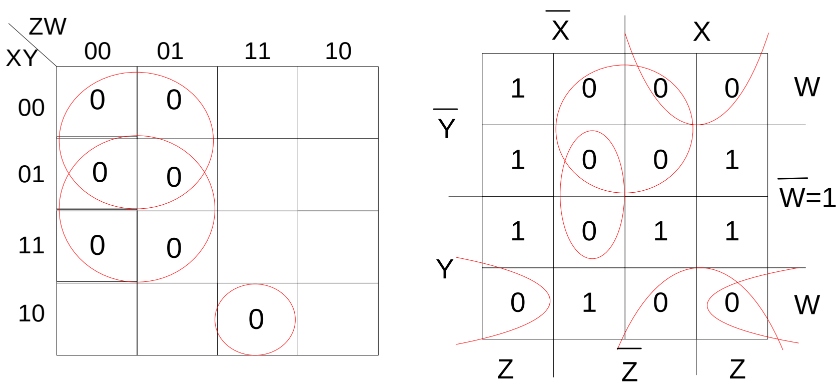
Slika 12. Karnoove mape za četiri promenljive

Određivanje analitičkog oblika minimalne konjunktivne forme logičke funkcije (MKF) na osnovu formiranih grupa polja sa **nulama** sastoji se u sledećem:

Na Slici 13 prikazane su primeri Karnoove mape sa grupisanim poljima u cilju dobijanja minimalne konjunktivne forme funkcije.

- Treba grupisati pojedinačna polja, parove, kvartete i oktete nula i to na isti način kao što se grupišu jedinice kod disjunktivne forme.
- Treba odrediti analitički izraz za minimalne disjunkcije, tako što se **promenljive jednake nuli** uzimaju bez negacije. U Karnoovoj mapio, blast **X=0** je oblast **X**, a oblast **X=1** je oblast $\overline{X}$.
- **MKF** ima onoliko logičkih suma (disjunkcija) koliko u tabeli postoji zaokruženih polja, parova, kvarteta i okteta nula.
- Finalni oblik MKF dobija se kada se napravi logički proizvod (konjunkcija) svih logičkih suma (disjunkcija):
- Karnoova mapa prvog tipa (Slika 13a) ima isti oblik i kao u slučaju nalaženja MDF (Slika 12a), dok se u Karnoovoj mapi drugog tipa (Slika 13b) položaj promenljivih sa negacijom i bez negacije razlikuje u odnosu na Sliku 12b.

$$F = (\overline{Y} + \overline{Z}) \cdot (\overline{X} + \overline{Z} + \overline{W}) \cdot (X + W) \cdot (Y + Z + W)$$



a) b)

Slika 13. Primer Karnoove mapa za funkciju sa četiri promenljive

Primer 1. Primenom Karnoove mape odrediti minimalnu disjunktivnu formu (MDF) funkcije koja je data izrazom:

$$F(A, B, C) = A \cdot B \cdot C + A \cdot \bar{B} \cdot C + \bar{A} \cdot B \cdot C + \bar{A} \cdot \bar{B} \cdot \bar{C}$$

Rešenje:

Izraz za funkciju Y sastoji se od sume potpunih proizvoda, odnosno može da se predstavi u obliku: $F(A, B, C) = \sum(0, 3, 5{,}7)$

A \ BC	00	01	11	10
0			1	
1		1	1	1

$$F_{MDF}(A,B,C) = A \cdot C + B \cdot C + \overline{A} \cdot \overline{B} \cdot \overline{C}$$

Primer 2. Primenom Karnoove mape odrediti minimalnu disjunktivnu (MDF) funkcije F koja je zadata kao suma potpunih proizvoda:

$$F(A,B,C) = \sum(0{,}1{,}2{,}4{,}6)$$

Rešenje:

A \ BC	00	01	11	10
0	1	1		1
1	1			1

	A	A	$\overline{A}$	$\overline{A}$
B	**1**			**1**
$\overline{B}$	**1**		**1**	**1**
	$\overline{C}$	C	C	$\overline{C}$

$$F_{MDF}(A,B,C) = \overline{A} \cdot \overline{B} + \overline{C}$$

Primer 3. Primenom Karnoove mape odrediti minimalnu disjunktivnu (MDF) funkcije F koja je zadata kao zadata na sledeći način:

$$F(A,B,C) = \overline{A}B\overline{C} + A\overline{C} + BC$$

Rešenje:

A \ BC	00	01	11	10
0			1	1
1	1		1	1

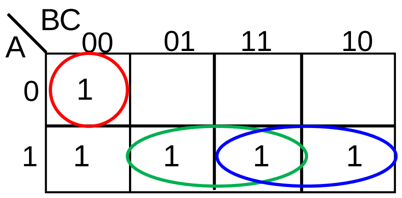
	A	A	$\overline{A}$	$\overline{A}$
B	**1**	**1**	**1**	**1**
$\overline{B}$	**1**			
	$\overline{C}$	C	C	$\overline{C}$

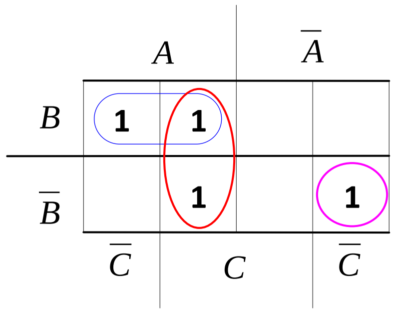
$$F_{MDF}(A,B,C) = B + A\overline{C}$$

Primer 4. Primenom Karnoove mape izvršiti minimizaciju funkcije F minimalnom disjunktivnom formulom **(MDF),** ako je funkcije F zadata izrazom:

$$F(A,B,C)=\overline{A}\,\overline{B}\,\overline{C}+A\overline{B}C+AB+AC$$

Rešenje:



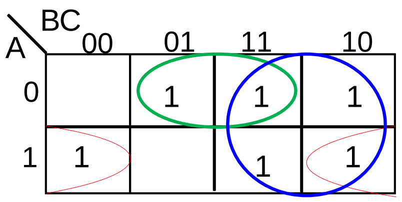


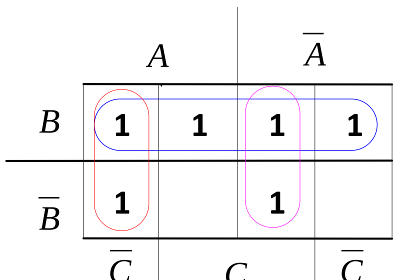
$$F(A,B,C)=AB+AC+\overline{A}\,\overline{B}\,\overline{C}$$

Primer 5. Primenom Karnoove mape izvršiti minimizaciju funkcije F minimalnom disjunktivnom formulom **(MDF),** ako je funkcije F zadata izrazom:

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

Rešenje:





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

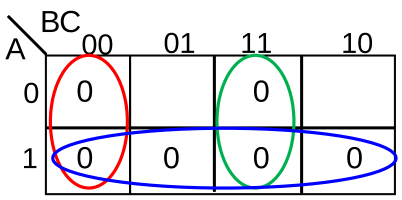
Primer 6. Primenom Karnoove mape odrediti minimalnu konjuktivnu formu (MKF) funkcije Y koja je zadata kao proizvod potpunih suma:

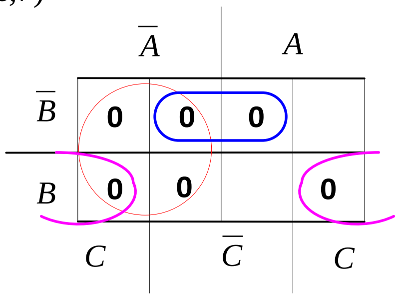
$$Y(A,B,C)=(\overline{A}+B+C)\cdot(\overline{A}+\overline{B}+C)\cdot(\overline{A}+\overline{B}+\overline{C})\cdot(\overline{A}+B+\overline{C})\cdot(A+\overline{B}+\overline{C})\cdot(A+B+C)$$

Rešenje:

Izraz za funkciju Y sastoji se od proizvoda potpunih suma, odnosno može da se predstavi u obliku:

$$Y(A,B,C)=\prod(0,3,4,5,6,7)$$



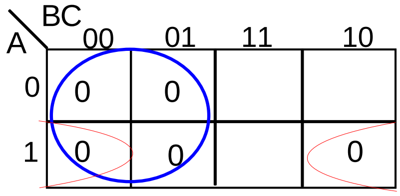


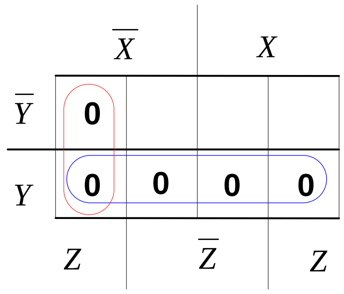
$$Y_{MKF}(A,B,C)=\bar{A}\cdot(\bar{B}+\bar{C})\cdot(B+C)$$

Primer 7. Primenom Karnoove mape odrediti minimalnu konjuktivnu formu (MKF) funkcije F koja je zadata kao suma potpunih proizvoda:

$$F(X,Y,Z)=\prod(0,1,4,5,6)$$

Rešenje:





$$F_{MKF}(X,Y,Z)=Y\cdot(\overline{X}+Z)$$

Primer 8. Primenom pravila Bulove algebre transformisati izraz za funkciju *X(A, B, C)* tako da se dobije zapis u disjunktivnoj normalnoj formi, a zatim odrediti izraz za minimalnu disjunktivnu formu primenom Karnoove mape.

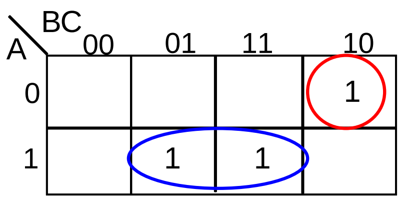
$$X(A,B,C)=\overline{A}B\overline{C}+AC[\overline{B}+A(B+\overline{C})]$$

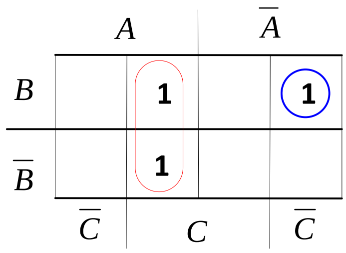
Rešenje:

Izraz za funkciju *X(A, B, C)* transformiše se na sledeći način:

$$X(A,B,C)=\overline{A}\cdot B\cdot\overline{C}+A\cdot C\cdot[\overline{B}+A\cdot(B+\overline{C})]=\overline{A}\cdot B\cdot\overline{C}+A\cdot C\cdot[\overline{B}+A\cdot B+A\cdot\overline{C}]=$$
$$=\overline{A}\cdot B\cdot\overline{C}+A\cdot C\cdot\overline{B}+A\cdot C\cdot A\cdot B+A\cdot C\cdot A\cdot\overline{C}=\overline{A}\cdot B\cdot\overline{C}+A\cdot\overline{B}\cdot C+A\cdot B\cdot C$$

Odavde sledi da funkcija *X(A,B,C)* može da se predstavi kao suma potpunih proizvoda, odnosno u disjunktivnoj normaloj formi: $X_{DNF}(A,B,C)=\sum(2,5,7)$. Na osnovu toga se popunjava Karnoova mapa.





$$X_{MDF}(A,B,C)=\overline{A}\cdot B\cdot\overline{C}+A\cdot C$$

Primer 9. Data je logička funkcija

$$F=\overline{A}\cdot C\cdot D+\overline{B}\cdot\overline{C}\cdot\overline{D}+\overline{A}\cdot\overline{B}\cdot C+\overline{A}\cdot\overline{B}\cdot\overline{C}\cdot D+A\cdot\overline{B}\cdot C\cdot\overline{D}$$

Koristeći Karnoovu mapu pronaći minimalnu disjunktivnu formu ove funkcije

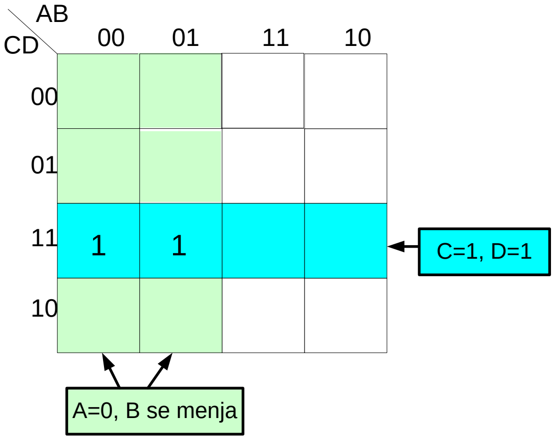
Rešenje:

Da se podsetimo:
Karnoovu mapa 4x4 može da se, na primer, formira tako što će se po horizontali unositi promenljive (ulazi) A i B, a po vertikali promenljive (ulazi) C i D. Raspored promenljivih koje će se naći po horizontali i vertikali može da bude potpuno drugačiji i svaki je podjednako dobar (na primer, moguće je da po horizontali budu ulazi A i D, a po vertikali B i C ili bilo koja druga kombinacija). Ipak, praktično je usvojiti jedan od mogućih rasporeda i držati ga se pri svakom korišćenju Karnoovih mapa jer je tada manja mogućnost greške prilikom raspoređivanja manja. Na primer, redosled kombinacija koji se najčešće koristi je da prvu kolonu određuje kombinacija AB=00, za njom AB=01, zatim AB=11 (primetite da ovde ne može biti kombinacija 10, koja bi išla po binarnom redosledu) i na kraju AB=10. Ako usvojimo isti redosled kombinacija i za ulaze C i D po vertikali, dobijamo da prvu vrstu određuje kombinacija CD=00, drugu, CD=01, zatim CD=11 i na kraju CD=10. Tako dobijamo sledeću tabelu:

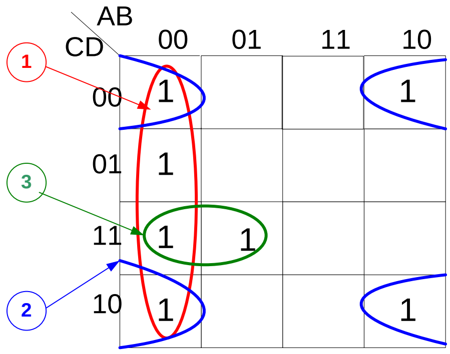
CD \ AB	00	01	11	10
00	0	4	12	8
01	1	5	13	9
11	3	7	15	11
10	2	6	14	10

Kod određivanja koje će se kombinacije ulaza A i B naći jedna pored druge po horizontali, jedino što je bitno je da se dve susedne kombinacije razlikuju u samo jednom ulazu. Na primer, pored kombinacije AB=01 se može naći kombinacija AB=00 (jer je A nula, a samo B se razlikuje, jer je u prvoj kombinaciji jedinica, u drugoj nula). Pored ove kombinacije (AB=01) se takođe može naći kombinacija AB=11 (samo je A različito, B je jedinica u obe). Međutim, kombinacije AB=01 i AB=10 ne smeju biti susedne jer se i A i B razlikuju.

Izraz za logičku funkcija koji je dat u zadatku je očigledno u disjunktivnoj formi. Na osnovu postojećih logičkih proizvoda u izrazu funkcije, može da se zaključi koje grupe polja u Karnoovoj mapi treba da sadrži vrednost 1. Grupa jedinica (dve jedinice) koja odgovaraje prvom članu $\overline{A} \cdot C \cdot D$ nalaze se u preseku polja u kojima je ulaz A na nuli, ulaz B je bilo nula ili jedinica (B ne učestvuje u članu), a oba ulaza C i D su na jedinici



Ako se pođe od činjenice da svako polje u Karnoovoj mapi odgovara tačno određenom potpunom logičkom proizvodu za koji funkcija predstavljena u disjunktivnoj normalnoj formi ima vrednost 1, do pozicije grupe polja koja odgovaraju članu $\overline{A}\cdot C\cdot D$ može se doći i analitički, odnosno proširivanjem nepotpunog logičkih proizvoda: $\overline{A}CD = \overline{A}(B+\overline{B})CD = \overline{A}BCD + \overline{A}\overline{B}CD$, što znači da grupu čine polja gde je A,B,C,D = 0,1,1,1 i A,B,C,D = 0,0,1,1. Na sličan način dolazimo do ostalih grupa. Grupisanjem polja u Karnoovoj mapi, dobija se izraz za funkciju u minimalnoj disjunktivnoj formi, koji je mnogo jednostavniji od početne forme:



$$F = \underline{\overline{B}\cdot\overline{D}} + \underline{\overline{A}\cdot C\cdot D} + \underline{\overline{A}\cdot\overline{B}}$$

Grupa ① nastala je od jedinica za sledeće ulaze:

A	B	C	D
0	0	0	0
0	0	0	1
0	0	1	1
0	0	1	0
$\overline{A}$	$\overline{B}$	-	-

Ulaz A je za celu grupu na nuli, ulaz B je za celu grupu na nuli, a ulazi B i D se menjaju. Po pravilima Karnoovih mapa, ulazi koji se menjaju ispadaju iz člana, ulazi koji su stalno na jedinici ulaze u član direktno, a ulazi koji su stalno na nuli ulaze u član kao invertovani. Član koji potiče od ove grupe je $\overline{A}\cdot\overline{B}$.

Grupu ② čine četiri logičke jedinice funkcije F:

A	B	C	D
0	0	0	0
0	0	1	0
1	0	0	0
1	0	1	0
-	$\overline{B}$	-	$\overline{D}$

Ulaz B je za celu grupu na nuli, ulaz D je za celu grupu na nuli, a ulazi A i C se menjaju. Član koji potiče od ove grupe je: $\overline{B}\cdot\overline{D}$.

Grupa označena sa ③ nastala je od jedinica za sledeće ulaze:

A B C D
0 0 1 1
0 1 1 1

$\overline{A}$ - C D

Ulaz A je za celu grupu na nuli, ulazi C i D su na jedinici, a ulaz B se menja. Tako član koji potiče od ove grupe ima tri elementa:. $\overline{A} \cdot C \cdot D$

Primer 10. Niz **1000 000x 1x10 1110** predstavlja izlaze funkcije četiri promenljive A,B,C,D pri čemu prva cifra u nizu predstavlja izlaz kada je A,B,C,D =0,0,0,0, a poslednja, kada je A,B,C,D =1,1,1,1. Cifre izmedju prve i poslednje su izlazi po binarnom redu. **x** je neodređen izlaz koji može biti bilo nula ili jedinica. Napisati logički izraz koji predstavlja:

a) Minimalnu disjunktivnu formu ove funkcije.
b) Minimalnu konjunktivnu formu ove funkcije.

Rešenje:

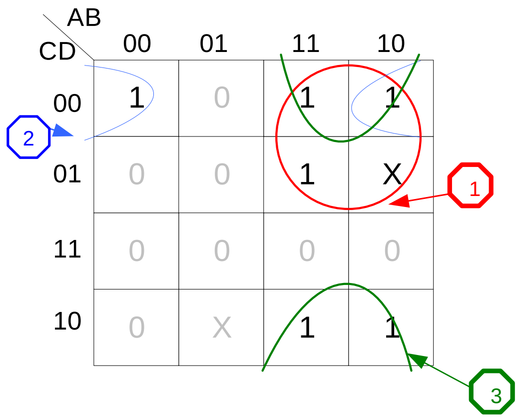
Zadati niz mogao bi da se predstavi pomoću tabele istinitosti. U tabelu se upisuju vrednosti funkcije F (0 ili 1) u polja koja određuju ulazi. Tako logičku jedinicu koja je vrednost funkcije F u prvom redu označenom brojem u tabeli istinitosti, treba upisati na mestu gde su ulazi ABCD=0000, tačnije u preseku kolone gde je AB=00 i vrste gde je CD=00. Logičku jedinicu koja je vrednost funkcije F u osmom redu u tabeli istinitosti, treba upisati na mestu gde su ulazi ABCD=1000, tačnije u preseku kolone gde je AB=10 i vrste gde je CD=00 i tako dalje.

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

CD \ AB	00	01	11	10
00	0		12	8
01			13	
11				
10			14	10

a) **Minimalna disjunktivna forma** određuje se grupisanjem polja u kojima je upisana vrednost 1, a nedefinisane vrednosti izlaza (označene sa X) treba uzeti kao jedinice samo ako pomažu formiranju veće grupe. Tako nedefinisanu vrednost izlaza na mestu gde su ulazi ABCD=1001 treba usvojiti kao jedinicu, jer nam pomaže da formiramo grupu četiri jedinice, označenu sa ① na sledećoj tabeli.

Koristeći pomenuta pravila moguće je formirati tri grupe, dve sa po četiri jedinice i jednu sa dve jedinice. Grupe se preklapaju, jedna jedinica (u gornjem desnom uglu tabele) je obuhvaćena čak sa sve tri grupe što ne predstavlja nedostatak. Grupe su označene na sledećoj tabeli:



$$F = A\overline{C} + A\overline{D} + \overline{B}\,\overline{C}\,\overline{D}$$

Grupu označenu sa ① čine četiri logičke jedinice funkcije F. Ova grupa je nastala od jedinica za sledeće ulaze:

A	B	C	D
1	1	0	0
1	0	0	0
1	1	0	1
1	0	0	1
A	-	$\mathbf{\overline{C}}$	-

Ulaz A je za celu grupu na jedinici, ulaz C je za celu grupu na nuli, a ulazi B i D se menjaju. Po pravilima Karnoovih mapa, ulazi koji se menjaju ispadaju iz člana, ulazi koji su stalno na jedinici ulaze u član direktno, a ulazi koji su stalno na nuli ulaze u član kao invertovani. Tako član koji potiče od ove grupe je $\mathbf{A\overline{C}}$.

Grupu označenu sa ② koja spaja dve jedinicu iz prve kolone sa jedinicom poslednje kolone ("spojena s druge strane papira") čine dve logičke jedinice funkcije F:
Ova grupa je nastala od jedinica za sledeće ulaze:

$$\begin{array}{cccc} \mathbf{A} & \mathbf{B} & \mathbf{C} & \mathbf{D}\\ 0 & 0 & 0 & 0\\ 1 & 0 & 0 & 0\\ \hline - & \overline{\mathbf{B}} & \overline{\mathbf{C}} & \overline{\mathbf{D}} \end{array}$$

Ulaz B je za celu grupu na jedinici, ulazi C i D takođe, a ulaz B se menja. Tako član koji potiče od ove grupe ima tri elementa: $\overline{\mathbf{B}}\overline{\mathbf{C}}\overline{\mathbf{D}}$.

Grupu označenu sa ③ koja spaja dve jedinice iz prve vrste sa dve jedinice poslednje vrste ("spojena s druge strane papira") čine četiri logičke jedinice funkcije F:

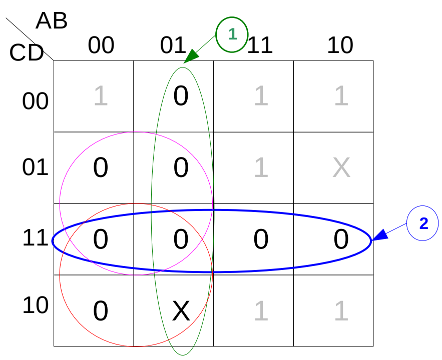
$$\begin{array}{cccc} \mathbf{A} & \mathbf{B} & \mathbf{C} & \mathbf{D}\\ 1 & 1 & 0 & 0\\ 1 & 0 & 0 & 0\\ 1 & 1 & 1 & 0\\ 1 & 0 & 1 & 0\\ \hline \mathbf{A} & - & - & \overline{\mathbf{D}} \end{array}$$

Ulaz A je za celu grupu na jedinici, ulaz D je za celu grupu na nuli, a ulazi B i C se menjaju. Po pravilima Karnoovih mapa, ulazi koji se menjaju ispadaju iz člana, ulazi koji su stalno na jedinici ulaze u član direktno, a ulazi koji su stalno na nuli ulaze u član kao invertovani. Tako član koji potiče od ove grupe je $\mathbf{A}\overline{\mathbf{D}}$.

b) Minimalna konjunktivna forma je forma logičkog proizvoda članova koji su logički zbirovi elemenata. Elementi su ili same ulazne promenljive ili njihove invertovane vrednosti.

Da bi se dobila ova forma pomoću Karnoovih mapa potrebno je grupisati nule u grupe po istom principu kao što su grupisane jedinice za dobijanje minimalne disjunktivne forme. Razlog ovakvog grupisanja i pravila su objašnjena u udžbeniku.

Na slici je prikazan jedan od mogućih načina grupisanja polja u Karnoovoj mapi. Neodređeni izlaz **x** na poziciji ABCD=0110 je usvojen kao nula da bi grupa ① mogla da ima četiri ulaza.



$$F = (A+\overline{B})(A+\overline{D})(A+\overline{C})(\overline{C}+\overline{D})$$

Minimalan broj grupa je četiri i svaka od grupa ima po četiri nule. Kao i u slučaju minimalne disjunktivne forme grupa koja obuhvata dve nule smanjuje, broj elemenata u tom članu za jedan, grupa od četiri nule smanjuje broj elemenata za dva i tako dalje. Za funkciju četiri ulaza grupa od četiri nule će dati član sa dva elementa. Svaka grupa čini jedan član u minimalnoj konjunktivnoj formi. Određivanje koji elementi čine član je po principu slično istom postupku u slučaju disjunktivne forme. Ovde će biti prikazano samo za jednu grupu označenu sa ②**.**

Grupu označenu sa ② (puna linija) čine četiri logičke nule funkcije F:

A	**B**	**C**	**D**
0	0	1	1
0	1	1	1
1	1	1	1
1	0	1	1
-	-	$\overline{\mathbf{C}}$	$\overline{\mathbf{D}}$

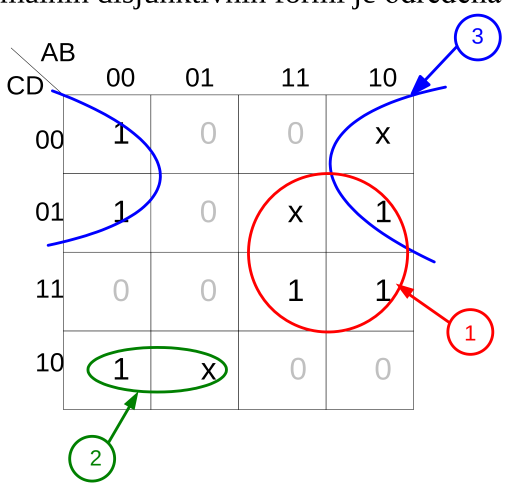
Ulaz C je za celu grupu na jedinici, ulaz D takođe, ulazi A i B se menjaju. Po pravilima Karnoovih mapa za minimalnu konjunktivnu formu, ulazi koji se menjaju ispadaju iz člana, ulazi koji su stalno na nuli ulaze u član direktno, a ulazi koji su stalno na jedinici ulaze u član kao invertovani. Tako član koji potiče od ove grupe je $\overline{\mathbf{C}}+\overline{\mathbf{D}}$.
Postupak određivanja ostalih članova je isti. Funkcija koja se na ovaj način dobija kao minimalna disjunktivna forma je sledeća (članovi su podvučeni linijom kakvom je označena odgovarajuća grupa)

<u>Primer 11.</u> Niz **1110 00x0 x101 0x01** predstavlja izlaze funkcije četiri promenljive A,B,C,D pri čemu prva cifra u nizu predstavlja izlaz kada je A,B,C,D=0,0,0,0, a poslednja, kada je A,B,C,D=1,1,1,1. Cifre izmedju prve i poslednje su izlazi po binarnom redu. **x** je nedefinisan izlaz, svaki od njih pojedinačno može biti bilo nula ili jedinica. Napisati logički izraz koja predstavlja:

a) Minimalnu disjunktivnu formu ove funkcije.
b) Minimalnu konjunktivnu formu ove funkcije.

Rešenje:

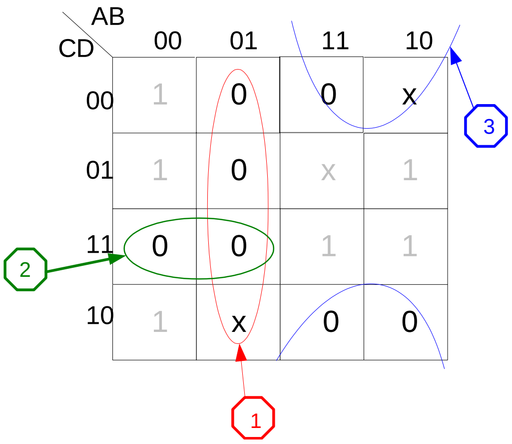
a) Jedna od mogućih minimalnih disjunktivnih formi je određena na osnovu tabele:



$$F = AD + \overline{B}\,\overline{C} + \overline{A}C\overline{D}$$

① ③ ②

b) Minimalna konjunktivna forma može da se odredi pomoću iste Karnoove mape, grupisanjem polja u kojima se nalaze nule.

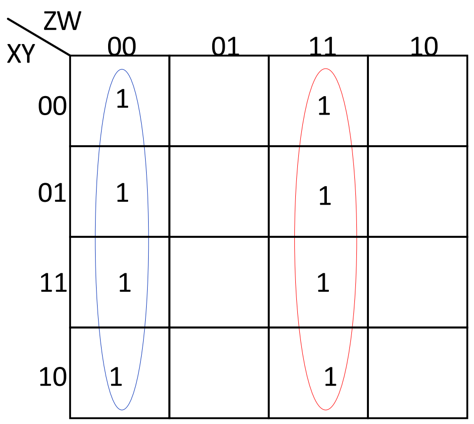


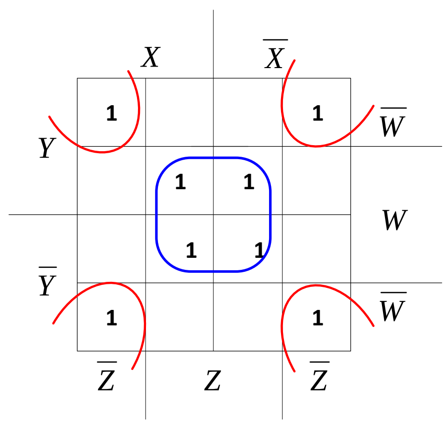
$$F = \underbrace{(A+\overline{B})}_{①}\underbrace{(A+\overline{C}+\overline{D})}_{②}\underbrace{(\overline{A}+D)}_{③}$$

<u>Primer 12.</u> Primenom Karnoove mape odrediti minimalnu disjunktivnu formu **(MDF)** funkcije F koja je zadata kao suma potpunih proizvoda:

$$F(X, Y, Z, W) = \Sigma\ (0,3,4,7,8,11,12,15)$$

Rešenje:



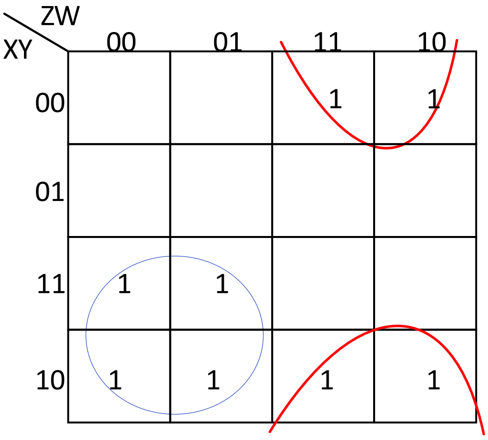


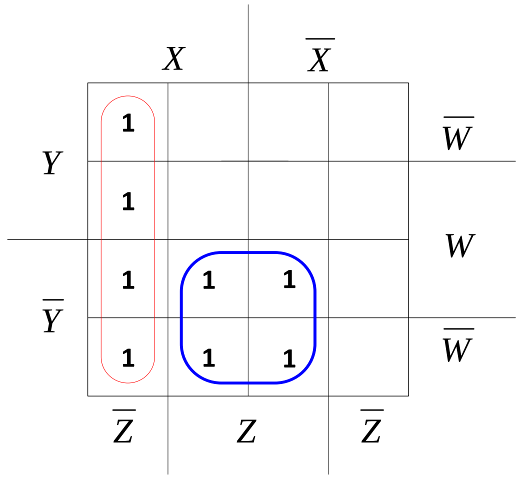
$$F_{MDF}(X,Y,Z,W) = ZW + \overline{Z}\,\overline{W}$$

<u>Primer 13.</u> Primenom Karnoove mape odrediti minimalnu disjunktivnu formu **(MDF)** funkcije F koja je zadata kao proizvod potpunih suma (DNF):

$$F(X, Y, Z, W) = \Sigma\ (2,3,8,9,10,11,12,13)$$

Rešenje:



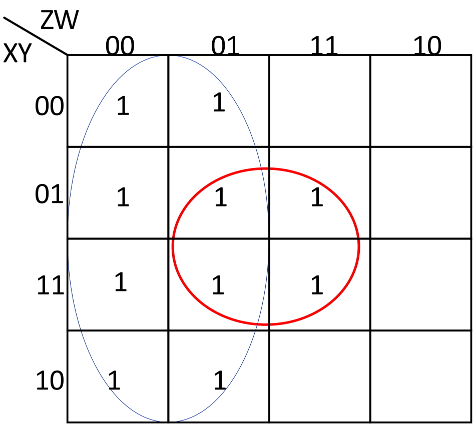


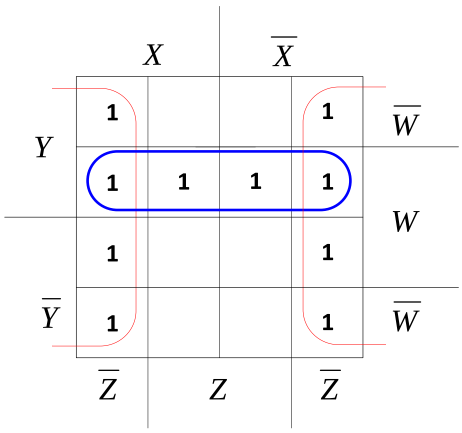
$$F_{MDF}(X, Y, Z, W) = X\bar{Z} + \bar{Y}Z$$

Primer 14. Primenom Karnoove mape odrediti minimalnu disjunktivnu formu **(MDF)** funkcije F koja je zadata kao suma potpunih proizvoda:

$$F(X, Y, Z, W) = \Sigma\ (15, 13, 12, 9, 8, 7, 5, 4, 1, 0)$$

Rešenje:



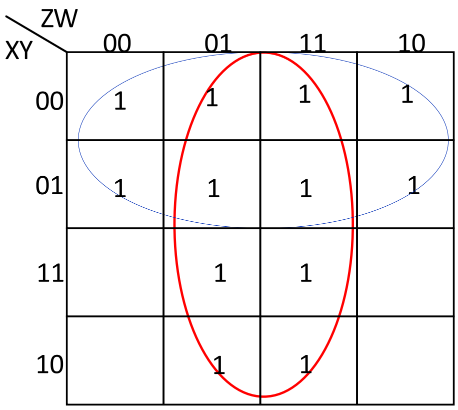


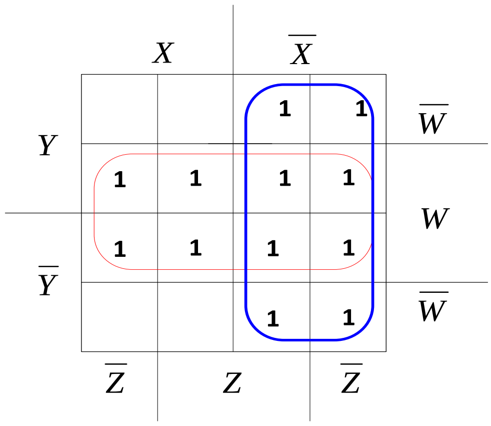
$$F_{MDF}(X, Y, Z, W) = \bar{Z} + YW$$

Primer 15. Primenom Karnoove mape izvršiti minimizaciju funkcije F minimalnom disjunktivnom formulom **(MDF),** ako je funkcije F zadata na sledeći način:

$$F(X, Y, Z, W) = YW + \overline{X}Z + X\overline{Y}W + \overline{X}Y\overline{Z} + \overline{X}\,\overline{Y}\,\overline{Z}$$

Rešenje:



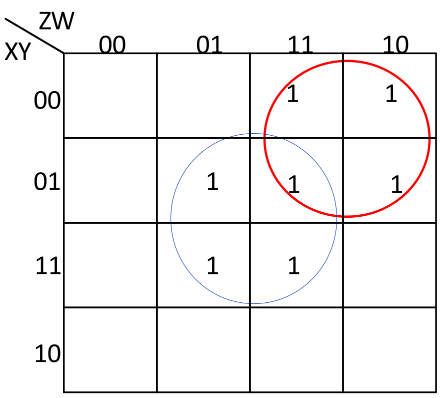


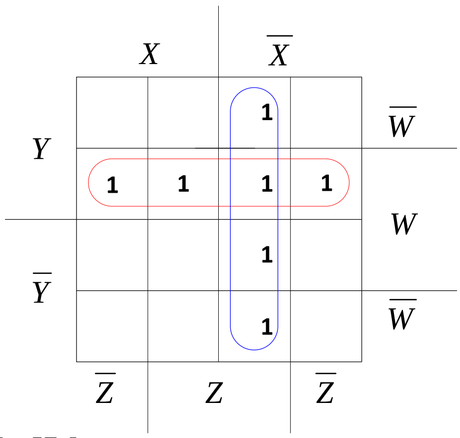
$$F_{MDF}(X,Y,Z,W) = \overline{X} + W$$

Primer 16. Primenom Karnoove mape izvršiti minimizaciju funkcije F minimalnom disjunktivnom formulom **(MDF),** ako je funkcije F zadata na sledeći način:

$$F(X, Y, Z, W) = XYW + \overline{X}Z + \overline{X}Y\overline{Z}W$$

Rešenje:





$$F_{MDF}(X,Y,Z,W) = \overline{X}Z + YW$$

Primer 17. Za funkciju:

$$F(A,B,C,D) = A\cdot\overline{C} + B\cdot C\cdot D + \overline{A}\cdot\overline{C}\cdot D + \overline{C}\cdot\overline{D} + A\cdot B\cdot\overline{C}\cdot D$$

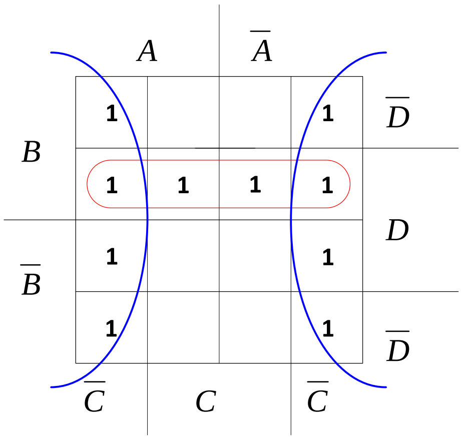
a) Odrediti minimalnu disjunktivnu formu funkcije (MDF) primenom Karnoove mape.

b) Odrediti logičku vrednost funkcije ako se na ulaz A dovede niz logičkih nivoa 01010111, na ulaz B: 00111001, na ulaz C: 01010101 i na ulaz D: 11001010.

c) Nacrtati šemu kombinacione mreže kojom se realizuje MDF date funkcije F, koristeći samo dva **NE** kola i dva **I** kola.

Rešenje:

a)



$$F_{MDF}(A,B,C,D)=\overline{C}+B\cdot D$$

b)

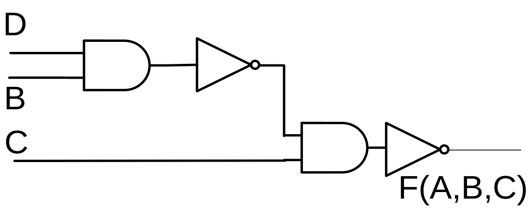
A	B	C	D	$\overline{C}$	$B\cdot D$	$\overline{C}+B\cdot D$
0	0	0	1	1	0	**1**
1	0	1	1	0	0	**0**
0	1	0	0	1	0	**1**
1	1	1	0	0	0	**0**
0	1	0	1	1	1	**1**
1	0	1	0	0	0	**0**
1	0	0	1	1	0	**1**
1	1	1	0	0	0	**0**

F(A,B,C,D)=10101010

c)

$$F(A,B,C,D)=\overline{C}+B\cdot D$$

$$F(A,B,C,D)=\overline{\overline{\overline{C}+B\cdot D}}=\overline{C\cdot\overline{B\cdot D}}$$



Primer 18. Za funkciju zadatu logičkim izrazom:

$$F(A,B,C,D)=A\cdot D+\overline{B}\cdot C+\overline{A}\cdot B\cdot\overline{C}\cdot D+B\cdot C$$

a) Odrediti sekvencu vrednosti, ako se na ulaz A dovede niz logičkih nivoa 10100011, na ulaz B: 11010101, na ulaz C: 10101000 i na ulaz D: 10101011.

b) Odrediti izraz za minimalnu disjunktivnu formu (MDF) primenom Karnoove mape.

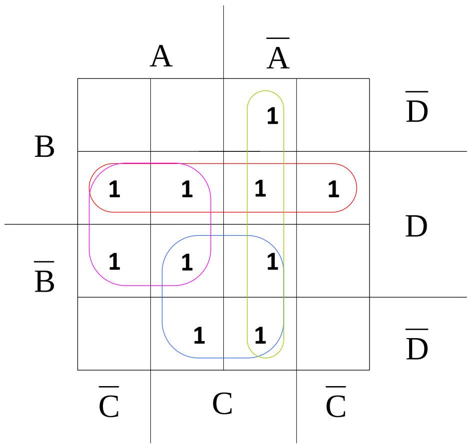
Rešenje:

a)

A	B	C	D	A+B	D(A+B)	$\overline{A}+\overline{B}$	$C(\overline{A}+\overline{B})$	F(A,B,C,D)
1	1	1	1	1	1	0	0	**1**
0	1	0	0	1	0	1	0	**0**
1	0	1	1	1	1	1	1	**1**
0	1	0	0	1	0	1	0	**0**
0	0	1	1	0	0	1	1	**1**
0	1	0	0	1	0	1	0	**0**
1	0	0	1	1	1	1	0	**1**
1	1	0	1	1	1	0	0	**1**

F(A,B,C,D)=10101011

b)

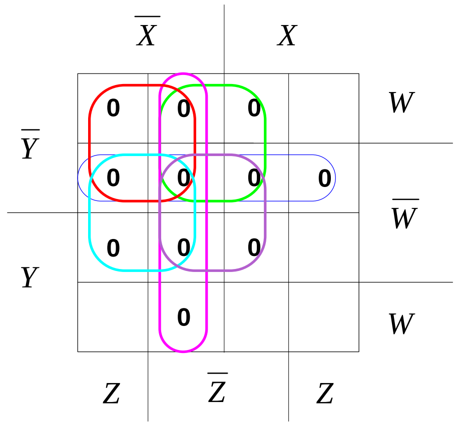


$$F(A,B,C,D) = AD + BD + C\overline{B} + C\overline{A} = D(A+B) + C(\overline{A}+\overline{B})$$

<u>Primer 19.</u> Primenom Karnoove mape odrediti minimalnu konjuktivnu formu **(MKF)** funkcije F koja je zadata kao suma potpunih proizvoda (KNF):

$$F(X, Y, Z, W) = \Pi\ (3, 5, 6, 7, 9, 10, 11, 12, 13, 14, 15)$$

Rešenje:

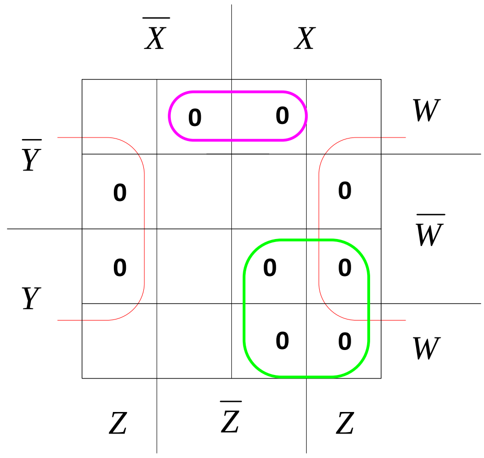


$$F_{MKF}(X,Y,Z,W) = (\overline{Z}+\overline{W})\cdot(\overline{X}+\overline{W})\cdot(\overline{Y}+\overline{W})\cdot(\overline{X}+\overline{Z})\cdot(\overline{X}+\overline{Y})\cdot(\overline{Y}+\overline{Z})$$

<u>Primer 20.</u> Primenom Karnoove mape odrediti minimalnu konjuktivnu formu **(MKF)** funkcije F koja je zadata kao proizvod potpunih suma:

$$F(X, Y, Z, W) = \Pi\,(0,1,2,3,5,6,9,13,14)$$

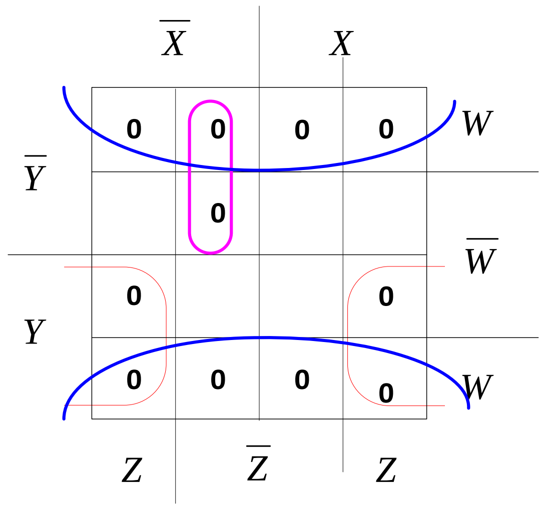
Rešenje:



$$F_{MKF}(X,Y,Z,W) = (X+Y)\cdot(Z+\overline{W})\cdot(\overline{Y}+\overline{Z}+W)$$

<u>Primer 21.</u> Primenom Karnoove mape izvršiti minimizaciju funkcije F minimalnom konjuktivnom formulom **(MKF),** ako je funkcija $F(X,Y,Z,W)$ zadata kao F(0001010100010100), pri čemu prva cifra u nizu predstavlja izlaz kada je X,Y,Z,W=0,0,0,0, a poslednja, kada je X,Y,Z,W=1,1,1,1.

Rešenje:

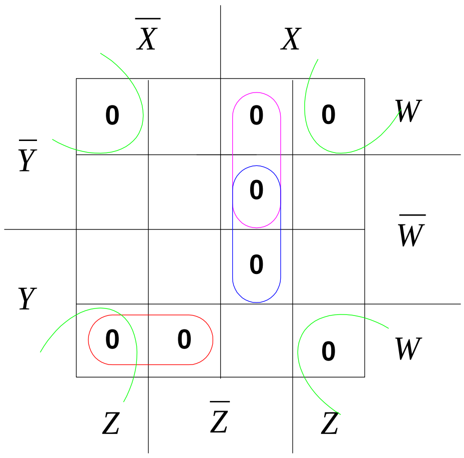


$$F_{MKF}(X,Y,Z,W) = Z \cdot Y + W + \overline{X} \cdot \overline{Y} \cdot \overline{Z}$$

<u>Primer 22.</u> Primenom Karnoove mape odrediti minimalnu konjuktivnu formu **(MKF)** funkcije F koja je zadata kao proizvod potpunih suma:

$$F(X, Y, Z, W) = \Pi\ (0,3,4,6,7,8,10,12)$$

Rešenje:

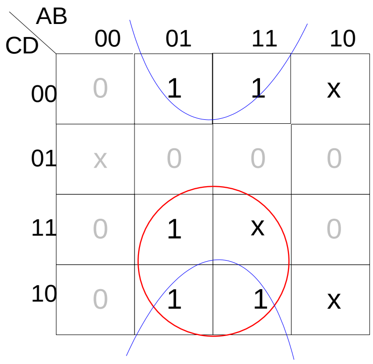


$$F_{MKF}(X,Y,Z,W) = (Z+W)\cdot(\overline{X}+Y+W)\cdot(X+\overline{Y}+\overline{Z})\cdot(X+\overline{Z}+\overline{W})$$

<u>Primer 23.</u> Niz **0x00 1011 10x0 101x** predstavlja izlaze funkcije četiri promenljive A,B,C,D pri čemu prva cifra u nizu predstavlja izlaz kada je A,B,C,D=0,0,0,0, a poslednja, kada je A,B,C,D=1,1,1,1. Cifre izmedju prve i poslednje su izlazi po binarnom redu. **x** je nedefinisan izlaz, svaki od njih pojedinačno može biti bilo nula ili jedinica. Napisati logički izraz koja predstavlja:
a) Minimalnu disjunktivnu formu ove funkcije.
b) Minimalnu konjunktivnu formu ove funkcije.

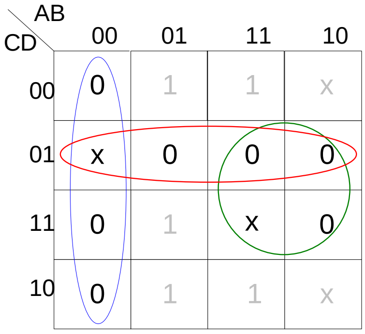
Rešenje:

a) Minimalna disjunktivna forma:



$$F = C \cdot B + B \cdot \overline{D}$$

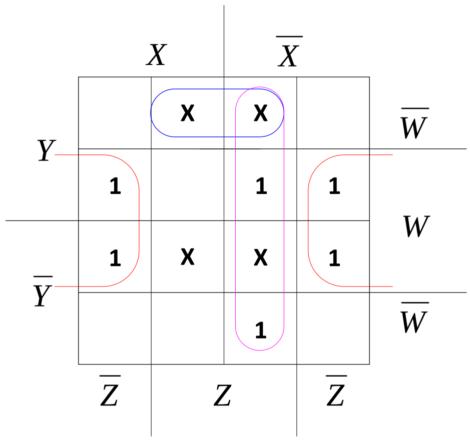
b) Minimalna konjunktivna forma



$$F = (C + \overline{D})(\overline{A} + \overline{D})(A + B)$$

Primer 24. Za datu funkciju četiri logičke promenljive $F(X,Y,Z,W)$, čiji je skup vrednosti $F(0,1,1,x,0,1,x,1,0,1,0,x,0,1,x,0)$, primenom Karnoovih mapa odrediti njenu minimalnu disjunktivnu formulu **(MDF)**, ako x predstavlja nedefinisano stanje funkcije.

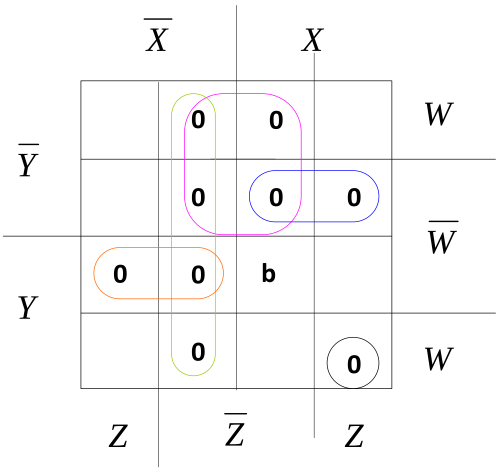
Rešenje:



$$F_{MDF}(X,Y,Z,W) = \overline{X}Z + \overline{Z}W + YZ\overline{W}$$

Primer 25. Primenom Karnoove mape izvršiti minimizaciju funkcije F minimalnom konjuktivnom formulom **(MKF),** ako je funkcija $F(X,Y,Z,W)$ zadata u formi F(011b100010001100), a sa **b** su označena nedefinisana stanja funkcije:

Rešenje:

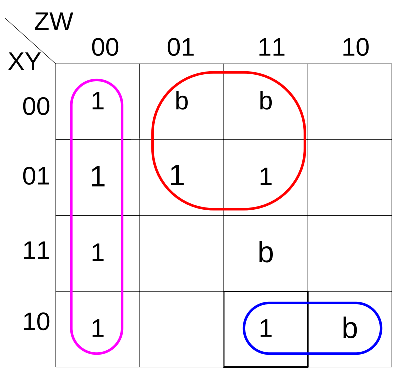


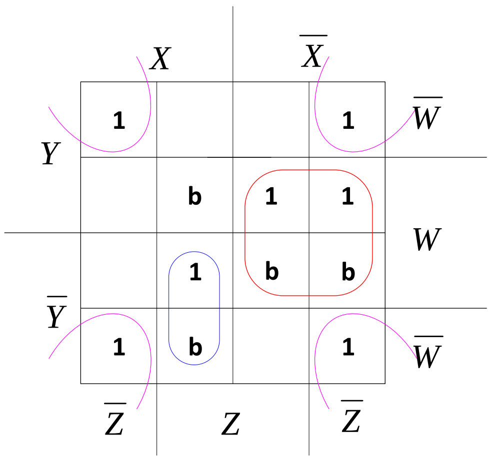
$$F_{MKF}(X,Y,Z,W) = (\overline{X}+\overline{Z})\cdot(\overline{Y}+\overline{Z})\cdot(X+\overline{Y}+\overline{W})\cdot(\overline{X}+Y+\overline{W})\cdot(X+Y+Z+W)$$

<u>Primer 26.</u> Primenom Karnoove mape odrediti minimalnu disjunktivnu formu **(MDF)** funkcije $F(X,Y,Z,W)$ čije su vrednosti date u sledećoj tabeli, a sa **b** su označena nedefinisana stanja funkcije.

i	X	Y	Z	W	F
0	0	0	0	0	1
1	0	0	0	1	b
2	0	0	1	0	0
3	0	0	1	1	b
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	1
9	1	0	0	1	0
10	1	0	1	0	b
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	b

Rešenje:





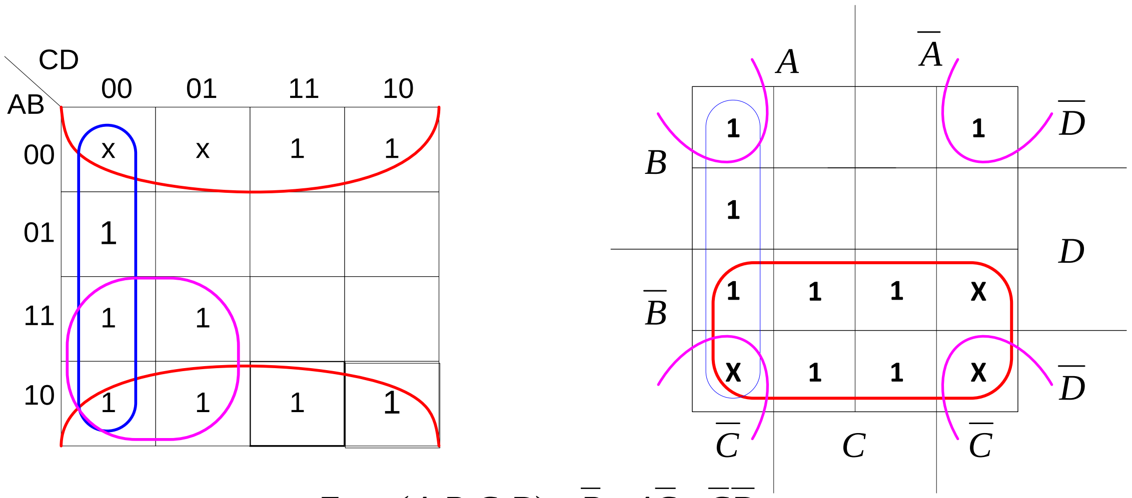
$$F_{MDF}(X,Y,Z,W) = \overline{X}W + \overline{Z}\,\overline{W} + X\overline{Y}Z$$

Primer 27. Primenom Karnoove mape odrediti minimalnu disjunktivnu formu **(MDF)** funkcije *F*(*A*,*B*,*C*,*D*) čije su vrednosti date u tabeli:

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

Sa x su označena nedefinisana stanja funkcije i po potrebi se grupišu sa jedinicama za nalaženje minimalne disjunktivne forme

Rešenje:



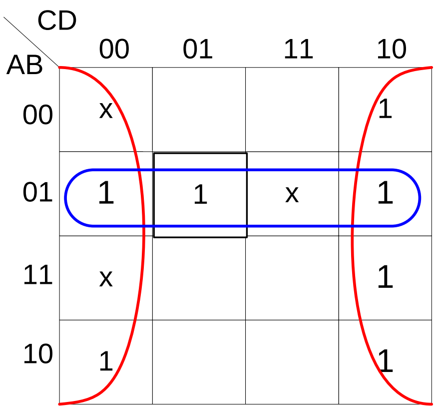
$$F_{MDF}(A,B,C,D) = \overline{B} + A\overline{C} + \overline{C}\,\overline{D}$$

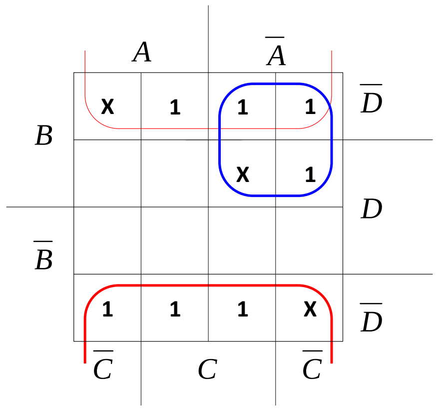
Primer 28. Primenom Karnoove mape odrediti minimalnu disjunktivnu formu **(MDF)** funkcije *F*(*A*,*B*,*C*,*D*) čije su vrednosti date u tabeli:

i	A	B	C	D	F
0	0	0	0	0	x
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	1
6	0	1	1	0	1
7	0	1	1	1	x
8	1	0	0	0	1
9	1	0	0	1	0
10	1	0	1	0	1
11	1	0	1	1	0
12	1	1	0	0	x
13	1	1	0	1	0
14	1	1	1	0	1
15	1	1	1	1	0

Sa x su označena nedefinisana stanja funkcije i po potrebi se grupišu sa jedinicama za nalaženje minimalne disjunktivne forme.

Rešenje:



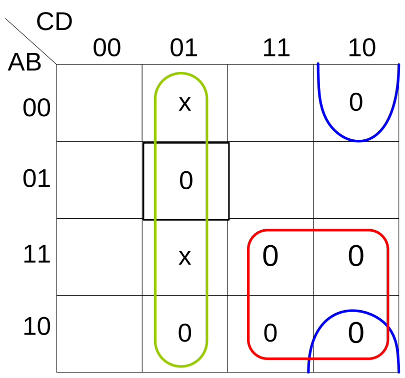


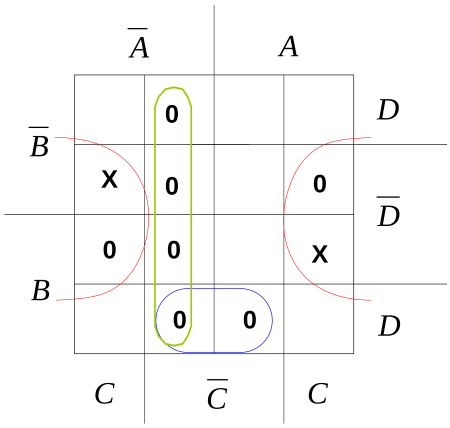
$$F_{MDF}(A,B,C,D) = \overline{D} + \overline{A}B$$

Primer 29. Primenom Karnoove mape odrediti minimalnu konjuktivnu formu **(MKF)** funkcije $F(X,Y,Z,W)$ čije su vrednosti date u sledećoj tabeli, a sa **b** su označena nedefinisana stanja funkcije:

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

Rešenje:



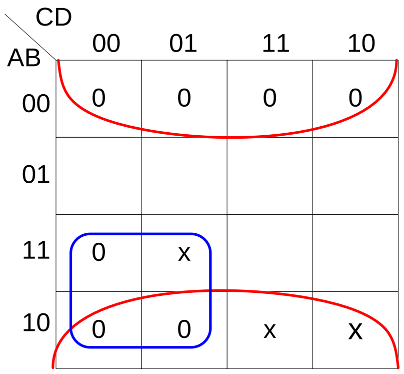


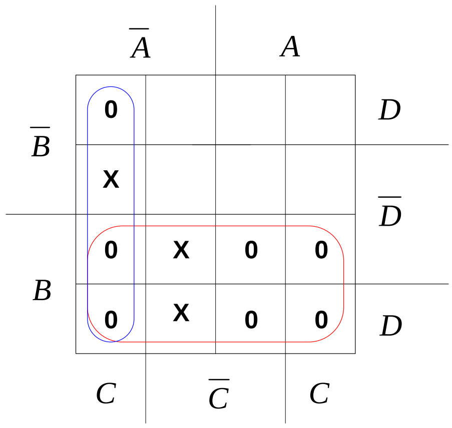
$$F_{MKF}(A,B,C,D) = (\overline{A} + \overline{C}) \cdot (C + \overline{D}) \cdot (B + \overline{C} + D)$$

Primer 30. Primenom Karnoove mape odrediti minimalnu konjuktivnu formu **(MKF)** funkcije $F(A,B,C,D)$ čije su vrednosti date u tabeli, a sa **X** su označena nedefinisana stanja funkcije:

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

Rešenje:



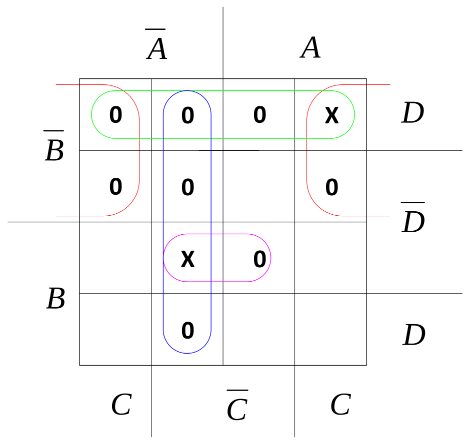


$$F_{MKF}(A,B,C,D) = (\overline{A}+C)\cdot B$$

Primer 31. Primenom Karnoove mape odrediti minimalnu konjunktivnu formu **(MKF)** funkcije *F(A,B,C,D)* čije su vrednosti date u tabeli, a sa **X** su označena nedefinisana stanja funkcije i po potrebi ih treba iskoristiti za nalaženje minimalne forme:

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

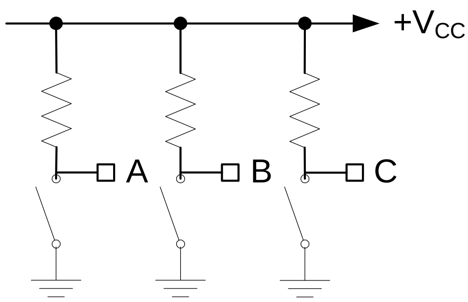
Rešenje:



$$F_{MKF}(A,B,C,D) = (\overline{A}+\overline{C})\cdot(\overline{B}+D)\cdot(\overline{B}+C)\cdot(B+\overline{C}+\overline{D})$$

<u>Primer 32.</u> Na slici su prikazana tri paralelna prekidača. U zavisnosti od toga da li su prekidači otvoreni ili zatvoreni napon u tačkama A, B i C ima vrednost V_{CC} (1) (logička “1”) ili 0 (logička “0”). Ako signali u tačkama A, B i C predstavljaju ulaz u kombinacionu mrežu koja daje na izlazu 1 ako su otvorena bar dva prekidača, inače je vrednost 0:

a) Definisati funkciju OUT(A, B, C) koja opisuje rad kombinacione mreže pomoću tabele istinitosti.
b) Odrediti minimalnu disjunktivnu formu funkcije $OUT_{MDF}(A, B, C)$.

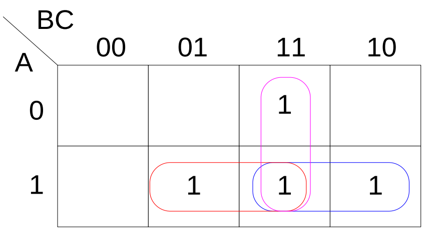


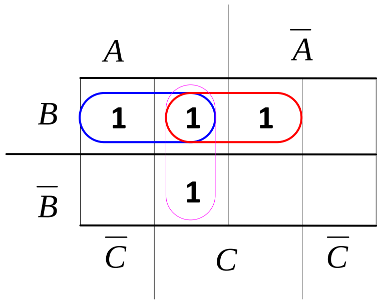
Rešenje:

a) Pošto se na ulaz kombinacione mreže dovode tri signala A, B i C broj kombinacija logičkih vrednosti ovih signala je $2^3 = 8$.

n	A	B	C	OUT
0	0	0	0	0
1	0	0	1	0
2	0	1	0	0
3	0	1	1	1
4	1	0	0	0
5	1	0	1	1
6	1	1	0	1
7	1	1	1	1

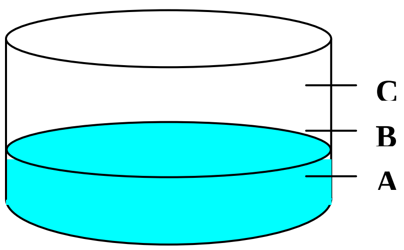
b)





$$OUT_{MDF}(A,B,C) = A \cdot B + A \cdot C + B \cdot C$$

Primer 33. Nivo tečnosti u rezervoaru detektuje se preko tri senzora A, B i C, čiji izlaz ima vrednost logička “1” kada nivo tečnosti dostigne ili premaši poziciju senzora, odnosno logička “0” kada je senzor iznad nivoa. Ako su senzori postavljeni kao što je prikazano na slici:



a) Definisati tabelu istinitosti za funkciju F(C, B, A) koja opisuje rad kombinacione mreže čiji su ulazi signali sa senzora, ako izlaz mreže ima vrednost logička “1” kada je nivo fluida između senzora A i B.

b) Primenom Karnoove mape odrediti minimalnu disjunktivnu formu funkcije F(C, B, A).

Rešenje:

a) Pošto vrednost funkcije F zavisi od tri ulazne promenljive, ukupan broj kombinacija vrednosti ulaznih veličina A, B i C je $2^3 = 8$. Pošto izlaz svakog senzora dobija vrednost logička “1” tek kada nivo dostigne ili premaši poziciju senzora, moguća je sledeća kombinacija vrednosti na izlazima senzora:

C	**B**	**A**
0	0	0
0	0	**1**
0	**1**	**1**
1	**1**	**1**

Funkcija F će imati vrednost 1 samo kada senzor A detektuje nivo (CBA = 001), dok će za ostale tri kombinacije vrednosti izlaza sa senzora imati vrednost 0. Za sve ostale kombinacije vrednosti izlaza senzora funkcija F će imati nedefinisanu vrednost (X). Na osnovu ovog razmatranja popunjena je kombinaciona tabela:

n	C	B	A	F
0	0	0	0	0
1	0	0	1	1
2	0	1	0	X
3	0	1	1	0
4	1	0	0	X
5	1	0	1	X
6	1	1	0	X
7	1	1	1	0

X su nedefinisana stanja

b) Na osnovu tabele iz tačke a) popunjena je Karnoova mapa sa tri promenljive, pri čemu nedefinisana stanja X u cilju bolje minimizacije mogu da se po potrebi grupišu sa poljima u kojima je 1.

C \ BA	00	01	11	10
0	0	1	0	X
1	X	X	0	X

	C		$\overline{C}$	
B	X	0	0	X
$\overline{B}$	X	X	1	0
	$\overline{A}$	A	A	$\overline{A}$

$$F_{MDF}(C,B,A) = \overline{B}A$$

Primer 34. Neka su X, Y i Z binarne cifre u trobitnom zapisu nekog binarnog broja.

a) Definisati kombinacionu tabelu za funkciju *F(X, Y, Z)* koja ima vrednost 0 ako bar dve susedne cifre u zapisu binarnog broja imaju istu vrednost, dok u ostalim slučajevima ima vrednost 1.

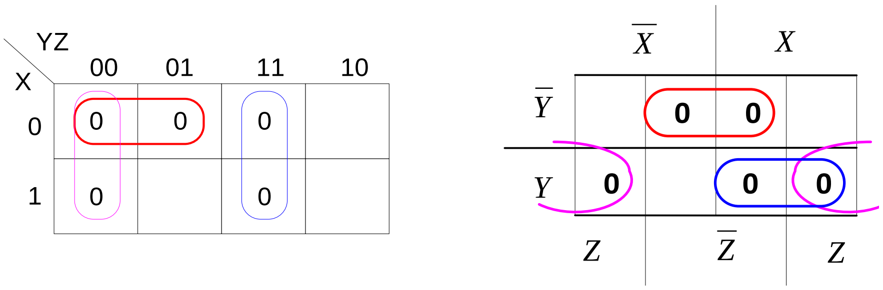
b) Odrediti minimalnu konjunktivnu formu funkcije F primenom Karnoove mape.

Rešenje:

a)

n	X	Y	Z	F
0	0	0	0	0
1	0	0	1	0
2	0	1	0	1
3	0	1	1	0
4	1	0	0	0
5	1	0	1	1
6	1	1	0	1
7	1	1	1	0

b)



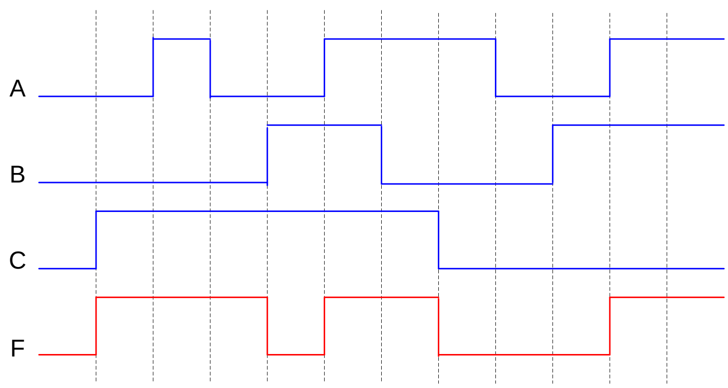
$$F_{MKF}(X,Y,Z) = (\overline{Y} + \overline{Z}) \cdot (Y + Z) \cdot (X + Y)$$

Primer 35. Na slici je prikazan vremenski dijagram promene vrednosti funkcije F(A,B,C) u zavisnosti od promene ulaznih signala A, B i C. Na osnovu zadatih vremenskih dijagrama:

a) Popuniti tabelu istinitosti.

b) Odrediti analitički izraz za minimalnu disjunktivnu formu funkcije primenom Karnoove mape.

c) Nacrtati hardversku realizaciju na osnovu izraza dobijenog minimizacijom primenom osnovnih logičkih kola.



Rešenje:

n	A	B	C	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	0
5	1	0	1	1
6	1	1	0	1
7	1	1	1	1

AB \ C	0	1
00		1
01		
11	1	1
10		1

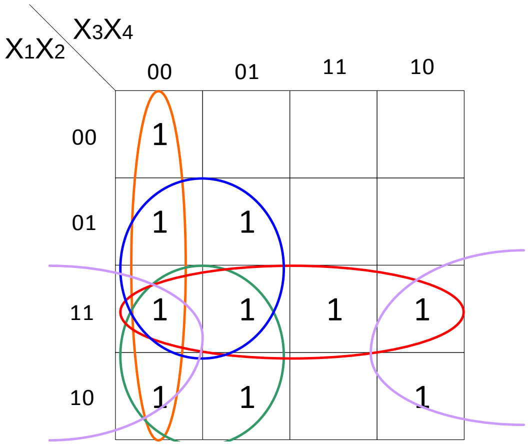
$$F(A,B,C) = A \cdot B + \overline{B} \cdot C$$

Primer 36. a) Opisati tabelom istinitosti kombinacionu mrežu sa ulazima X_1, X_2, X_3 i X_4 koja upoređuje binarne vrednosti ulaza X_1 i X_2 sa binarnim vrednostima na ulazima X_3 i X_4, tako da izlaz mreže Y ima vrednost 1 ukoliko je $X_1X_2 \geq X_3X_4$.

b) Odrediti analitički izraz za izlaz kombinacione mreže $Y(X_1,X_2,X_3,X_4)$ u minimalnoj disjunktivnoj formi primenom Karnoove mape.

Rešenje:

i	X_1	X_2	X_3	X_4	Y
0	0	0	0	0	1
1	0	0	0	1	0
2	0	0	1	0	0
3	0	0	1	1	0
4	0	1	0	0	1
5	0	1	0	1	1
6	0	1	1	0	0
7	0	1	1	1	0
8	1	0	0	0	1
9	1	0	0	1	1
10	1	0	1	0	1
11	1	0	1	1	0
12	1	1	0	0	1
13	1	1	0	1	1
14	1	1	1	0	1
15	1	1	1	1	1



$$Y(X_1, X_2, X_3, X_4) = X_1 \cdot X_2 + \overline{X_3} \cdot \overline{X_4} + X_2 \cdot \overline{X_3} + X_1 \cdot \overline{X_3} + X_1 \cdot \overline{X_4}$$

<u>Primer 37.</u> Neka su A, B i C tri senzora, čiji su digitalni izlazi dovedeni na ulaz kombinacione mreže. Definisati tabelu istinitosti za mrežu čija logička funkcija $Y(A, B, C)$ ima vrednost 0:

a) kada senzori A i C daju istu vrednost na izlazu
b) kada senzori A i C daju različite vrednosti na izlazu
c) odrediti minimalnu konjuktivnu formu (MKF) funkcije $Y(A, B, C)$ za primere a) i b).

<u>Primer 38.</u> Za funkciju koja je zadata kao proizvod potpunih suma (DNF)
$F(A_4, A_3, A_2, A_1) = \Sigma\,(4, 6, 7, 10, 12, 13, 14, 15)$
a) Odrediti minimalnu disjunktivnu formu funkcije (**MDF**).
b) Odrediti minimalnu konjuktivnu formu funkcije (**MKF**).

<u>Primer 39.</u> Za funkciju koja je zadata kao suma potpunih proizvoda (KNF)
$F(X_4, X_3, X_2, X_1) = \Pi\,(0, 1, 2, 4, 6, 8, 9, 10, 12, 14, 15)$
a) Odrediti minimalnu konjuktivnu formu funkcije (**MKF**).
b) Odrediti minimalnu disjunktivnu formu funkcije (**MDF**).

<u>Primer 40.</u> Za funkciju koja je zadata kao suma potpunih proizvoda (KNF)
$F(W, Z, Y, X) = \Pi\,(0, 3, 4, 7, 8, 11, 12, 13, 15) = 0$
a) Odrediti minimalnu konjuktivnu formu funkcije (**MKF**).
b) Odrediti minimalnu disjunktivnu formu funkcije (**MDF**).

<u>Primer 41.</u> Za funkciju $F_{DNF}(A_1, A_2, A_3, A_4) = \Sigma\,(1, 4, 5, 6, 12, 13) = 1$, F(X) = (8, 9, 14,15), gde je X nedefinisana vrednost funkcije:
a) Odrediti minimalnu disjunktivnu formu funkcije (**MDF**).
b) Odrediti minimalnu konjuktivnu formu funkcije (**MKF**).

<u>Primer 42.</u> Za funkciju $F_{KNF}(X_1, X_2, X_3, X_4) = \Pi\,(0, 4, 7, 8, 12) = 0$, F(b) = (1, 3, 9, 12, 14,15), gde je b nedefinisana vrednost funkcije:
a) Odrediti minimalnu konjuktivnu formu funkcije (**MKF**).
b) Odrediti minimalnu disjunktivnu formu funkcije (**MDF**).

7. PRIMENA UNARNIH I BINARNIH LOGIČKIH OPERACIJA

Unarne logičke operacije su instrukcije sa **jednim** operandom. Osnovni tipovi unarnih operacija su:

- **Postavljanje (SET) -** logička vrednost 0 nekog bita pretvara se u logičku vrednost 1. Primenjuje se na bite u registru stanja, registrima CPU i memorijskim lokacijama.

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

- **Brisanje (CLEAR, RESET) -** logička vrednost 1 željenog bita se pretvara u logičku vrednost **0**. Primenjuje se na bite u registru stanja, registrima CPU i memorijskim lokacijama.

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

- **Komplementiranje (COMPLEMENT) -** sve logičke 0 u jednom broju se pretvaraju u 1 i obrnuto, tako da se dobija prvi komplement. Operacija se odnosi na sadržaj celog registra.

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

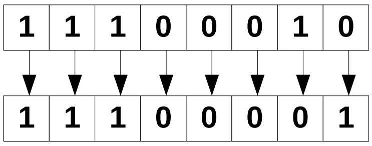
- **Negacija (NEGATE)** – kao rezultat se dobija drugi komplement broja. Operacija se odnosi na sadržaj celog registra.

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

- **Inkrementiranje (INCREMENT) - povećava** vrednosti operanda za 1. Odnosi se na sadržaj celog registra. Kod nekih procesora se primenjuje samo na registre u okviru CPU, a kod drugih se primenjuje i na sadržaj memorijskih lokacija.

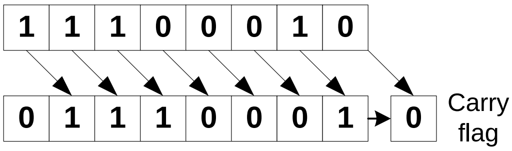
1	1	1	0	0	0	1	0
1	1	1	0	0	0	1	1

- **Dekrementiranje (DECREMENT) -** smanjuje vrednosti operanda za 1. Odnosi se na sadržaj celog registra. Kod nekih procesora se primenjuje samo na registre u okviru CPU, a kod drugih se primenjuje i na sadržaj memorijskih lokacija.

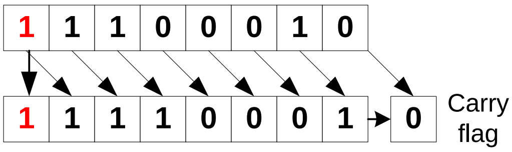


- **Pomeranje (SHIFT)** - pomeraju se biti u okviru jednog registra ulevo ili udesno. Bit koji napušta registar se gubi ili se upisuje u Carry flag. Kod mikroprocesora se obično u jednom taktu obavi pomeranje za jednu poziciju, a kod većih računara se u jednom taktu pomera sadržaj za više pozicija. Upražnjena mesta se obično popunjavaju logičkim nulama (0).

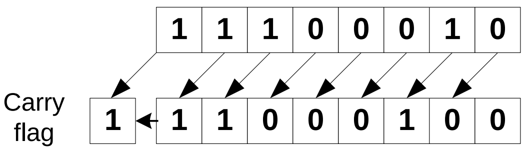
  - **Pomeranje udesno**
    Primena: Ispitivanje sadržaja bita u registru
    Deljenje sa 2^N, N = broj pomeranja



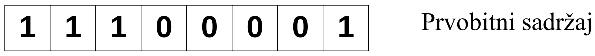
  - **Aritmetičko pomeranje udesno** - bit znaka ostaje na MSB poziciji, a pomeraju se svi ostali biti.



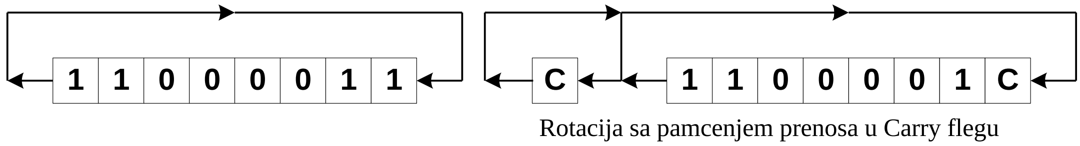
  - **Pomeranje ulevo**
    Primena: Ispitivanje sadržaja bita u registru
    Množenje sa 2^N, N = broj pomeranja



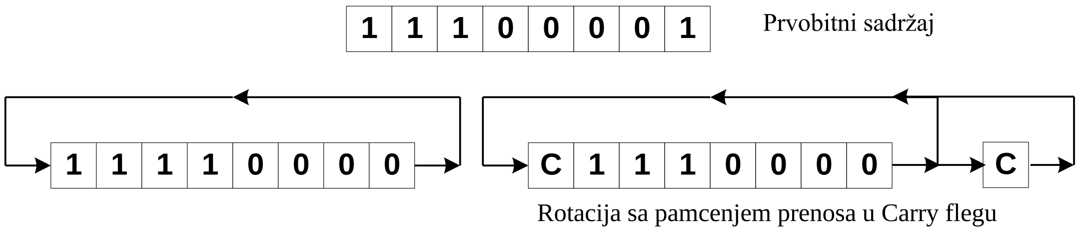
- **Rotacija (ROTATE)** - rotiraju se biti u okviru jednog registra ulevo ili udesno. Bit koji se rotira upisuje se na upražnjeno mesto na drugom kraju registra, a pri tome može da se čuva i u Carry flag-u. Kod nekih procesora i sam bit prenosa (C, carry) učestvuje u rotaciji.



  - **Rotacija ulevo**



  - **Rotacija udesno**



Binarne logičke operacije su instrukcije su instrukcije sa **dva** operanda, od kojih je najčešće bar jedan u nekom od akumulatora (ili drugim registrima CPU), dok je drugi u nekom od registara memorije.

Logičke instrukcije obrađuju sve bite u registru, ali se primenjuju na svaki par bita ponaosob i za svaki par generišu poseban rezultat, koji ne utiče na rezultat u drugim bitima (nema prenosa između dva susedna razreda u registru).

- **Logička I (AND) operacija**

  Primena: Izdvajanje sadržaja dela registra.
  Resetovanje željenih bita.
  pomoću **maske** koja sadrži **1** na mestu bita čiji sadržaj treba da ostane **nepromenjen** (koji treba **izdvojiti**), a **0** na mestu bita koji treba **obrisati** (**resetovati**).

A	0 0 1 0 0 1 1 1	originalan sadržaj registra
B	0 0 0 0 1 1 1 1	**maska**
A **AND** B	0 0 0 0 0 1 1 1	sadržaj posle AND operacije

- **Logička ILI (OR) operacija**

  Primena: Pakovanje podataka u jednu celinu.
  Setovanje željenih bita.
  pomoću **maske** koja sadrži **0** na mestu bita čiji sadržaj treba da ostane **nepromenjen**, a **1** na mestu bita koje treba **postaviti** (**setovati**).

A	1 0 1 1 0 0 0 0	originalan sadržaj registra
B	1 0 0 0 0 1 0 1	**maska**
A **OR** B	1 0 1 1 0 1 0 1	sadržaj posle OR operacije

- **Logička EX-ILI (XOR) operacija**
  Primena: Komplementiranje sadržaja željenih bita, pomoću **maske** koja sadrži **0** na mestu bita čiji sadržaj treba da ostane **nepromenjen**, a **1** na mestu bita koji treba **invertovati**.

A	1 0 1 0 0 0 0 1	originalan sadržaj registra
B	1 0 0 1 0 1 0 1	**maska**
A **XOR** B	0 0 1 1 0 1 0 0	sadržaj posle XOR operacije

Primer 1 U osmobitnom registru A nalazi se broj $7E_{(16)}$. Koji se HEX broj nalazi u registru posle:

a) pomeranja prvobitnog sadržaja za 3 mesta ulevo

b) rotacije prvobitnog sadržaja za 3 mesta ulevo

c) pomeranja prvobitnog sadržaja za 3 mesta udesno

d) rotacije prvobitnog sadržaja za 3 mesta udesno.

Rešenje:

Prvobitni sadržaj u registru A: $01111110_{(2)}$

a) Posle pomeranja za 3 mesta ulevo, sadržaj u registru A je: $\mathbf{11110000_{(2)}}$

b) Posle rotacije za 3 mesta ulevo, sadržaj u registru A je: $\mathbf{11110011_{(2)}}$

c) Posle pomeranja za 3 mesta udesno, sadržaj u registru A je: $\mathbf{00001111_{(2)}}$

d) Posle rotacije za 3 mesta udesno, sadržaj u registru A je: $\mathbf{11001111_{(2)}}$

Primer 2 Iz tabele sa **ASCII** kodovima pročitati kod za broj $9_{(10)}$ i sadržaj sačuvati u osmobitnom registru A. Pokazati kako primenom operacija pomeranja i rotiranja za odgovarajući broj mesta možemo da postignemo da:

a) viši nibl u registru A bude 0

b) niži nibl u registru A bude 0

c) viši nibl dođe na mesto nižeg nibla

d) niži nibl dođe na mesto višeg nibla

Rešenje:

Prvobitni sadržaj u registru A je $00111001_{(2)} = 39_{(16)}$

a) Posle pomeranja za 4 mesta udesno, sadržaj u registru A je $\mathbf{00000011_{(2)} = 03_{(16)}}$

b) Posle pomeranja za 4 mesta ulevo, sadržaj u registru A je $\mathbf{10010000_{(2)} = 90_{(16)}}$

c) Posle rotacije za 4 mesta udesno, sadržaj u registru A je $\mathbf{10010011_{(2)} = 93_{(16)}}$

d) Posle rotacije za 4 mesta ulevo, sadržaj u registru A je $\mathbf{10010011_{(2)} = 93_{(16)}}$

Primer 3 Iz tabele sa **EBCDIC** kodovima pročitati kod za broj $6_{(10)}$ i sadržaj sačuvati u osmobitnom registru A. Pokazati kako primenom operacija pomeranja ili rotiranja za odgovarajući broj mesta možemo da postignemo da:

a) sadržaj u registru A bude $3D_{(16)}$

b) sadržaj u registru A bude $60_{(16)}$

c) viši nibl dođe na mesto nižeg nibla

d) niži nibl dođe na mesto višeg nibla

Rešenje:

Prvobitni sadržaj u registru A je $11110110_{(2)}$

a) Posle pomeranja za 2 mesta udesno, sadržaj u registru A je $\mathbf{00111101_{(2)} = 3D_{(16)}}$

b) Posle pomeranja za 4 mesta ulevo, sadržaj u registru A je $\mathbf{01100000_{(2)} = 60_{(16)}}$

c) Posle rotacije za 4 mesta udesno, sadržaj u registru A je $\mathbf{01101111_{(2)} = 6F_{(16)}}$

d) Posle rotacije za 4 mesta ulevo, sadržaj u registru A je $\mathbf{01101111_{(2)} = 6F_{(16)}}$

Primer 4 U registru A se nalazi broj $B3_{(16)}$. Koji će se binarni sadržaj nalaziti u registru nakon rotacije primenom C flega, ako je:

a) C = 1, a obavljena je rotacija za 4 mesta udesno

b) C = 1, a obavljena je rotacija za 2 mesta ulevo

c) C = 0, a obavljena je rotacija za 4 mesta udesno

d) C = 0, a obavljena je rotacija za 2 mesta ulevo

Rešenje:

Prvobitni sadržaj u registru A je $10110011_{(2)}$

a) Posle rotacije za 4 mesta udesno, ako je **C = 1** sadržaj u registru A je $\mathbf{01111011_{(2)}}$

b) Posle rotacije za 2 mesta ulevo, ako je **C = 1** sadržaj u registru A je $\mathbf{11001111_{(2)}}$

c) Posle rotacije za 4 mesta udesno, ako je **C = 0** sadržaj u registru A je $\mathbf{01101011_{(2)}}$

d) Posle rotacije za 2 mesta ulevo, sadržaj u registru A je $\mathbf{11001101_{(2)}}$

Primer 5 Sadržaj akumulatora predstavljaju slova **Lt** (ASCII kod). Napraviti masku za komplementiranje parnih bitova.

Rešenje:

Sadržaj registra maske je: $\mathbf{0001100100100001_{(2)}}$

Primer 6 Sadržaj akumulatora predstavljaju slova **Cp** (ASCII kod). Napraviti masku za izdvajanje po tri krajnja bita.

Rešenje:

Sadržaj registra maske je: $\mathbf{1100000000000111_{(2)}}$

Primer 7 Sadržaj akumulatora predstavljaju slova **sP** (ASCII kod). Napraviti masku za izdvajanje 1.,3.,7.,9.,12. i 14. bita.

Rešenje:

Sadržaj registra maske je: $\mathbf{0101001000000000_{(2)}}$

Primer 8 Predstaviti u EBCDIC kodu slova **NM** kao sadržaj akumulatora A. Primenom odgovrajućeg sadržaja registra maske izvršiti uključivanje prva tri i poslednja tri prekidača, koji su direktno priključeni na akumulator A bez promene ostalih prekidača.

Rešenje:

Sadržaj registra maske je: $\mathbf{1110010111010111_{(2)}}$

Primer 9 U registru A se nalazi binarni zapis broja $6_{(10)}$.

a) Koji dekadni broj se nalazi u registru posle 5 pomeranja binarnog sadržaja ulevo?

b) Koja aritmetička operacija je realizovana na ovaj način?

Rešenje:

Prvobitni sadržaj u registru A je $00000110_{(2)}$

a) Posle pomeranja za 5 mesta ulevo sadržaj registra A je $\mathbf{11000000_{(2)} = 192_{(10)}}$

b) Na ovaj način je izvršeno množenje broja 5 sa $\mathbf{2^5 = 32}$

Primer 10 U 8-bitnom registru A nalazi se ASCII kod broja $9_{(10)}$, a u 8-bitnom registru B se nalazi se ASCII kod broja $7_{(10)}$.

a) Primenom odgovarajuće logičke operacije i maske pretvoriti ASCII kod brojeva u registru A i B u raspakovani format BCD koda "8421".

b) Kako primenom rotacije i odgovarajuće logičke operacije nad sadržajem registara A i B dobijenim u tački a) može da se postigne da se u registru A nalazi sadržaj $79_{(16)}$ (pakovani BCD format).

c) Kako primenom pomeranja i odgovarajuće logičke operacije nad sadržajem registara A i B dobijenim u tački a) može da se postigne da se u registru B nalazi sadržaj $79_{(16)}$ (pakovani BCD format).

Rešenje:

Prvobitni sadržaj u registru A je $00111001_{(2)}$, a u registru B je $00110111_{(2)}$

a) Primenom logičke I (AND) operacije nad sadržajem registara A, odnosno B i maske $00001111_{(2)}$ dobija se zapis brojeva u raspakovanom formatu BCD koda "8421":

$$A = 00111001_{(2)}\ \mathbf{AND}\ 00001111_{(2)} = \mathbf{00001001_{(2)}}$$

$$B = 00110111_{(2)}\ \mathbf{AND}\ 00001111_{(2)} = \mathbf{00000111_{(2)}}$$

b) Primenom rotacije sadržaja registra B za 4 mesta udesno dobija se $B = 01110000_{(2)}$

Primenom operacije **ILI (OR)** između registara A i B, pri čemu se rezultat čuva u registru A dobija se A = A **OR** B = $\mathbf{01111001_{(2)}}$ = $\mathbf{79_{(16)}}$

c) Pomeranjem sadržaja registra B za četiri mesta ulevo dobija se B = $01110000_{(2)}$

Primenom operacije **ILI (OR)** između registara A i B, pri čemu se rezultat čuva u registru B dobija se B = A **OR** B = **$01111001_{(2)}$** = **$79_{(16)}$**

Primer 11 Primenom odgovarajuće maske i logičke operacije **invertovati** 2., 3., 8., 9., 12. i 13. bit u 16-bitnom broju koji se nalazi u akumulatoru A:

A: $1101010111100010_{(2)}$

Nakon izvršenja logičke operacije rezultat se nalazi u akumulatoru A.

Rešenje:

	A:	1	1	0	1	0	1	0	1	1	1	1	0	0	0	1	0
	M:	0	0	1	1	0	0	1	1	0	0	0	0	1	1	0	0
A *XOR* **M** →	**A:**	1	1	1	0	0	1	1	0	1	1	1	0	1	1	1	0

Primer 12 Primenom odgovarajuće maske i logičke operacije **invertovati** 2., 5., 6., 8., 11., 13. i 14. bit u 16-bitnom broju koji se nalazi u akumulatoru A: $0011010101010101_{(2)}$. Nakon izvršenja logičke operacije rezultat se nalazi u akumulatoru A.

Rešenje:

	A:	0	0	1	1	0	1	0	1	0	1	0	1	0	1	0	1
	M:	0	1	1	0	1	0	0	1	0	1	1	0	0	1	0	0
A *XOR* **M** →	A:	0	1	0	1	1	1	0	0	0	0	1	1	0	0	0	1

Primer 13 U akumulatoru A nalazi se sledeći 16-bitni podatak: $1011011011101001_{(2)}$. Primenom odgovarajuće maske i logičke operacije **invertovati** 0.,1.,4.,7.,10. i 11. bit. Nakon izvršenja logičke operacije rezultat se nalazi u akumulatoru A.

Rešenje:

	A:	1	0	1	1	0	1	1	0	1	1	1	0	1	0	0	1
	M:	0	0	0	0	1	1	0	0	1	0	0	1	0	0	1	1
A *XOR* **M** →	A:	1	0	1	1	1	0	1	0	0	1	1	1	1	0	1	0

Primer 14 U akumulatoru A nalazi se sledeći 16-bitni podatak: $1100000110000101_{(2)}$. Primenom odgovarajuće maske i logičke operacije **komplementirati** po tri krajnja stanja u ovom registru. Nakon izvršenja logičke operacije rezultat se nalazi u akumulatoru A.

Rešenje:

	A:	1	1	0	0	0	0	0	1	1	0	0	0	0	1	0	1
	M:	1	1	1	0	0	0	0	0	0	0	0	0	0	1	1	1
A *XOR* **M** →	A:	0	0	1	0	0	0	0	1	1	0	0	0	0	0	1	0

Primer 15 U akumulatoru A nalazi se sledeći podatak A: $1001110111011110_{(2)}$, koji služi za upravljanje (uključivanje-1 i isključivanje-0). U određenom trenutku treba **isključiti** uređaje koji su povezani na b_2 , b_4 , b_6, b_8 , b_{10} i b_{12} bitove ovog 16-to bitnog registra. Smatrati da se nakon izvršenja logičke operacije rezultat se nalazi u akumulatoru A.

Rešenje:

	A:	1	0	0	1	1	1	0	1	1	1	0	1	1	1	1	0
	M:	1	1	1	0	1	0	1	0	1	0	1	0	1	0	1	1
A *AND* **M** →	A:	1	0	0	0	1	0	0	0	1	0	0	0	1	0	1	0

Primer 16 Primenom odgovarajuće maske i logičke operacije **resetovati** 0., 1., 6., 7., 14. i 15. bit u 16-bitnom broju koji se nalazi u akumulatoru A:

A: $1110100111000111_{(2)}$

Nakon izvršenja logičke operacije rezultat se nalazi u akumulatoru A.

Rešenje:

A:	1	1	1	0	1	0	0	1	1	1	0	0	0	1	1	1
M:	0	0	1	1	1	1	1	1	0	0	1	1	1	1	0	0
A *AND* **M** → **A:**	0	0	1	0	1	0	0	1	0	0	0	0	0	1	0	0

<u>Primer 17</u> Primenom odgovarajuće maske i logičke operacije **resetovati** 0.,3.,4.,8.,11.,14. i 15. bit u 16-bitnom broju koji se nalazi u akumulatoru A: $1010101010101010_{(2)}$. Nakon izvršenja logičke operacije rezultat se nalazi u akumulatoru A.

Rešenje:

A:	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
M:	0	0	1	1	0	1	1	0	1	1	1	0	0	1	1	0
A *AND* **M** → A:	0	0	1	0	0	0	1	0	1	0	1	0	0	0	1	0

<u>Primer 18</u> U akumulatoru A nalazi se sledeći podatak A : $1100100101001001_{(2)}$, koji služi za upravljanje određenim prekidačima. Izvršiti selektivno **uključivanje** 1., 2., 4., 5., 9. i 10. prekidača, bez promene ostalih prekidača. Nakon izvršenja logičke operacije rezultat se nalazi u akumulatoru A.

Rešenje:

A:	1	1	0	0	1	0	0	1	0	1	0	0	1	0	0	1
M:	0	0	0	0	0	1	1	0	0	0	1	1	0	1	1	0
A *OR* **M** → **A:**	1	1	0	0	1	1	1	1	0	1	1	1	1	1	1	1

<u>Primer 19</u> Primenom odgovarajuće maske i logičke operacije **setovati** 4., 6., 8., 10. i 12. bit u 16-bitnom broju koji se nalazi u akumulatoru A:

A: $0010101010101001_{(2)}$

Nakon izvršenja logičke operacije rezultat se nalazi u akumulatoru A.

Rešenje:

A:	0	0	1	0	1	0	1	0	1	0	1	0	1	0	0	1
M:	0	0	0	1	0	1	0	1	0	1	0	1	0	0	0	0
A *OR* M → A:	0	0	1	1	1	0	1	0	1	1	1	1	1	0	0	1

Primer 20 Primenom odgovarajuće maske i logičke operacije **izdvojiti** 1., 3., 5., 11., 13. i 15. bit u 16-bitnom broju koji se nalazi u akumulatoru A:

A: $1110110111101011_{(2)}$

Nakon izvršenja logičke operacije rezultat se nalazi u akumulatoru A.

Rešenje:

A:	1	1	1	0	1	1	0	1	1	1	1	0	1	0	1	1
M:	1	0	1	0	1	0	0	0	0	0	1	0	1	0	1	0
A *AND* M → A:	1	0	1	0	1	0	0	0	0	0	1	0	1	0	1	0

Primer 21 Primenom odgovarajuće maske i logičke operacije **izdvojiti** 3., 5., 9. i 10. bit u 16-bitnom broju koji se nalazi u akumulatoru A: 1001001001001001. Nakon izvršenja logičke operacije rezultat se nalazi u akumulatoru A.

Rešenje:

A:	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1
M:	0	0	0	0	0	1	1	0	0	0	1	0	1	0	0	0
A *AND* M → A:	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0

Primer 22 U akumulatoru A nalazi se sledeći 16-bitni podatak: $0100110011001101_{(2)}$. Primenom odgovarajuće maske i logičke operacije izdvojiti dva najviša i tri najniža bita iz ovog akumulatora. Nakon izvršenja logičke operacije rezultat se nalazi u akumulatoru A.

Rešenje:

A:	0	1	0	0	1	1	0	0	1	1	0	0	1	1	0	1
M:	1	1	0	0	0	0	0	0	0	0	0	0	0	1	1	1
A *AND* **M** → A:	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	1

Primer 23 U registru A se nalazi binarni zapis broja $72_{(10)}$.

a) Koji je decimalan broj se u registru posle 2 pomeranja binarnog sadržaja ulevo?

b) Koja aritmetička operacija je realizovana na ovaj način?

Primer 24 U registru A se nalazi binarni zapis broja $9_{(10)}$.

a) Koji dekadni broj se nalazi u registru posle 4 pomeranja binarnog sadržaja ulevo?

b) Koja aritmetička operacija je realizovana na ovaj način?

Primer 25 U registru A se nalazi binarni zapis broja $40_{(10)}$.

a) Kako primenom operacije pomeranja u odgovarajućem smeru broj može da se podeli sa 8?

b) Šta se nalazi u registru nakon poslednje operacije pomeranja?

Primer 26 U registru A se nalazi binarni zapis broja $16_{(10)}$.

a) Kako primenom operacije pomeranja u odgovarajućem smeru ovaj broj može da se pomnoži sa 16?

b) Šta se nalazi u registru nakon poslednje operacije pomeranja?

Primer 27 U 16-bitnom registru A nalazi se binarni broj $0011000011101011_{(2)}$. Primenom odgovarajuće maske i logičke operacije **invertovati**:

a) 1.,2.,5.,8.,10.,12. i 14. bit;

b) niži bajt i najviši bit;

c) po četiri krajnja bita;

d) svaki neparni bit;

e) 4.,5.,8.,9.,11.,13. i 15. bit.

Primer 28 U 16-bitnom registru A nalazi se binarni broj $0011010100001111_{(2)}$. Primenom odgovarajuće maske i logičke operacije **isključiti**:

a) 0.,2.,7.,10.,11.,12. i 15. bit;

b) sve parne bitove;

c) sve neparne bitove;

d) viši bajt;

e) 3.,4., 8.,9.,12. i 13. bit.

Primer 29 U 16-bitnom registru A nalazi se binarni broj $1111000010101010_{(2)}$. Primenom odgovarajuće maske i logičke operacije **uključiti**:

a) 2.,3.,4.,8.,9.,11.,13. i 15. bit;

b) dva najniža bita i tri najviša bita;

c) svaki parni bit;

d) niži bajt;

e) 0.,1.,5.,6.,10.,12. i 14. bit.

Primer 30 U 16-bitnom registru A nalazi se binarni broj $1100110101011100_{(2)}$. Primenom odgovarajuće maske i logičke operacije **izdvojiti**:

a) 2.,3.,5.,7.,10.,11. i 13. bit;

b) najviši i najniži bit;

c) viši bajt;

d) najviši bit i niži bajt;

e) 0.,1.,7.,8.,14. i 15. bit.

Primer 31 U 16-bitnom registru A nalazi se binarni broj $1111100111101111_{(2)}$. Primenom odgovarajuće maske i logičke operacije:

a) izdvojiti viši bajt

b) izdvojiti niži bajt

c) izdvojiti 5., 6., 7., 8., 12., 13., 14. i 15. bit.

d) resetovati najniža 4 bita i najviša 4 bita

e) resetovati 1., 3., 5., 12. i 14. bit

Primer 32 U 16-bitnom registru A nalazi se binarni broj $1000001100101000_{(2)}$. Primenom odgovarajuće maske i logičke operacije:

a) setovati najniža 3 bita

b) setovati 2., 3., 4., 5., 11., 12. i 14. bit

Primer 33 U 16-bitnom registru A nalazi se binarni broj $1001100111101001_{(2)}$. Primenom odgovarajuće maske i logičke operacije:

a) invertovati viši bajt

b) invertovati niži bajt

c) invertovati svaki parni bit

d) invertovati svaki neparni bit

e) invertovati 0., 1., 4., 6., 8, 10. i 13. bit

Primer 34 U 8-bitnom registru A nalazi se broj $C5_{(16)}$. Koji se HEX broj nalazi u registru posle:

a) pomeranja prvobitnog sadržaja za 5 mesta ulevo

b) rotacije prvobitnog sadržaja za 5 mesta ulevo

c) pomeranja prvobitnog sadržaja za 5 mesta udesno

d) rotacije prvobitnog sadržaja za 5 mesta udesno.

Primer 35 Iz tabele sa ASCII kodovima pročitati kodove za veliko slovo **M** i malo slovo **s**. Pročitane kodove čuvati u 16-bitnom registru A, tako da se u višem bajtu nalazi veliko slovo **M**. Pokazati kako primenom operacija pomeranja ili rotiranja za odgovarajući broj mesta možemo da postignemo da:

a) viši bajt u registru A bude 0

b) niži bajt u registru A bude 0

c) viši bajt dođe na mesto nižeg bajta

d) niži bajt dođe na mesto višeg bajta.

Primer 36 Iz tabele sa **EBCDIC** kodovima pročitati kodove za velika početna slova vašeg imena i prezimena. Pročitane kodove čuvati u 16-bitnom registru A, tako da se u višem bajtu nalazi početno slovo imena. Pokazati kako primenom operacija pomeranja ili rotiranja za odgovarajući broj mesta možemo da postignemo da:

a) viši bajt u registru A bude 0

b) niži bajt u registru A bude 0

c) viši bajt dođe na mesto nižeg bajta

d) niži bajt dođe na mesto višeg bajta.

Primer 37 U 8 – bitnom registru A se nalazi broj $4_{(16)}$, a u registru B je broj $5_{(16)}$.

a) Primenom odogovarajuće logičke operacije i maske konvertovati sadržaj registara A i B u odgovarajuće ASCII kodove.

b) Kako primenom pomeranja i odgovarajuće logičke operacije nad zadatim prvobitnim sadržajem registara A i B može da se postigne da se u registru A nalazi HEX broj $45_{(16)}$ (pakovani BCD format).

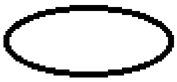
c) Kako primenom pomeranja i odgovarajuće logičke operacije nad zadatim prvobitnim sadržajem registara A i B može da se postigne da se u registru B nalazi HEX broj $54_{(16)}$ (pakovani BCD format).

8. ALGORITMI

Reč algoritam je arapskog porekla i znaci veština računanja, četiri osnovne računarske radnje. Izraz potiče od arapskog matematičara Mohameda ibn Musa Alhariymi. U srednjem veku upotrebljavali su se i izrazi alorizam i algaritam. Algoritam je opis za rešavanje nekog problema. U novije vreme, algoritam je pojam koji se gotovo isključivo vezuje za informatiku i, mada ne postoji jedinstvena opšte prihvaćena definicija, podrazumeva se da je u pitanju nekako opisna procedura za obavljanje posla.

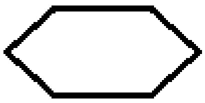
Definicija:
Algoritam je konačna i precizno definisana procedura, formulisana u cilju rešavanja određenog tipa zadatka, kojom se ulazne vrednosti transformišu u izlazne, ili se opisuje izvršavanje nekog postupka. Dijagram toka se često koristi za grafički prikaz algoritma.

Grafički prikaz elemenata**:**

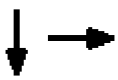

Početak ili kraj programa


Unošenje podataka


Obrada


Odluka (pitalica)


Izlaz (štampanje) podataka


Povezivanje algoritamskih koraka (strelice)


Priključna tačka (povezivanje algoritamskih strelica)

Algoritmi mogu biti prikazani ili realizovani na više načina i to:

- Prirodnim jezikom,
- Pseudokodom, veštačkim precizno definisanim jezikom,
- Grafički, dijagramom toka ili strukturnim dijagramima ili
- Odgovarajućim programskim jezikom.

Postoji više načina za razvrstavanje algoritama, a metodologija klasifikacije je tema mnogih rasprava. Jedan način razvrstavanja je po metodologiji projektovanja ili primenjenom obrascu. Postoji izvestan broj različitih obrazaca kako se pristupa realizaciji algoritama.

Drugi način razvrstavanja je po implementaciji. Rekurzivni algoritam je takav algoritam koji poziva (referencira) sam sebe uzastopno dok se neki uslov ne ispuni, što je metoda primenjena kod funkcionalnog programiranja. Algoritmi se obično razmatraju uz pretpostavku da u jednom trenutku izvršavaju jednu instrukciju jednog algoritma. Takvi računari se ponekad zovu serijski računari. Algoritam osmišljen za takvo okruženje se zove serijski algoritam, nasuprot paralelnim algoritmima, koji koriste prednosti računarske arhitekture kod koje više procesora u istom trenu rešava isti problem.

U teoriji složenosti, što nije isto što i teorija izračunljivosti, se izučava problematika složenosti, kompleksnosti, algoritma, u smislu zauzimanja resursa. Vremenska složenost algoritma se iskazuje kao broj elementarnih koraka za obavljanje algoritma, što je zavisno od veličine ulaza, a koja može biti izražena u bitovima ili nekim sličnim merilom. Ako kažemo da je algoritam (A) uređivanja niza od n elemenata problem vremenske složenosti n^2, znači da dvostruko veći broj elemenata zahteva četiri puta više vremena za uređivanje. Ako je, pak drugi algoritam (B) malo pažljivije napisan i brži je dvostruko, on će raditi dvostruko brže za bilo koju veličinu niza.

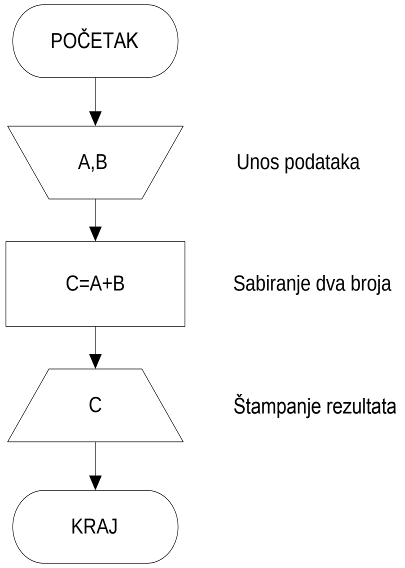
Međutim, ako se namučimo i osmislimo suštinski drugačiji algoritam (V) za uređivanje, on stvarno može biti reda složenosti $n \cdot log(n)$. Algoritmi (A) i (B) su iste složenosti, jer se u notaciji sa velikim O obeležavaju sa $O(n^2)$, a u govoru se zovu 'algoritmi kvadratne složenosti', dok je algoritam (V) 'algoritam složenosti $n \cdot log(n)$'. Zaključak: algoritam (V) je najbolji sa stanovišta korišćenja vremena. Prostorna složenost se na isti način odnosi na funkciju zavisnosti zauzimanja memorijskog prostora u zavisnosti od veličine ulaza.

U programima imamo delove koji se više puta ponavljaju i onda za njih obrazujemo cikličnu strukturu ili programski ciklus. Obrazovanje programskih ciklusa znatno smanjuje dužinu programa.

Postoje *while*, *untile*, *if* i *for* petlje. Dokaz ispravnosti, korektnosti, algoritma je teoretski matematički postupak dokaza teoreme predikatskim računom. Algoritam se izražava logičkim izrazima, definiše se invarijanta - izraz čija vrednost ostaje nepromenjena sve vreme rada algoritma - i dokazuje da izraz koji je važio pre početka važi i po završetku obrade. Potpun dokaz ispravnosti podrazumeva još i dokaz da će se algoritam završiti u konačnom vremenu, međutim to ume biti komplikovanije od prvog dela.

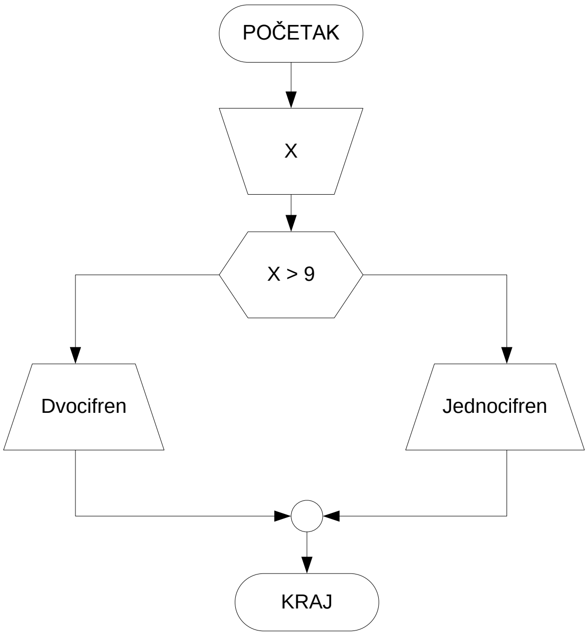
Primer 1. Napraviti algoritam za sabiranje dva broja.

Rešenje:



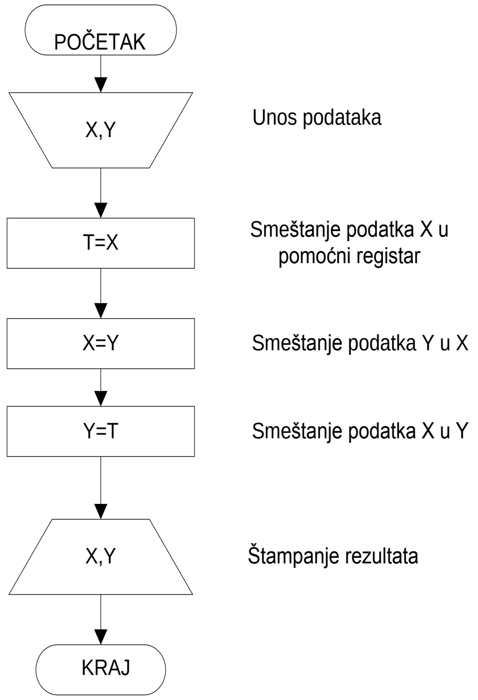
Primer 2. Napisati algoritam koji za uneti prirodni broj do 100 ispisuje da li je broj dvocifren ili jednocifren.

Rešenje:



Primer 3. Napraviti algoritam koji učitava brojeve X i Y i vrši zamenu njihovih vrednosti.

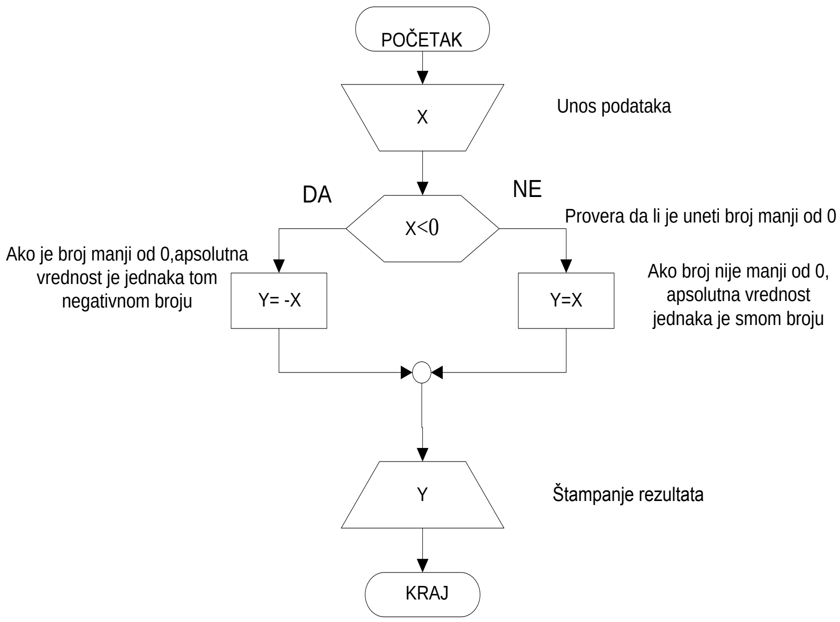
Rešenje:



Primer 4. Napraviti algoritam koji određuje apsolutnu vrednost broja X.

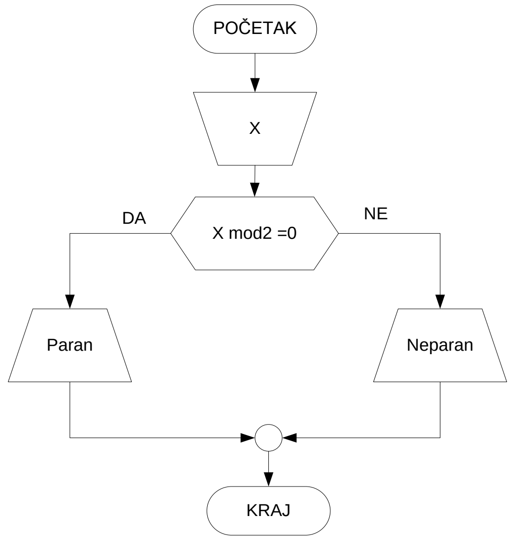
$$y = |x| = \begin{cases} -x;\ x < 0 \\ x;\ x \geq 0 \end{cases}$$

Rešenje:



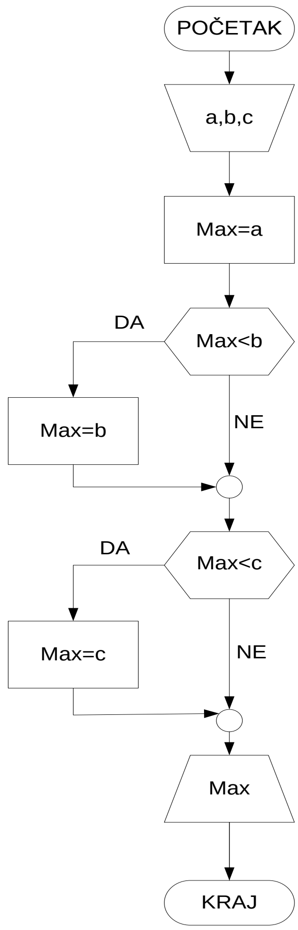
Primer 5. Napisati algoritam koji ispituje da li je broj paran ili neparan

Rešenje:



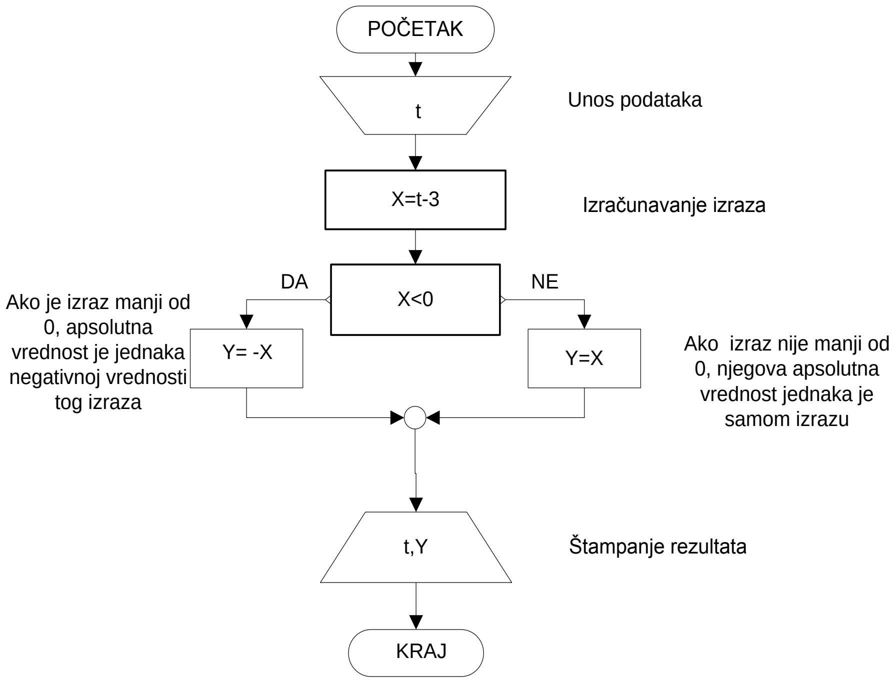
Primer 6. Napraviti algoritam koji promenljivoj max dodeljuje najvecu vrednost tri zadata broja a, b, c

Rešenje:



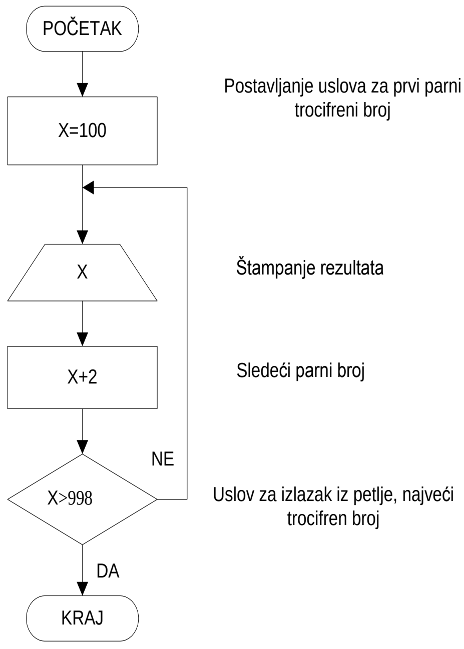
Primer 7. Napraviti algoritam koji izračunava izraz |t-3| i štampa uneti podatak t i apsolutnu vrednost izraza .

Rešenje:



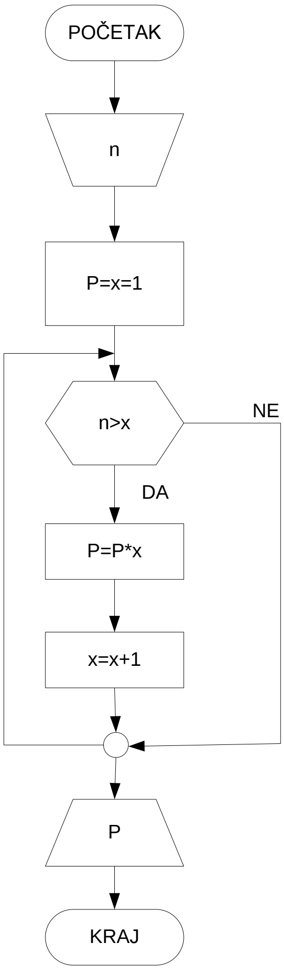
Primer 8. Napraviti algoritam koji štampa sve parne trocifrene brojeve.

Rešenje:



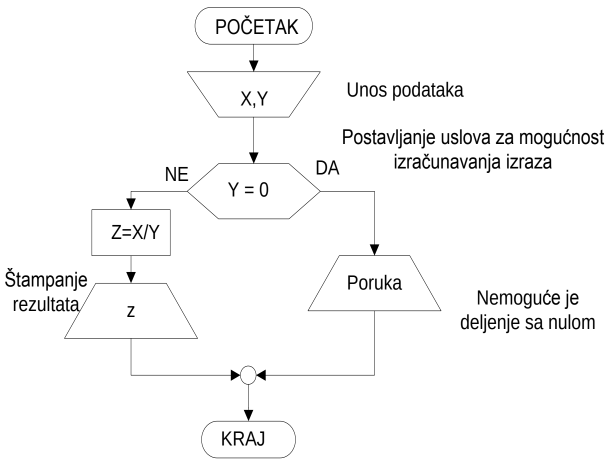
Primer 9. Napisati algoritam za izračunavanje faktorijala.

Rešenje:



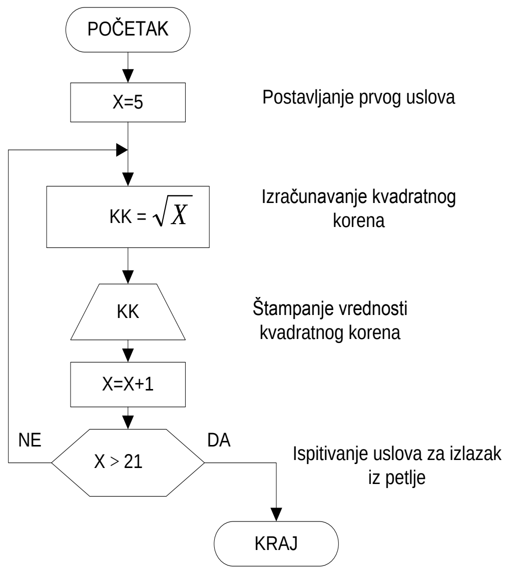
Primer 10. Napraviti algoritam za unošenje dva prirodna broja (x,y) i izračunavanje izraza: z = x/y. Štampati vrednost izraza, z.

Rešenje:



Primer 11. Napraviti algoritam koji izračunava i štampa kvadratni koren brojeva od 5 do 21.

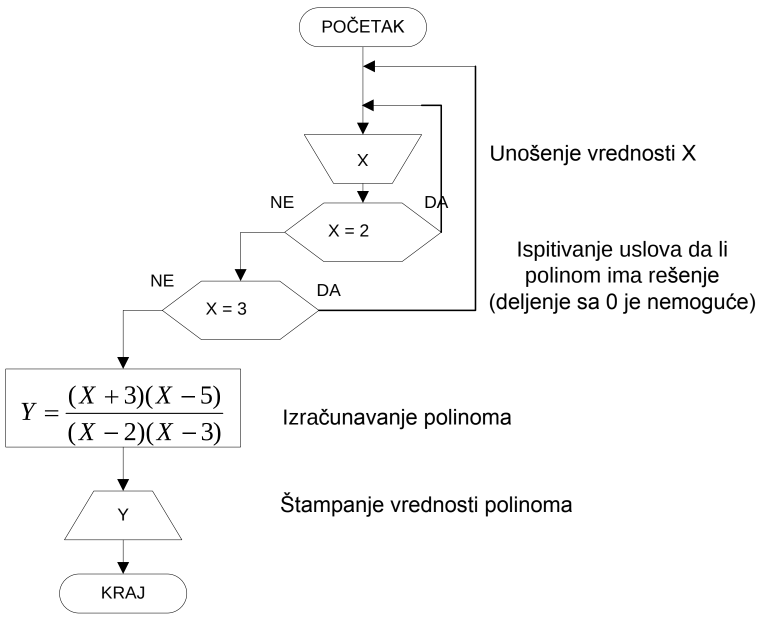
Rešenje:



Primer 12. Napraviti algoritam koji učitava vrednost x, izračunava i štampa vrednost polinoma y .

$$Y = \frac{(X+3)(X-5)}{(X-2)(X-3)}$$

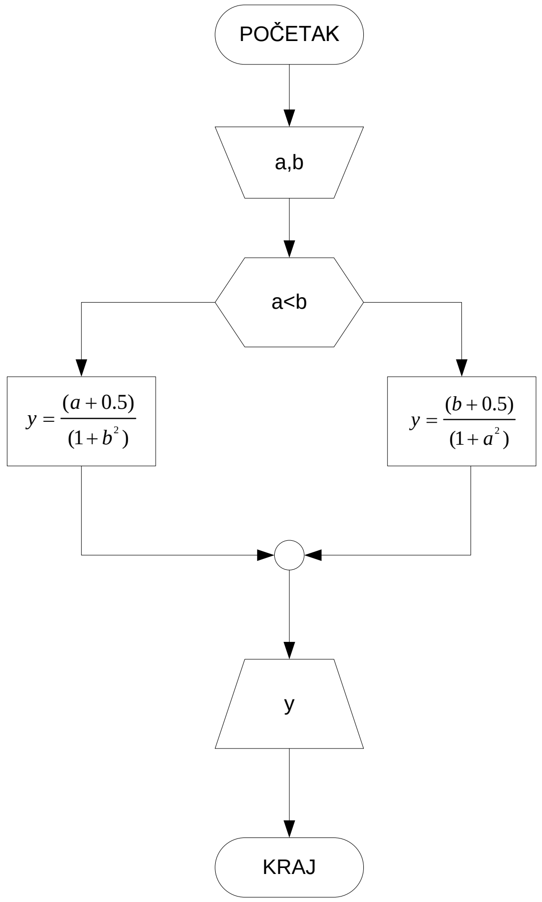
Rešenje:



Primer 13. Sastaviti algoritam kojim se za zadate realne brojeve a i b, izračunava funkcija y prema formuli:

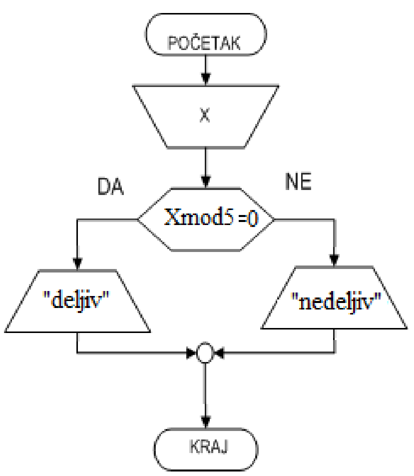
$$y = \frac{\min(a, b) + 0{,}5}{1 + max^2(a, b)}$$

Rešenje:



Primer 14. Napisati algoritam koji proverava da li je broj deljiv sa 5.

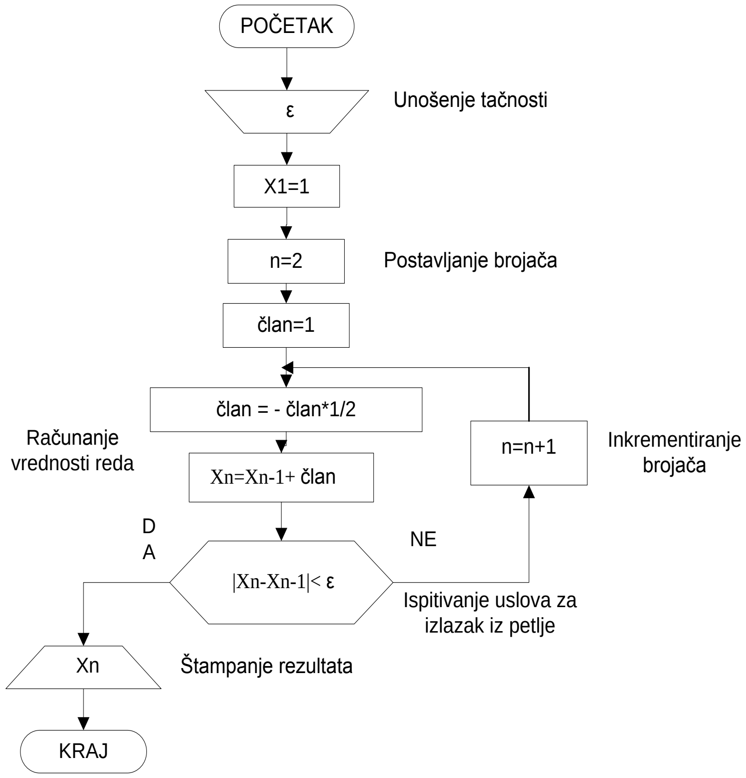
Rešenje:



Primer 15. Napraviti algoritam za izračunavanje i štampanje reda $1-\frac{1}{2}+\frac{1}{4}-\frac{1}{8}...$ sa tačnošću ε.

Rešenje:

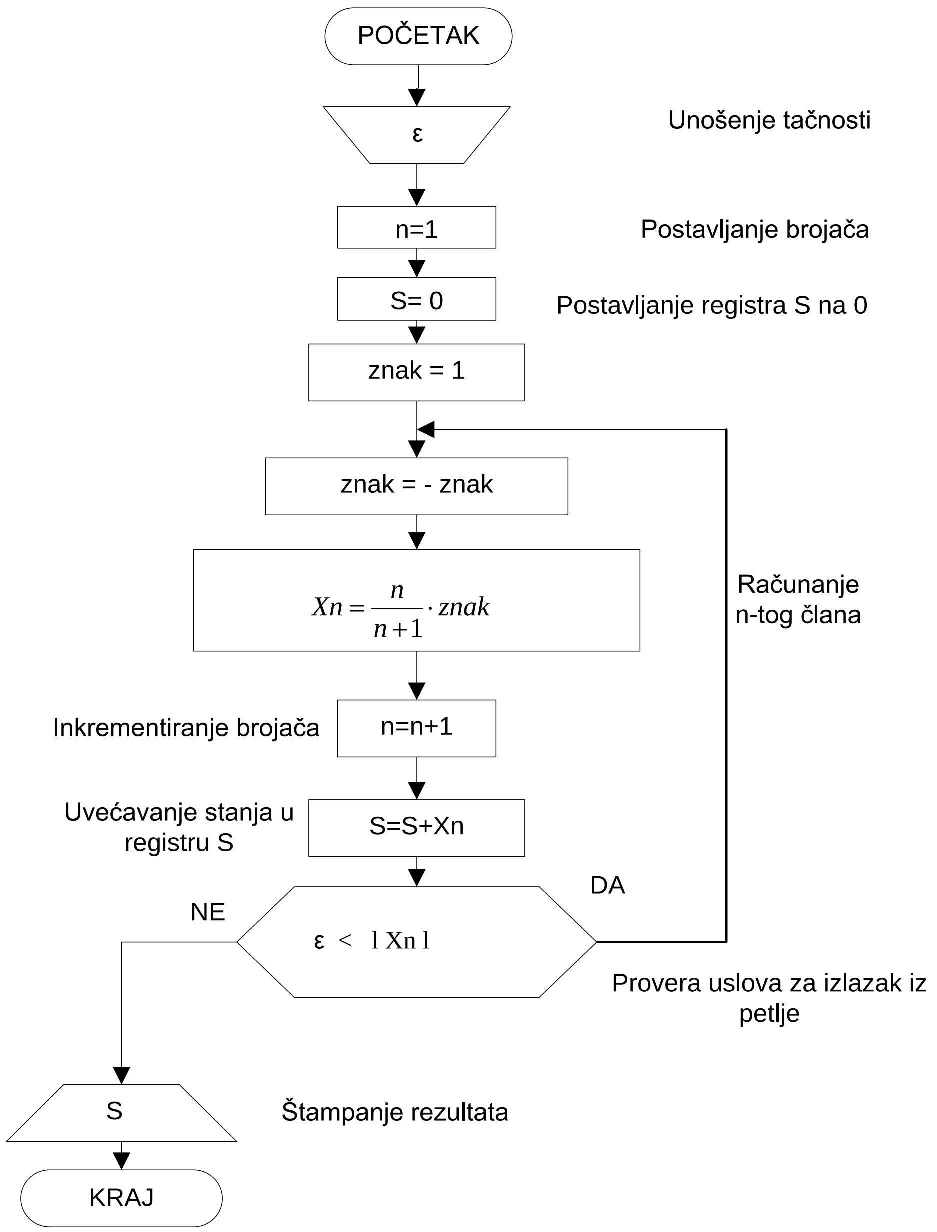
$X_1=1$
$X_2=X_1-1/2^1$
$X_3=X_2+1/2^2$
$X_4=X_3-1/2^3$
$X_n=X_{n-1}+((1)^{n+1}*1/2^{n1})$



Primer 16. Napraviti algoritam za izračunavanje i štampanje reda $-\frac{1}{2}+\frac{2}{3}-\frac{3}{4}+\frac{4}{5}-\frac{5}{6}+\frac{7}{6}...$ sa tačnošću ε.

Rešenje: Opšti član je:

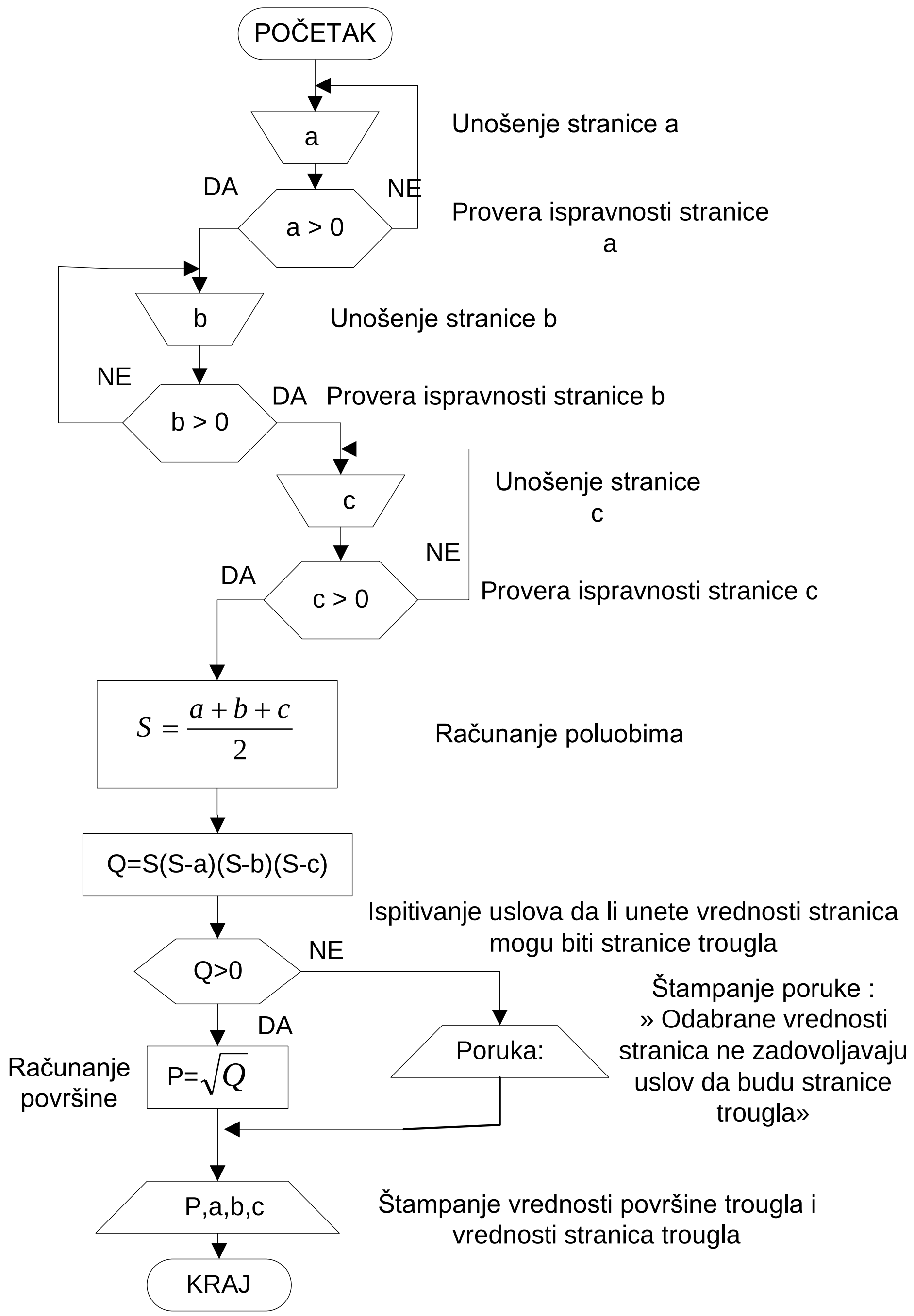
$$Xn=\frac{n}{n+1}\cdot(-1)^n$$



<u>Primer 17.</u> Napraviti algoritam koji učitava stranice trougla *a, b, c* i izračunava površinu trougla po Heronovom obrascu i štampa vrednost površine trougla i stranice tog trougla.

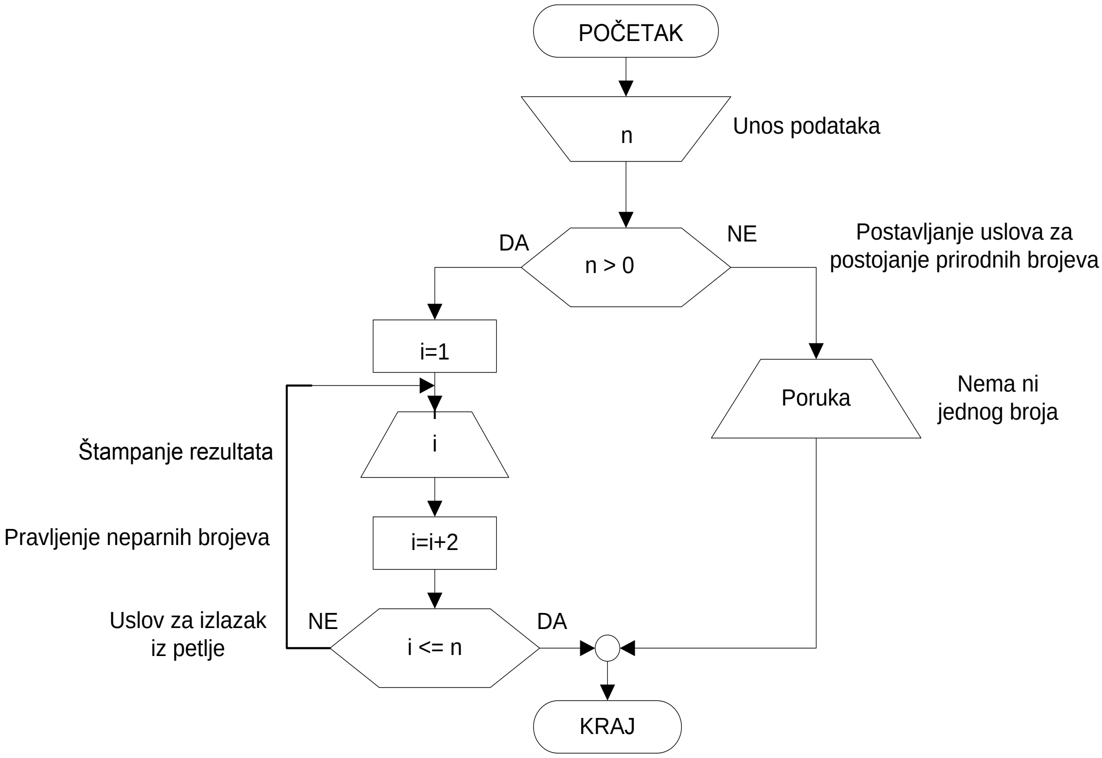
Rešenje: $P = \sqrt{S(S-a)(S-b)(S-c)}$ Heronov obrazac.

$S = \frac{a+b+c}{2}$ Poluobim.


POČETAK
a
Unošenje stranice a
DA
NE
a > 0
Provera ispravnosti stranice a
b
Unošenje stranice b
NE
b > 0
DA
Provera ispravnosti stranice b
c
Unošenje stranice c
NE
DA
c > 0
Provera ispravnosti stranice c
$S = \frac{a+b+c}{2}$
Računanje poluobima
Q=S(S-a)(S-b)(S-c)
Ispitivanje uslova da li unete vrednosti stranica mogu biti stranice trougla
Q>0
NE
DA
Štampanje poruke : » Odabrane vrednosti stranica ne zadovoljavaju uslov da budu stranice trougla»
Računanje površine
P=$\sqrt{Q}$
Poruka:
P,a,b,c
Štampanje vrednosti površine trougla i vrednosti stranica trougla
KRAJ

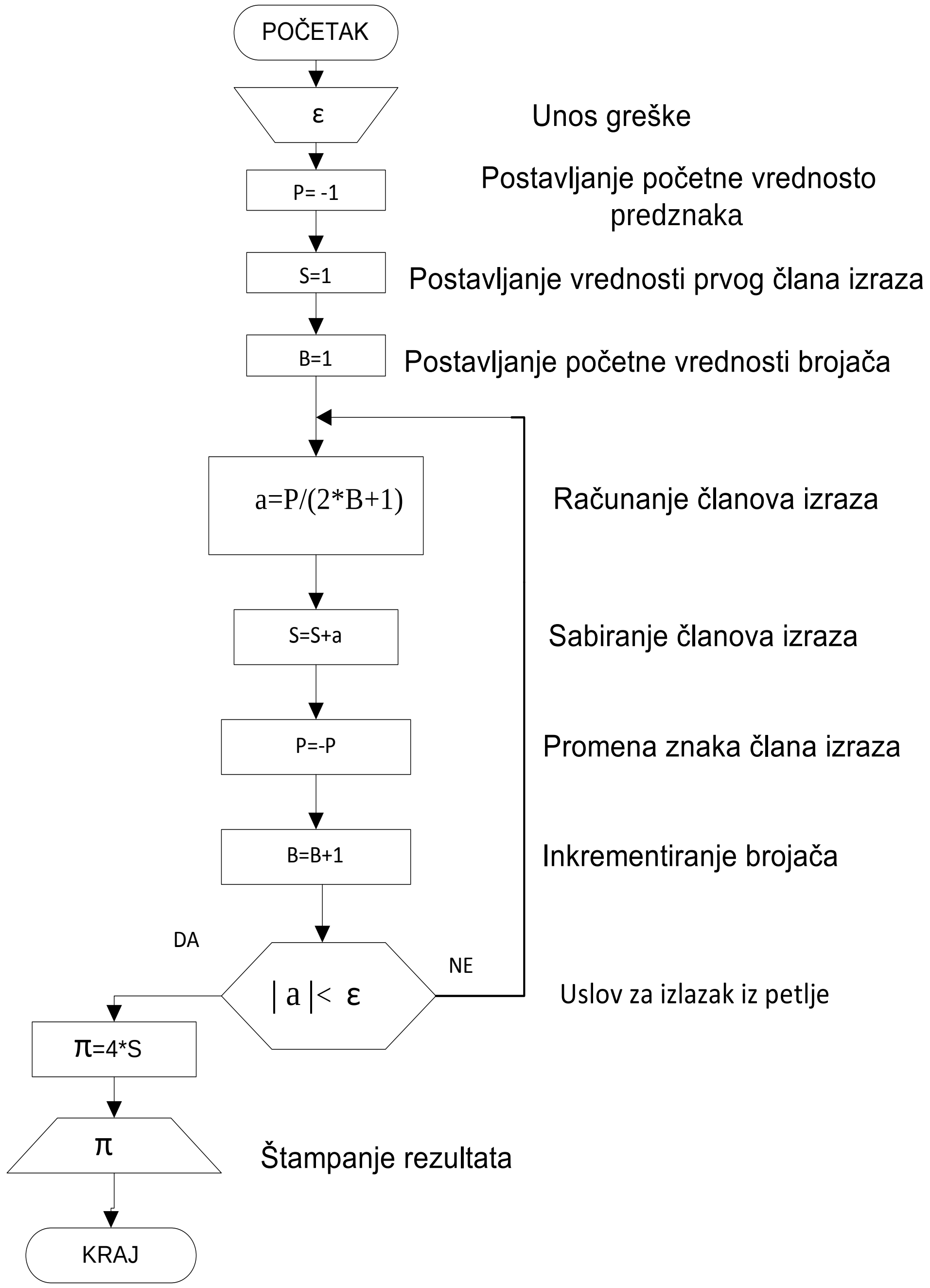
Primer 18. Napraviti algoritam koji štampa sve neparne prirodne brojeve koji su manji i jednaki n.

Rešenje:



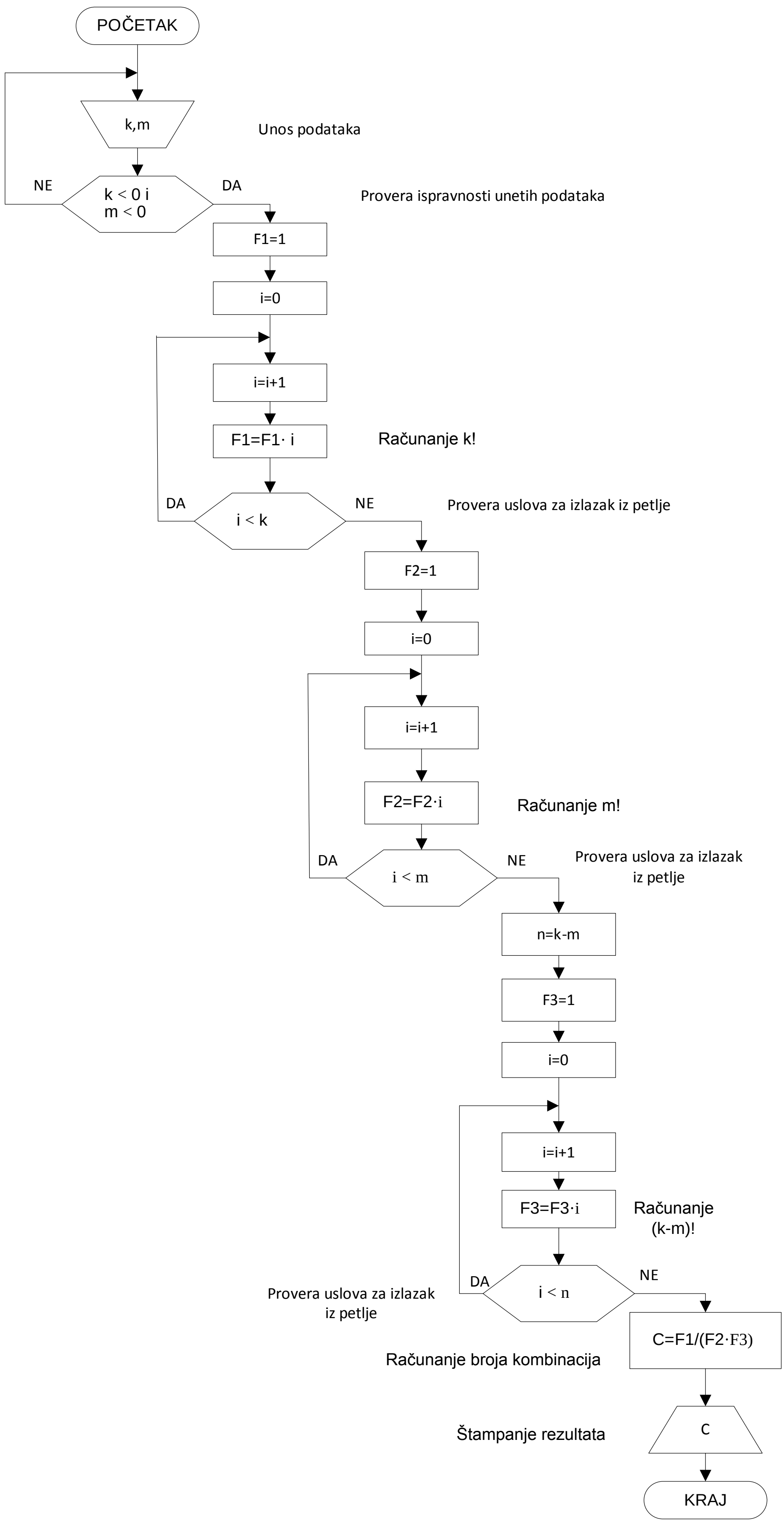
Primer 19. Napraviti algoritam koji izračunava i štampa vrednost broja π primenom izraza: $\frac{\pi}{4} = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} ...$. Uslov izračunavanja je da je n-ti član reda $|a_n| < \varepsilon$.

Rešenje:



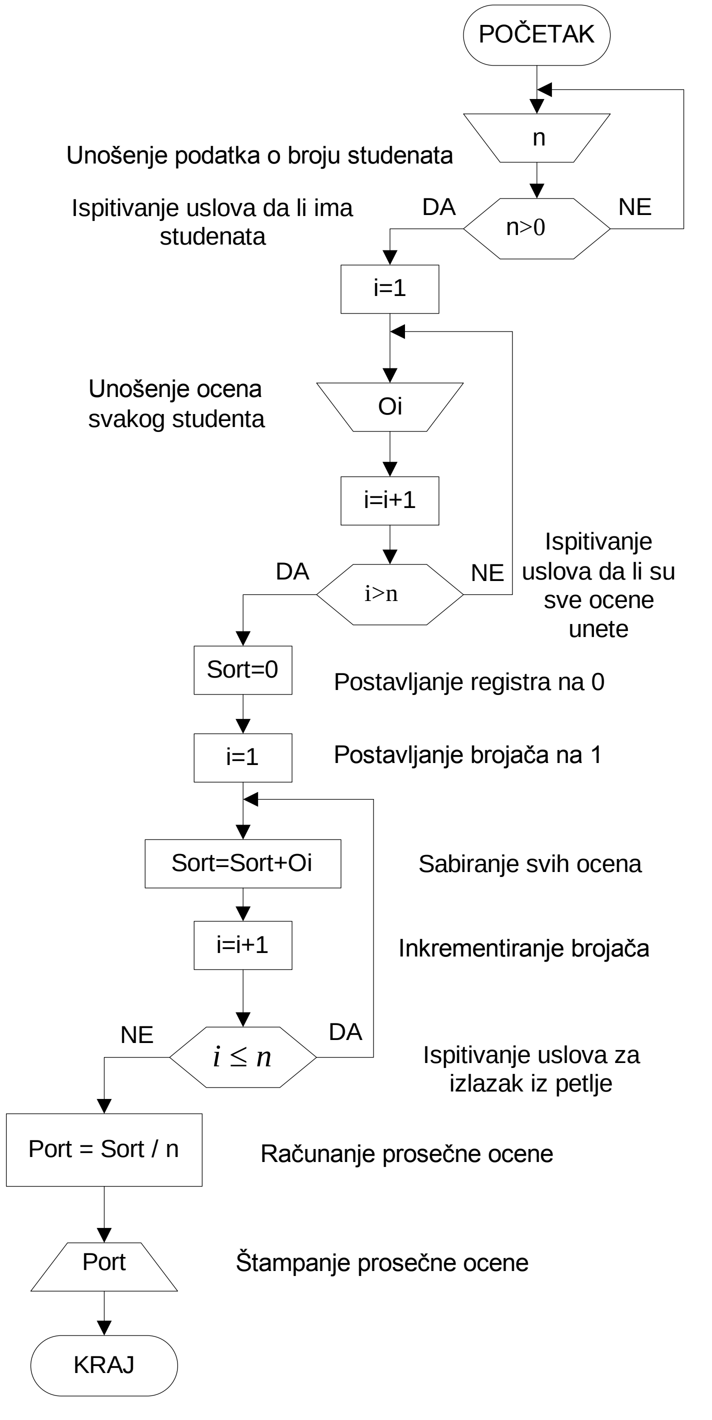
Primer 20. Zadata su dva prirodna broja k i m (k<m). Sastaviti algoritam koji određuje broj kombinacija k-te klase od m elemenata.

Rešenje: Broj kombinacija k-te klase od m elemenata: $C = \dfrac{k!}{m!(k-m)!}$


POČETAK
k,m
Unos podataka
NE
DA
k < 0 i
m < 0
Provera ispravnosti unetih podataka
F1=1
i=0
i=i+1
F1=F1· i
Računanje k!
DA
NE
i < k
Provera uslova za izlazak iz petlje
F2=1
i=0
i=i+1
F2=F2·i
Računanje m!
DA
NE
i < m
Provera uslova za izlazak
iz petlje
n=k-m
F3=1
i=0
i=i+1
F3=F3·i
Računanje
(k-m)!
DA
NE
i < n
Provera uslova za izlazak
iz petlje
C=F1/(F2·F3)
Računanje broja kombinacija
C
Štampanje rezultata
KRAJ

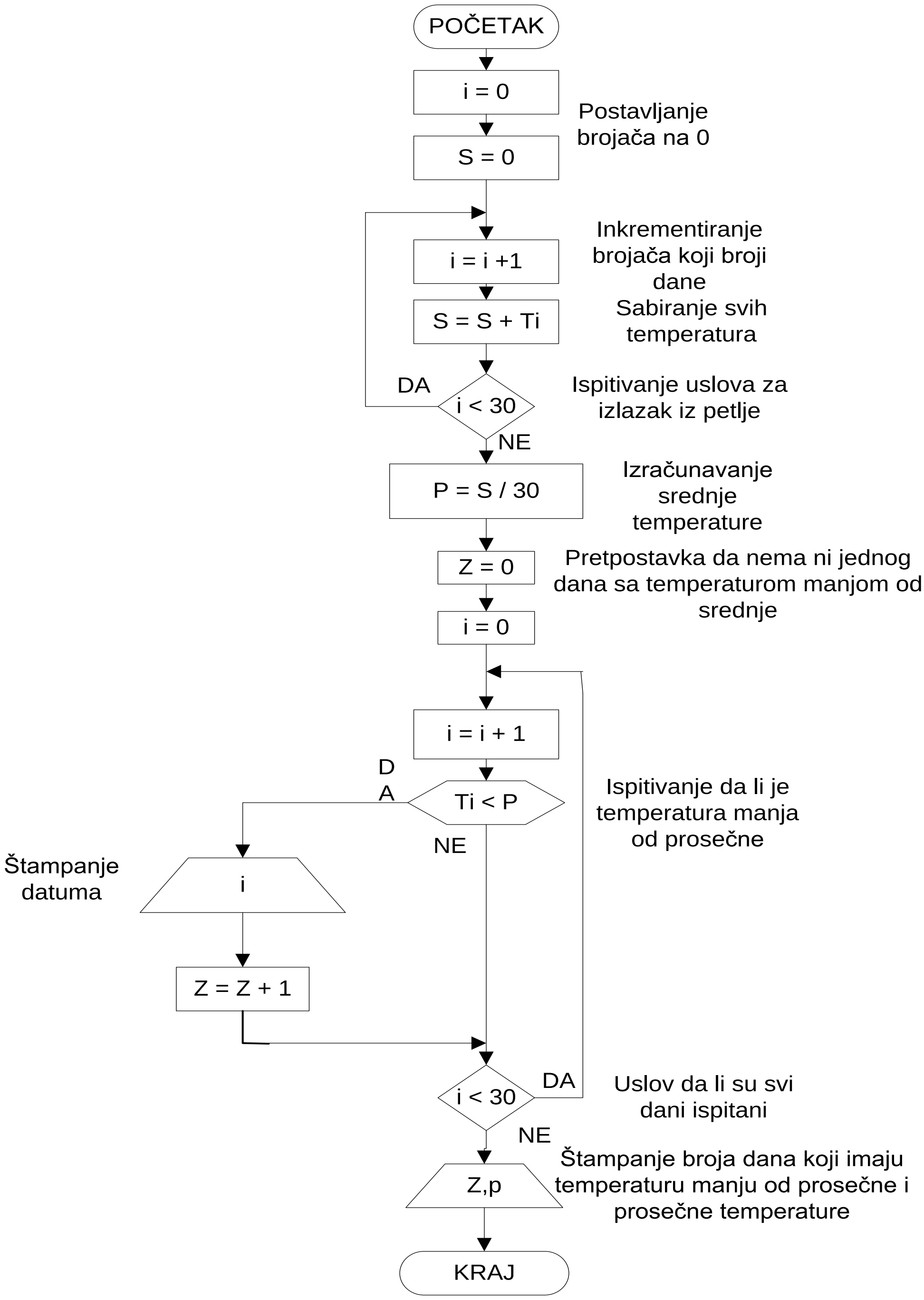
Primer 21. Napraviti algoritam koji izračunava prosečnu srednju ocenu na ispitu iz ORT-a.

Rešenje:



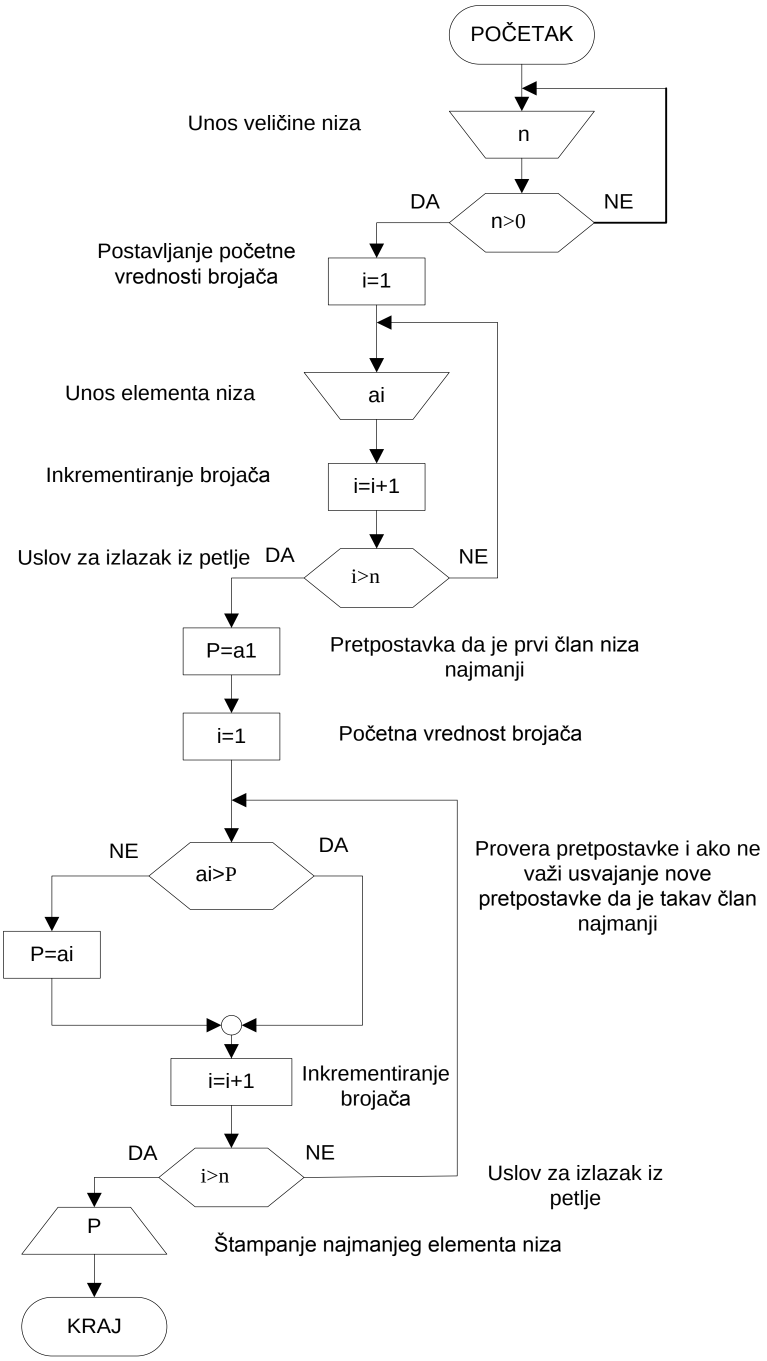
Primer 22. U datoteci se nalazi 30 brojeva koji predstavljaju temperaturu od 1–30 juna 2004. godine (T_1- T_{30}). Sastaviti algoritam koji izračunava prosečnu temperaturu u mesecu junu, štampa datum dana u kojima je temperatura bila manja od prosečne i broj dana kada je temperatura bila manja od prosečne.

Rešenje:



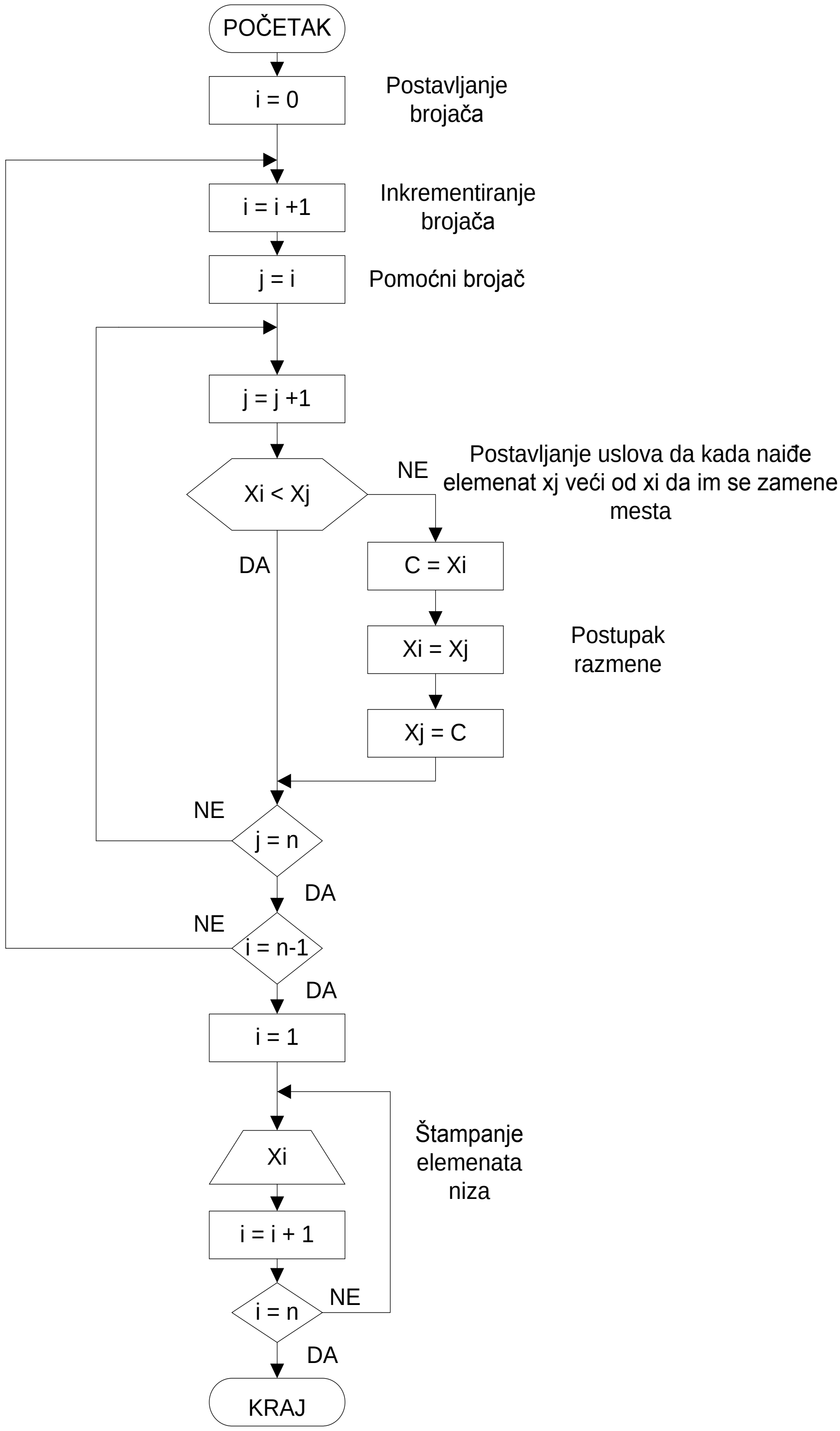
Primer 23. Primer 23:Napraviti algoritam koji učitava niz a_i, pronalazi i štampa najmanji elemenat niza.

Rešenje:



Primer 24. Napraviti algoritam koji dati niz dimenzije n, uređuje u rastućem redosledu.

Rešenje:



Primer 25. Napraviti algoritam koji određuje rešenja kvadratne jednačine $ax^2+bx+c=0$.

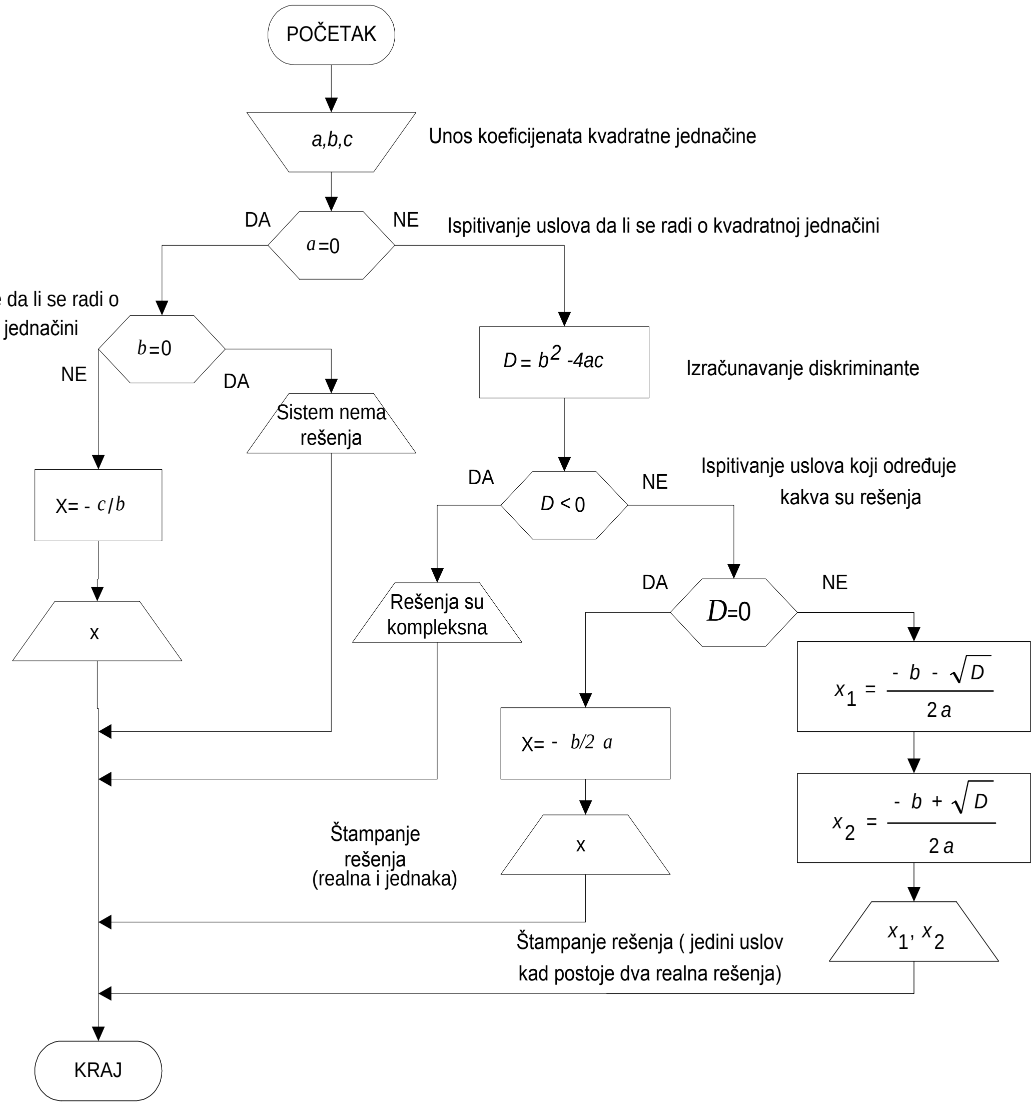
Rešenje:

$a\neq0$, uslov da je jednačina kvadratna; Diskriminanta: $D=b^2-4ac$

Realna i različita rešenja: $x_1=\dfrac{-b-\sqrt{D}}{2a}$ i $x_2=\dfrac{-b+\sqrt{D}}{2a}$ (a,b,c≠0)

Realna i jednaka rešenja: $x=\dfrac{-b}{2a}$ (a,b,c≠0 i $b^2=4ac$).

Rešenja su konjugovano kompleksna ($b^2-4ac<0$). Jednačina nije kvadratna, x=-c/b (a=0 i b≠0).



Primer 26. Sastaviti algoritamsku šemu za rešavanje i štampanje rešenja sistema linearnih jednačina ax + by = p i mx + ny = q

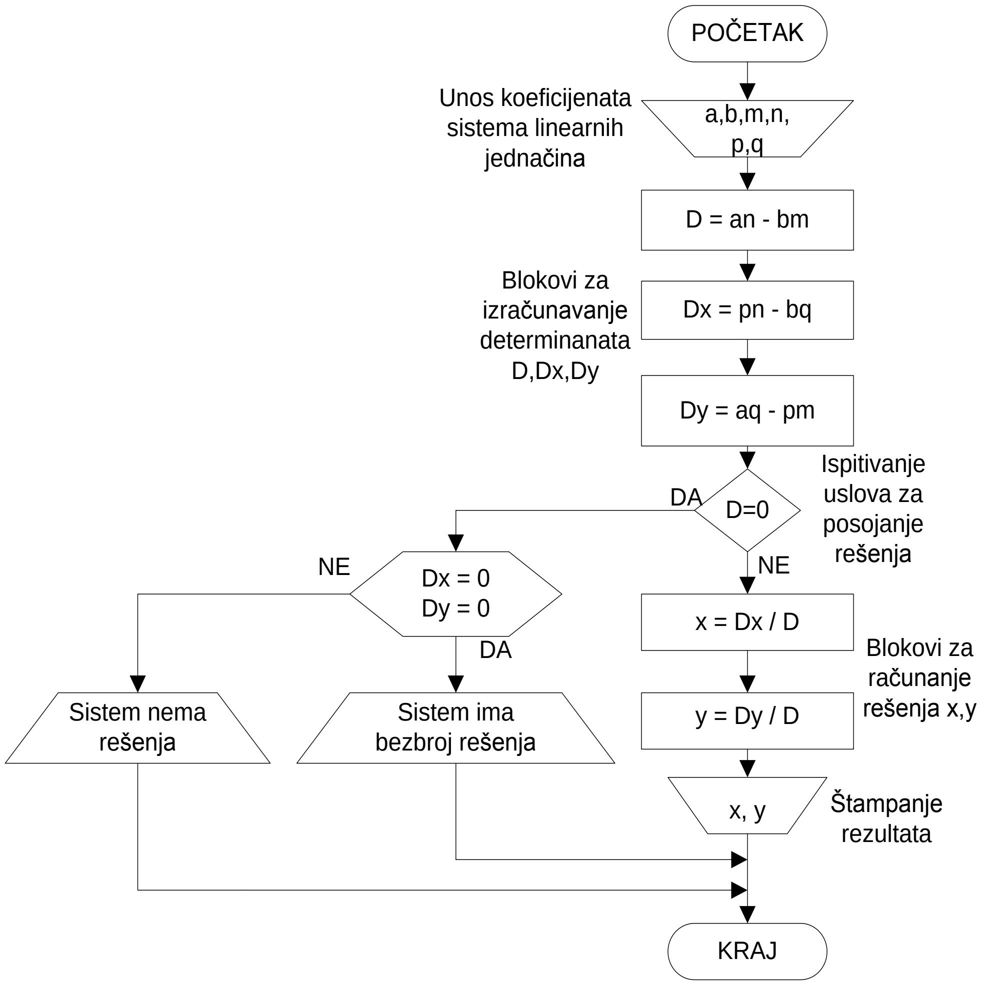
Rešenje:

ax + by = p, mx + ny = q,

$$x = \frac{Dx}{D} \quad , \; y = \frac{Dy}{D}$$

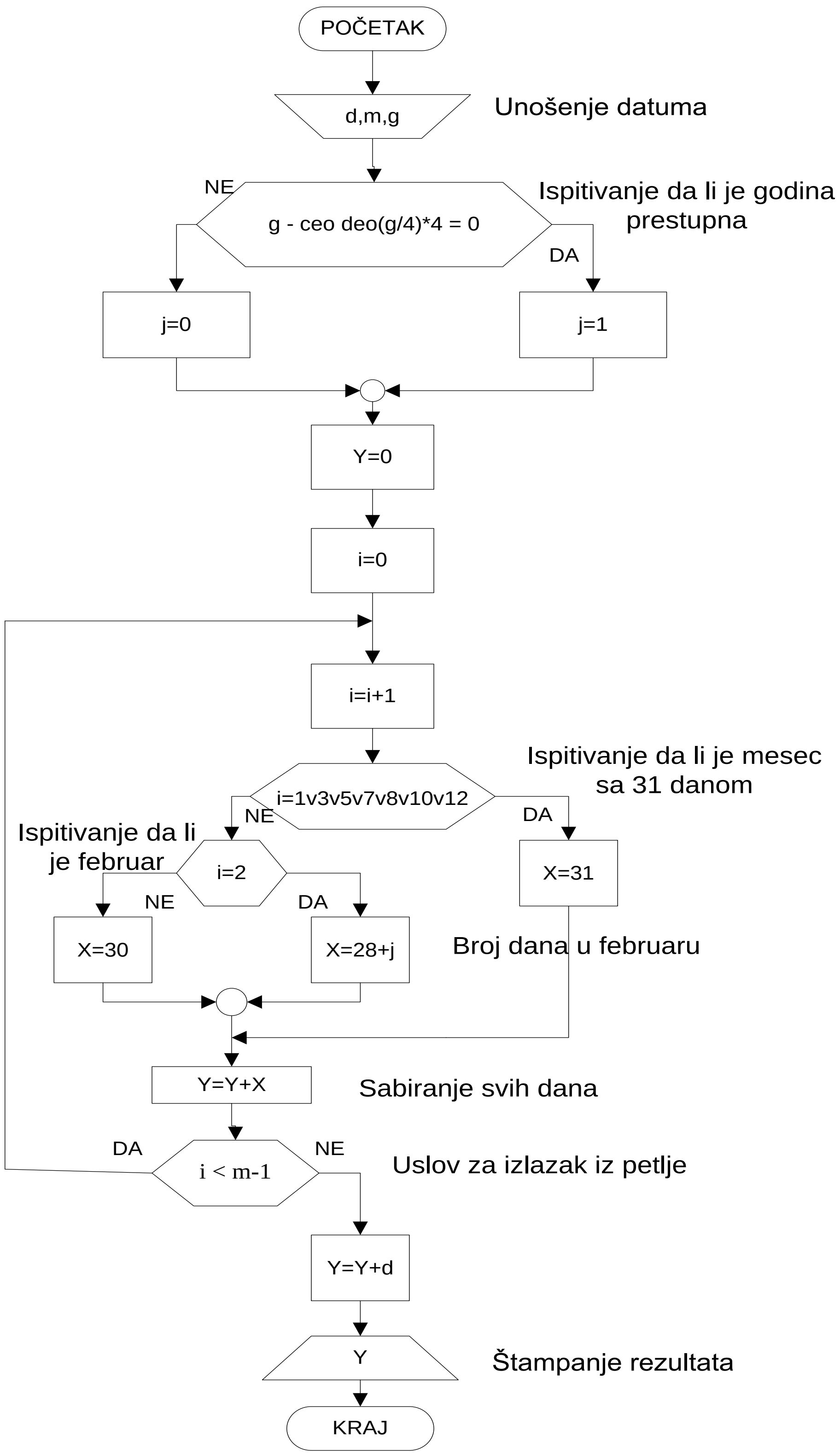
$$D = \begin{pmatrix} a & b \\ m & n \end{pmatrix} = an - bm \qquad D_x = \begin{pmatrix} p & b \\ q & n \end{pmatrix} = pn - bq \qquad D_y = \begin{pmatrix} a & p \\ m & q \end{pmatrix} = aq - pm$$

D=0 ; Dx ≠0 ; Dy ≠0 sistem nema rešenje. D =0 ; Dx = 0 i Dy = 0 sistem ima bezbroj rešenja



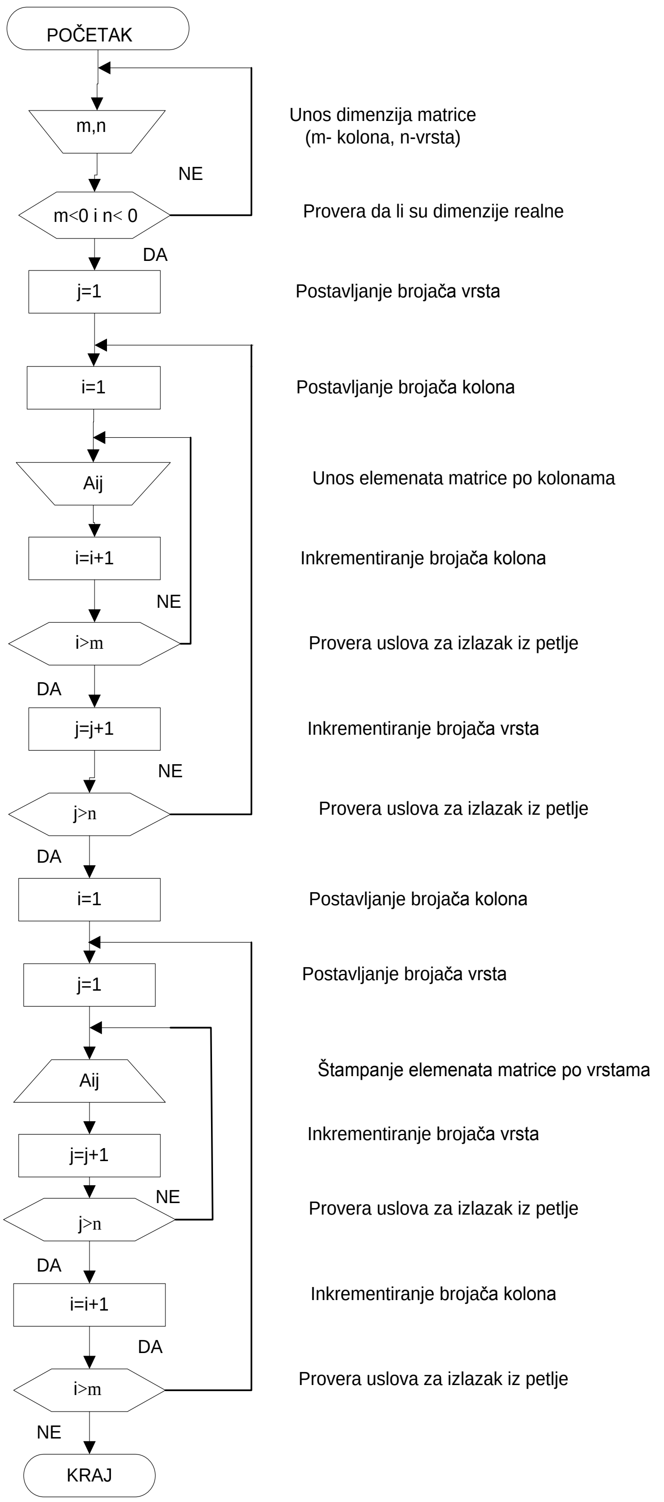
Primer 27. Napraviti algoritam koji učitava datum u obliku d,m,g ; što znači dan, mesec, godina, a zatim izračunava i štampa koji je to dan u godini.

Rešenje:



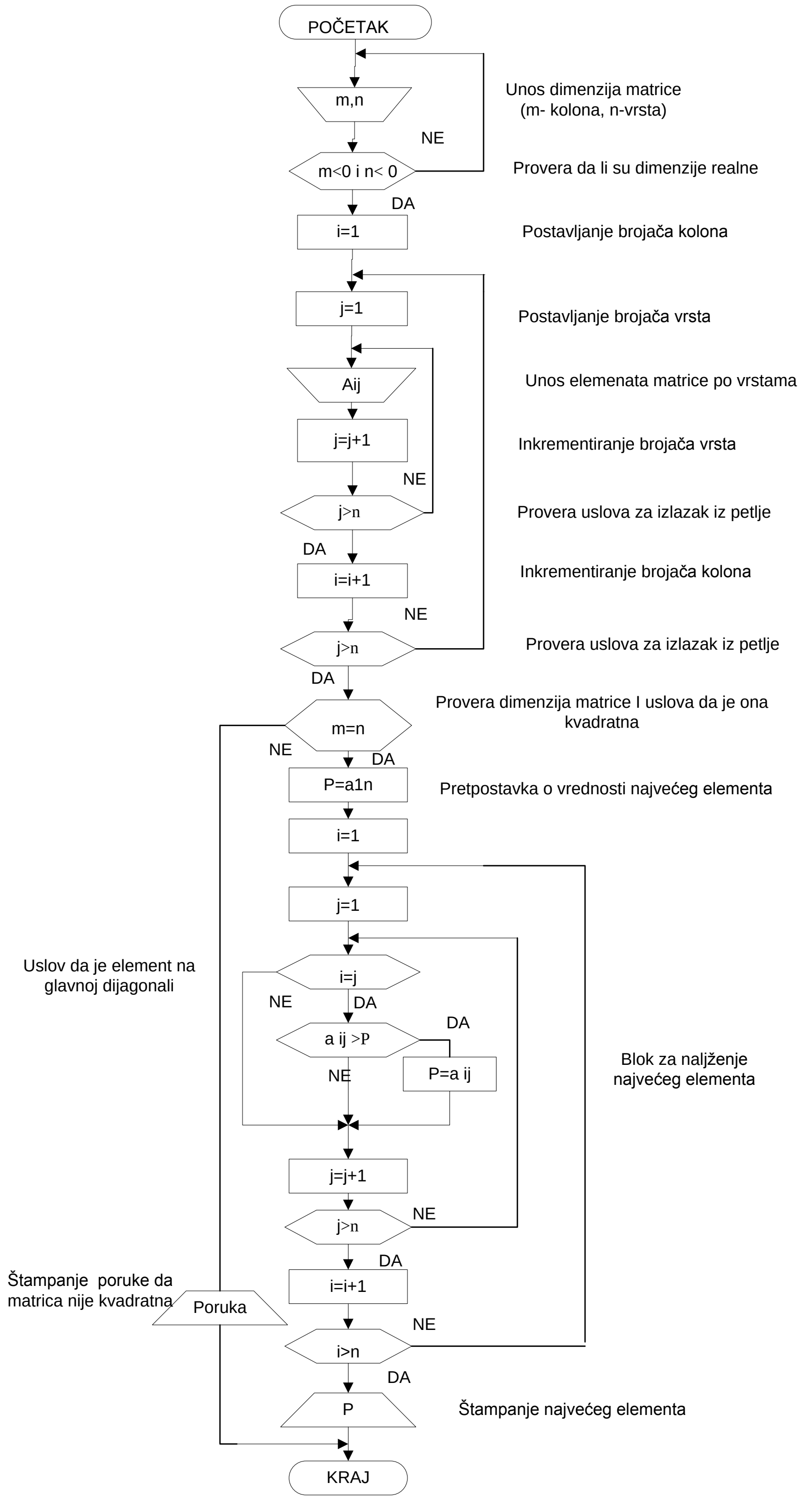
Primer 28. Napraviti algoritam koji učitava elemente matrice kolonu po kolonu, a štampa matricu vrstu po vrstu.

Rešenje:



Primer 29. Napraviti algoritam koji učitava elemente kvadratne matrice (m=n) vrstu po vrstu, nalazi i štampa najveći element na glavnoj dijagonali.

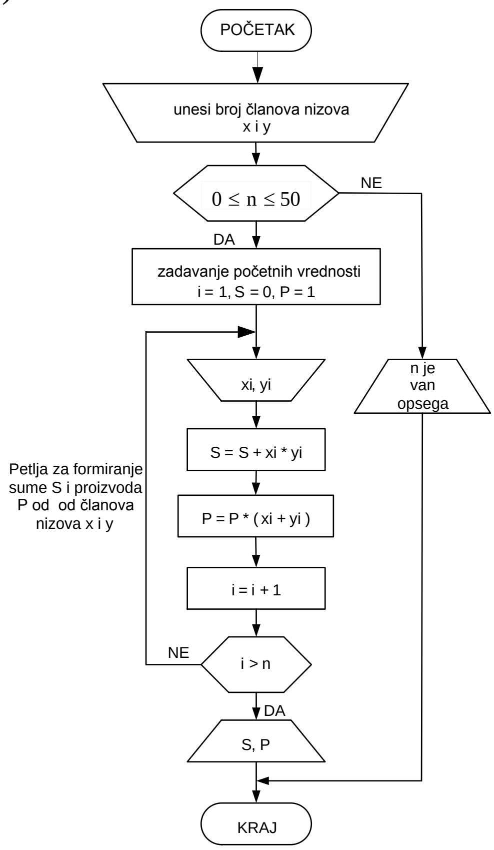
Rešenje:



Primer 30. Nacrtati algoritam za program koji učitava n parova brojeva ($1 \leq n \leq 50$) x_i i y_i ($i = 1, 2,..,n$) i izračunava sumu proizvoda $S = \sum(x_i \ y_i)$ i proizvod sume $P = \prod(x_i + y_i)$.

Rešenje:

Izračunavanje S i P obavlja se u petlji. Broj prolazaka kroz petlju određen je brojem n članova nizova *x* i *y*. Broj parova n unosi se na početku programa. Ukoliko se uneta vrednost ne nalazi u zadatom opsegu, program generiše izveštaj i skače na kraj. Kad se broj n nalazi u zadatom opsegu, novi parovi brojeva (x_i y_i) se učitavaju u petlji pre formiranja sume proizvoda (S), odnosno proizvoda suma (P).



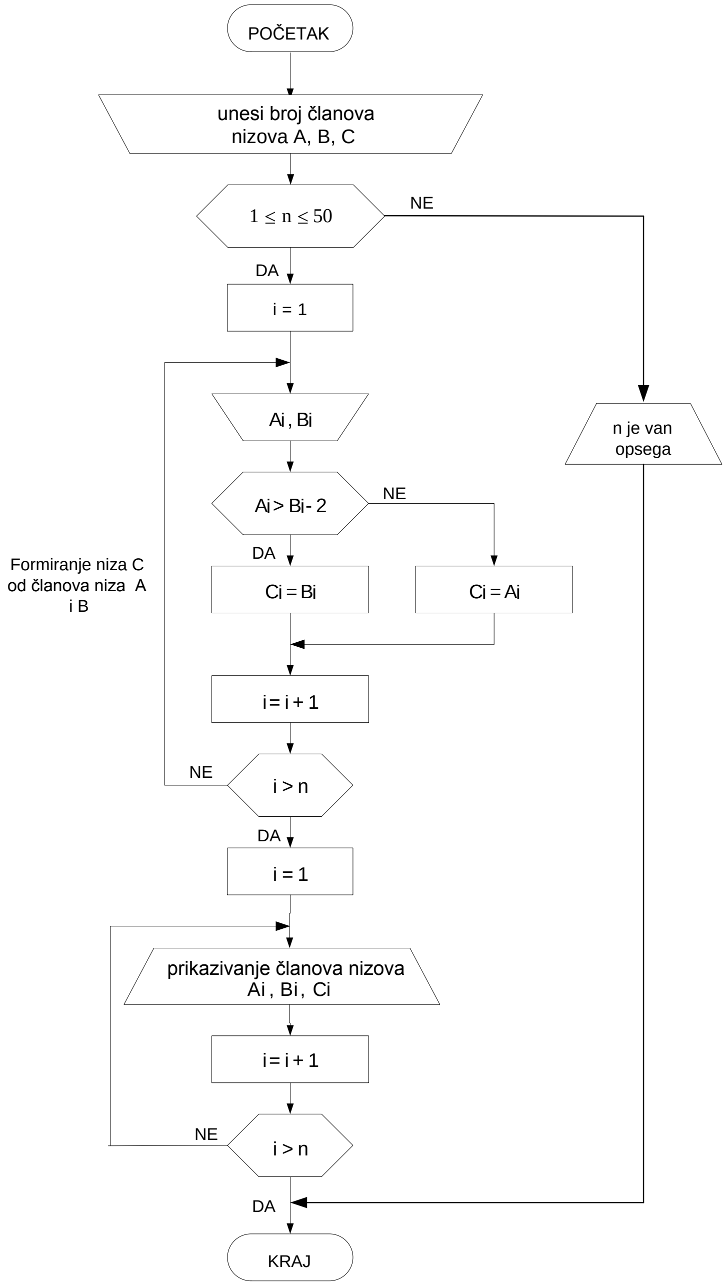
Primer 31. Nacrtati algoritam za program koji od dva niza A i B koji imaju po $1 \leq n \leq 15$ brojeva formira niz C sa elementima:

$$C_i = A_i,\ A_i \leq B_i - 2,\ i = 1, 2, ...,n \ \text{ i } C_i = B_i,\ A_i > B_i - 2,\ i = 1, 2,...,n$$

i daje izveštaj u kome se prikazuju članovi niza A_i, B_i i C_i.

Rešenje:

Na početku programa unosi se vrednost za n i to predstavlja broj članova nizova A, B i C. Ukoliko se uneta vrednost za n ne nalazi u zadatom opsegu, program generiše izveštaj i skače na kraj.Algoritam treba da sadrži dve petlje. U prvoj petlji učitavaju se vrednosti nizova A i B, na osnovu čega se formira niz C, dok se u drugoj petlji omogućava formiranje izveštaja u kome se prikazuju svi elementi nizova A, B i C.

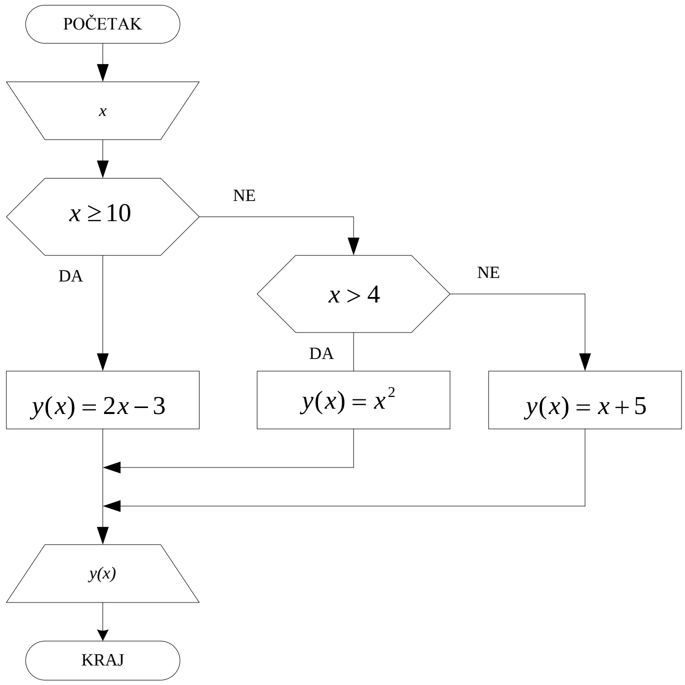


Primer 32. Nacrtati algoritam programa za izračunavanje i prikazivanje funkcije *y(x)*, gde je *x* zadata celobrojna ulazna promenljiva:

$$y(x) = \begin{cases} 2x-3 & x \geq 10 \\ x^2 & 4 < x < 10 \\ x+5 & x \leq 4 \end{cases}$$

Rešenje:

U ovom primeru se primenjuje struktura sa višestrukim odlučivanjem koja određuje putanju izvršavanja programa u zavisnosti od vrednosti ulazne promenljive *x*.



Primer 33. Nacrtati algoritam programa koji izračunava i prikazuje vrednosti polinoma *f(x)*:

$$f(x) = a_0 + a_1 x + a_2 x^2 + ... + a_n x^n \quad 0 \le n \le 20$$
$$x = x_0 + k\Delta x \qquad k = 0,1,..., m$$

Poznate su ulazne vrednosti: n, a_0, a_1, ... a_n, x_0 i Δx i m.

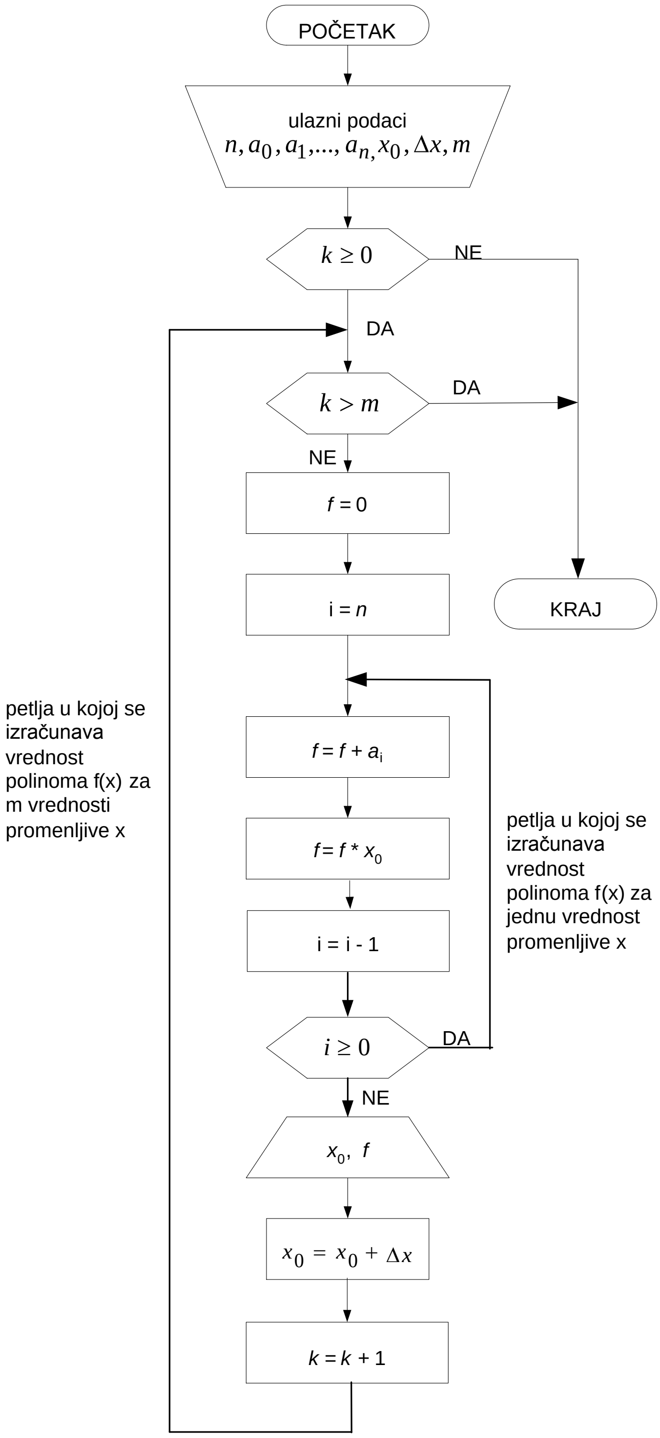
Rešenje:

Za organizovanje programskog ciklusa za izračunavanje polinoma *f(x),* pogodno je da se polinom zapiše u obliku:

$$f(x) = a_0 + x(a_1 + x(a_2 + ...x(a_{n-1} + xa_n)))$$

Zbog toga algoritam treba da sadrži dva koncentrična ciklusa izračunavanja, odnosno dve petlje. U unutrašnjoj petlji vrši izračunavanje vrednosti polinoma za $x = x_0$, počev od xa_n.

Unutrašnji ciklus se izvršava za $i = n, n-1, ..., 0$ sa korakom -1. U spoljašnjem ciklusu se vrši postavljanje $f = 0$, izdavanje izračunate vrednosti polinoma i uvećanje vrednosti x za naredno izračunavanje. Spoljašnji ciklus se izvršava za $k = 0, 1, ..., m$ sa korakom +1, odnosno onoliko puta koliko puta treba izračunati vrednost polinoma.



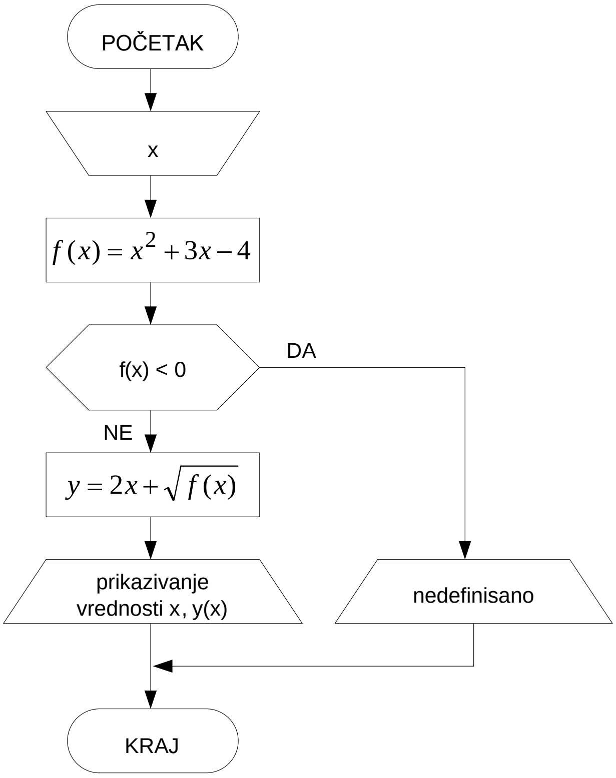
<u>Primer 34.</u> Definisati dijagram toka programa koji za zadatu vrednost promenljive x izračunava vrednost funkcije:

$$y(x) = 2x + \sqrt{x^2 + 3x - 4}$$

i prikazuje vrednosti za *x* i *y(x)* pod uslovom da je zadovoljeno $x^2+3x-4>0$. Ukoliko navedeni uslov nije zadovoljen, izlazi se iz programa i štampa se izveštaj da je stanje nedefinisano.

Rešenje:

Polinom koji se nalazi ispod korena funkcije *y(x)* može da se proglasi za funkciju *f(x)*, čija se vrednost izračunava na početku programa za unetu vrednost promenljive *x***,** a zatim se ispituje da li je zadovoljen uslov da je vrednost polinoma veća od 0. Ukoliko je uslov ispunjen nastavlja se sa izračunavanjem vrednosti funkcije *y(x)* i izdaje se izveštaj u kome se nalazi vrednost promenljive *x* i funkcije *y(x).*



<u>Primer 35.</u> Definisati dijagram toka koji za zadate celobrojne vrednosti x_i i y_i , i = 1, 2 izračunava i prikazuje vrednost funkcije Z:

$$Z=\frac{F(x_1,y_1)}{F(x_2,y_2)}$$

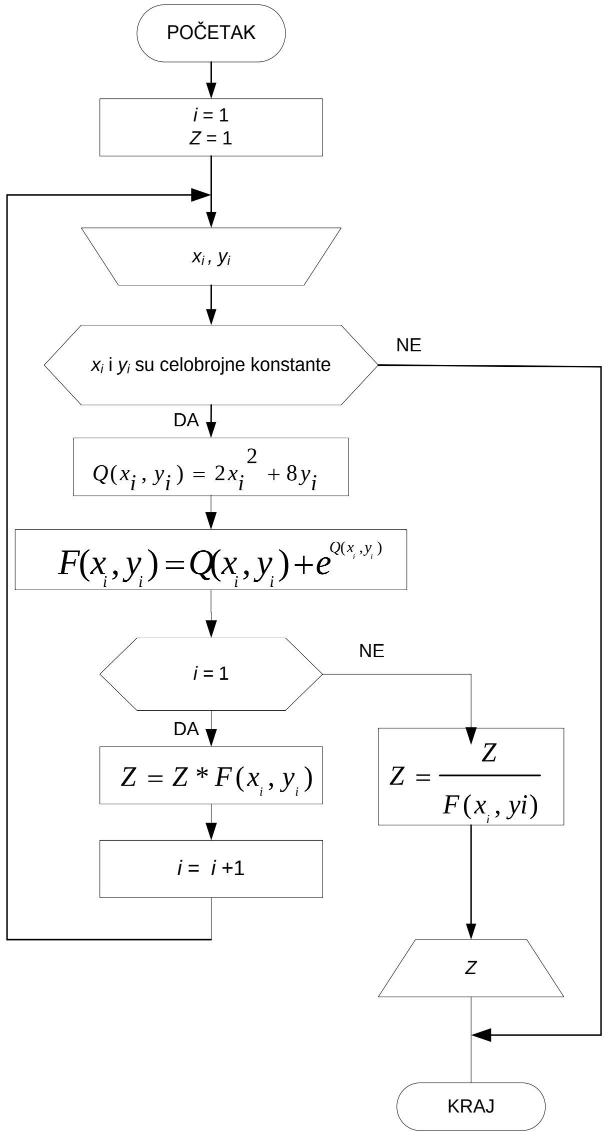
ako je $F(x,y)=2x^2+8y+e^{2x^2+8y}$

Rešenje:

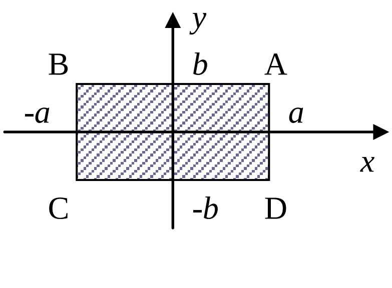
Izraz koji se ponavlja u funkciji F može da se definiše kao posebna funkcija $Q(x, y)$:

$$Q(x, y) = 2x^2 + 8y$$

tako da funkcija F(x, y) može da se zapiše u obliku: $F(x, y) = Q(x, y) + e^{Q(x,y)}$ Vrednost izraza Z može da se izračuna u petlji u kojoj se učitavaju vrednosti za parove promenljivih x i y (x_i, y_i, i = 1, 2), pri čemu se u prvom prolasku kroz petlju izrazu Z dodeljuje vrednost funkcije F(x_1, y_1), dok se u drugom prolasku tako dobijena vrednost za Z deli sa vrednošću funkcije F(x_2, y_2).

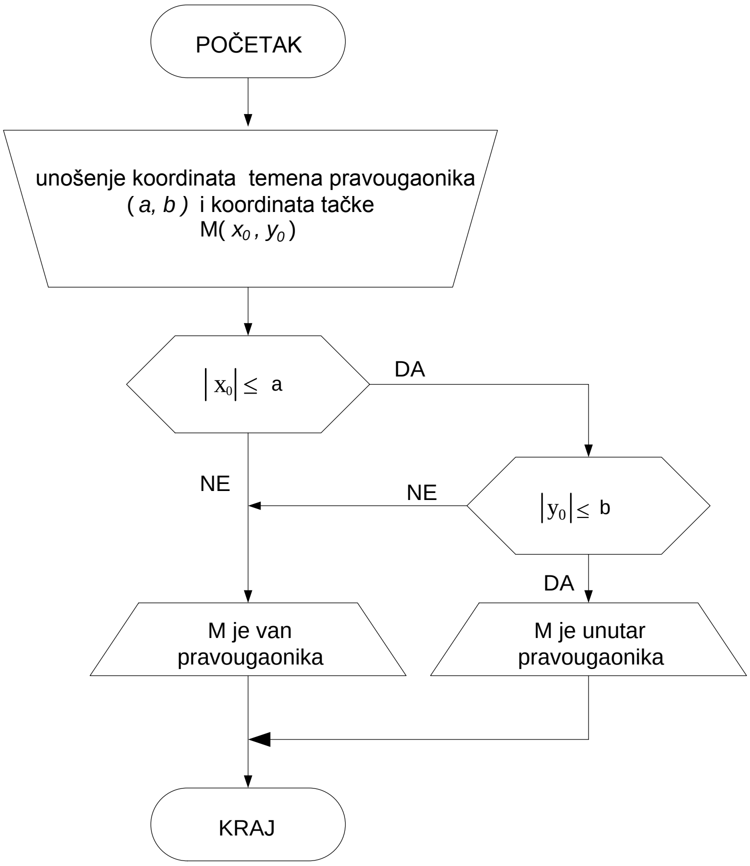


Primer 36. U pravougaonom koordinatnom sistemu zadat je pravougaonik sa temenima A, B, C i D Ako su poznate koordinate temena, nacrtati dijagram toka kojim se određuje da li je tačka M (x_0, y_0) unutar ili izvan pravougaonika.



Rešenje:

Ako su zadate veličine *a* i *b*, tada su poznate koordinate svih temena praovougaonika prikazanog na slici: A(*a*, *b*), B(-*a*, *b*), C(-*a*, -*b*) i D(*a*, -*b*). Tačka M(x_0, y_0) nalazi se unutar pravougaonika ako je $|x_0| \leq a$ i $|y_0| \leq b$.



Primer 37. Za ulazne podatke *k* i *x* izračunati vrednost trigonometrijske funkcije:

$$y(x) = \begin{cases} \sin 3x & k = 1 \\ \sin(\frac{\pi}{4} - x)\cos x & k = 2 \\ \cos(\pi + x) & k = 3 \\ 0 & k \notin \{1, 2, 3\} \end{cases}$$

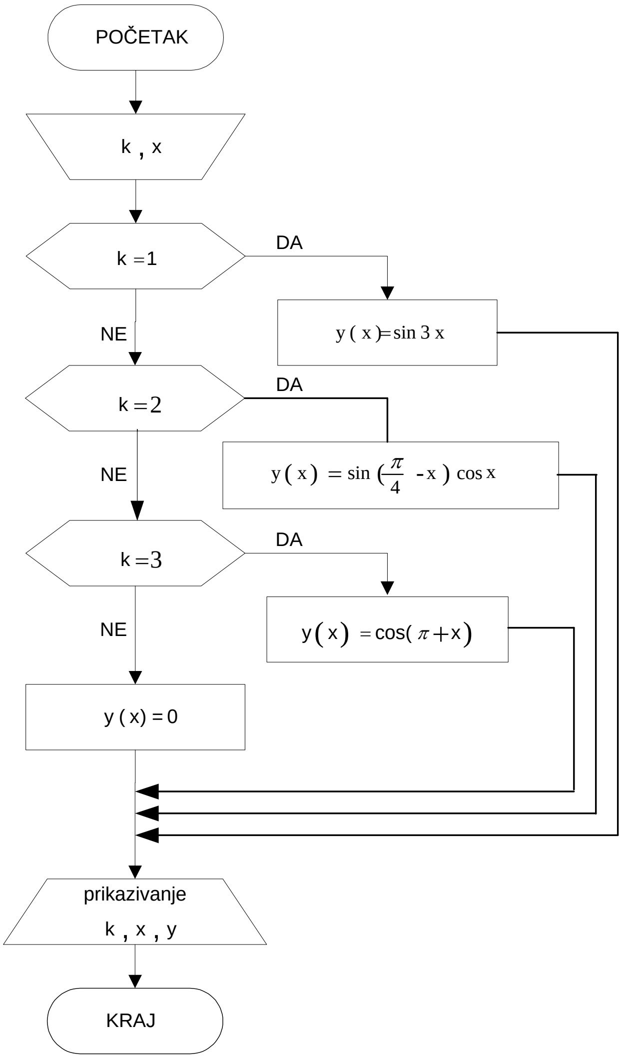
Rešenje:

Algoritam može da se realizuje kao:

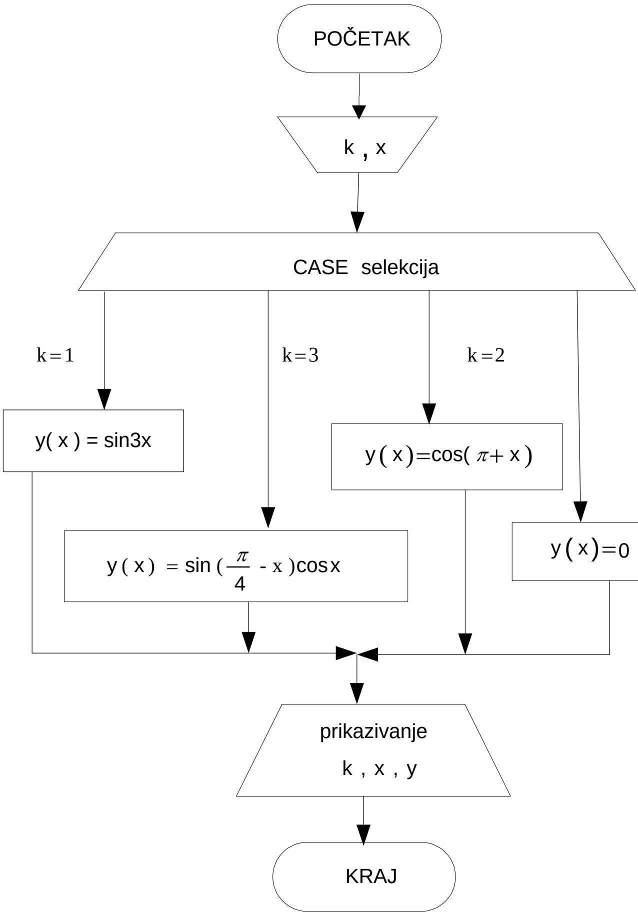
- klasična struktura sa višestrukim odlučivanjem i
- kao struktura sa CASE odlučivanjem.

U oba slučaja, na početku se učitavaju vrednosti parametra *x* i kontrolne promenljive *k* od čije vrednosti zavisi način izračunavanja vrednosti funkcije *y(x)*.

a)



b)



Primer 38. Nacrtati dijagram toka za program koji niz brojeva $a_1, a_2, ..., a_n,$ $1 \le n \le 10$ uređuje u opadajući niz.

Rešenje: Niz brojeva može da se uredi upoređivanjem svaka dva susedna broja u nizu:

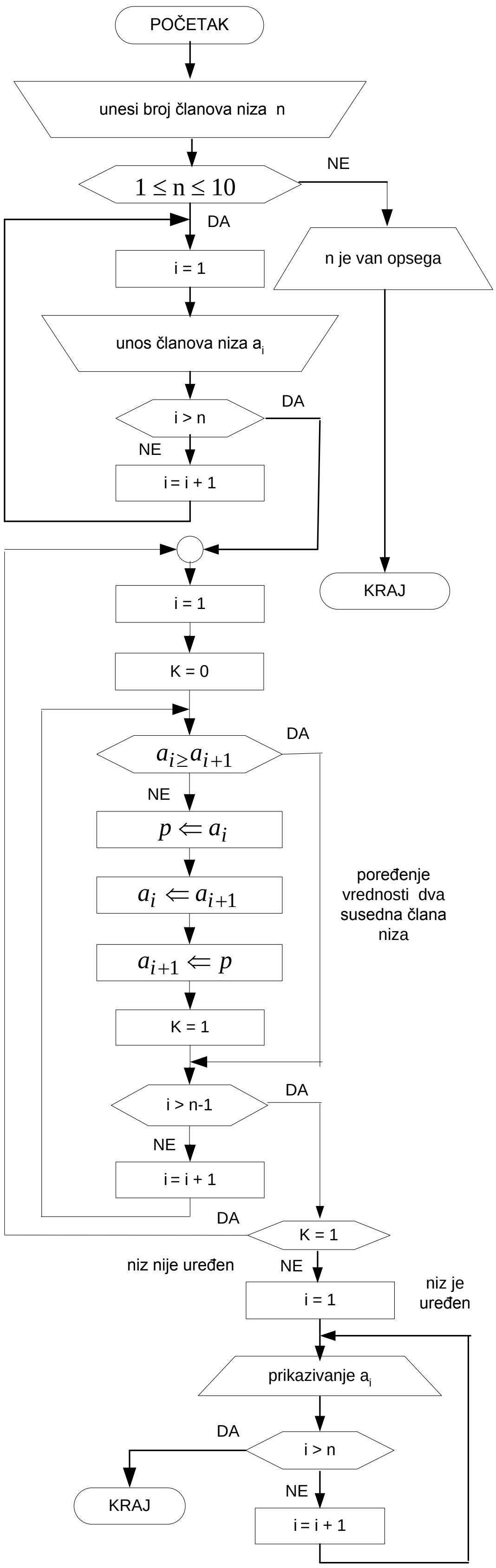
$$a_i \ge a_{i+1}, \quad i = 1, 2, ..., n$$

Mogu da nastanu dva slučaja: Brojevi a_i i a_{i+1} zadovoljavaju zahtevanu relaciju, pa takve članove niza ne treba premeštati. Brojevi a_i i a_{i+1} ne zadovoljavaju zahtevanu relaciju, pa takvim brojevima treba promeniti mesta u nizu.

Razmena mesta članova u nizu može da se uradi na sledeći način:

$$p \Leftarrow a_i$$
$$a_i \Leftarrow a_{i+1}$$
$$a_{i+1} \Leftarrow p$$

Algoritam treba da se sastoji iz dve koncentrične petlje: U unutrašnjoj petlji se vrši uzajamno poređenje dva susedna člana niza. U spoljašnjoj petlji se odlučuje da li je završeno uređivanje članova i u tu svrhu može da se koristi neka pomoćna promenljiva K koja predstavlja indikaciju da li je došlo do promene mesta među članovima niza. Pre prolaska kroz niz postavi se vrednost promenljive $K = 0$, a zatim ako dođe do promene mesta članova niza postavi se $K = 1$. Ako je po izlasku iz unutrašnjeg ciklusa $K = 1$, nastavlja se uređenje niza, a ako je $K = 0$ niz je uređen i izlazi se iz programa, odnosno prikazuje se ceo uređeni niz.


POČETAK
unesi broj članova niza n
$1 \le n \le 10$
NE
DA
n je van opsega
i = 1
unos članova niza a_i
i > n
DA
NE
i = i + 1
KRAJ
i = 1
K = 0
$a_i \ge a_{i+1}$
DA
NE
$p \Leftarrow a_i$
$a_i \Leftarrow a_{i+1}$
$a_{i+1} \Leftarrow p$
poređenje vrednosti dva susedna člana niza
K = 1
i > n-1
DA
NE
i = i + 1
DA
K = 1
niz nije uređen
NE
i = 1
niz je uređen
prikazivanje a_i
DA
i > n
KRAJ
NE
i = i + 1

Primer 39. Izračunati vrednost funkcije $f(x) = e^{-4x} \cos 4x$ u zadatom intervalu nezavisne promenljive *x* [x_0, x_n] i sa zadatim priraštajem Δx ($\Delta x \geq 0$). Nakon svakog izračunavanja prikazati u izveštaju vrednost promenljive *x* i odgovarajuću vrednost funkcije *f(x)*.

Rešenje:

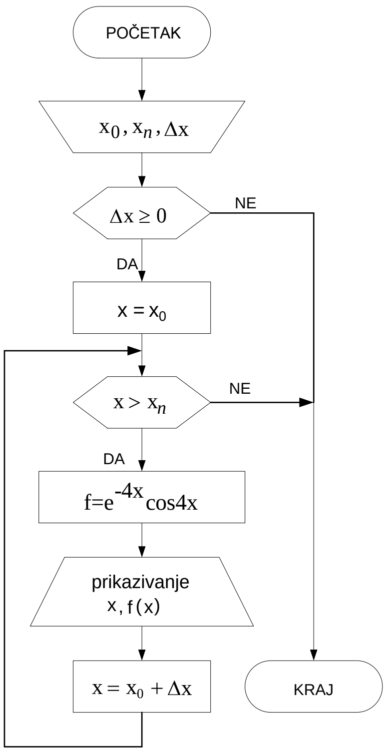
Vrednost funkcije *f(x)* određuje se u petlji, počev od vrednosti x_0. Petlja se izvršava sve dok promenljiva *x* ne dostigne vrednost x_n, s tim što se pri svakom prolasku kroz petlju vrednost promenljive *x* uvećava za Δx. Na početku programa treba uneti vrednosti za x_0, x_n i Δx, pri čemu je uslov za izvršavanje petlje $x_0 \leq x_n$. Pošto je vrednost priraštaja $\Delta x \geq 0$, broj ciklusa izvšavanja petlje je prirodan broj koji direktno zavisi od vrednosti priraštaja Δx.

U slučaju da je $\Delta x = 0$, petlja se izvršava samo jednom, odnosno dobija se samo jedna vrednost funkcije *f(x)*, jer je tada $x = x_0$.

Ako je $\Delta x > 0$, broj prolazaka kroz petlju određuje se kao količnik razlike maksimalne (x_n) i minimalne (x_0) vrednosti promenljive x i priraštaja Δx:

$$\frac{x_n - x_0}{\Delta x}$$

Broj različitih vrednosti funkcije *f(x)* u zavisnosti od promenljive *x* odgovara broju prolazaka kroz petlju. Program se završava kada je $x = x_n$. Predviđeno je da se vrednost funkcije *f(x)*i odgovarajuća vrednost promenljive *x* prikazuju pri svakom prolasku kroz petlju.

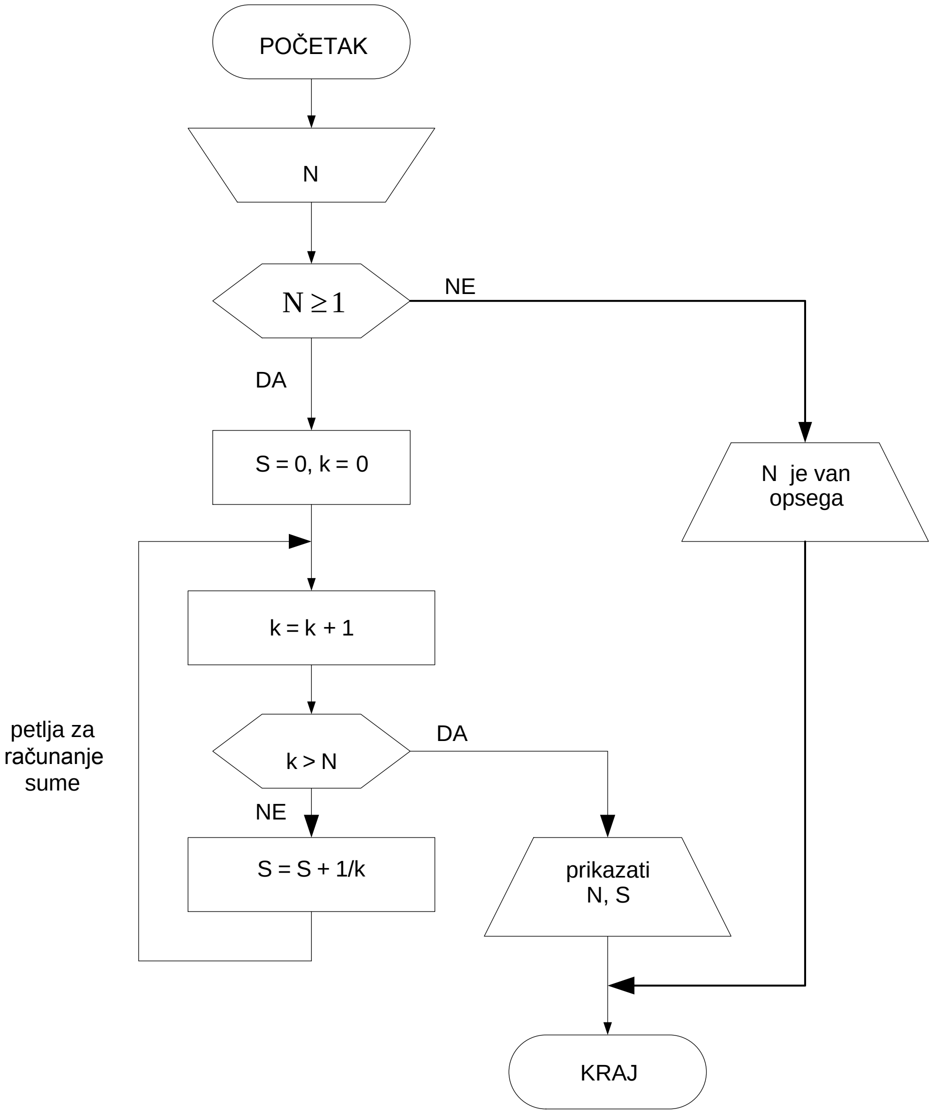


Primer 40. Nacrtati dijagram toka programa koji za zadatu vrednost promenljive $N \geq 1$ računa sumu:

$$S = \sum_{k=1}^{N} \frac{1}{k}$$

Rešenje:

Pošto je N poznata vrednost koja se učitava na početku programa, tražena suma može da se izračuna jedino ako je N u dozvoljenom opsegu, što se ispituje na početku programa. Ako je N van opsega, formira se poruka i izlazi se iz programa. Ako je N u dozvoljenom opsegu, vrednost sume S računa se u petlji u kojoj se inkrementira vrednost promenljive k sve dok je $k \leq N$. Po izlasku iz petlje prikazuje se zadata vrednost N i suma S.



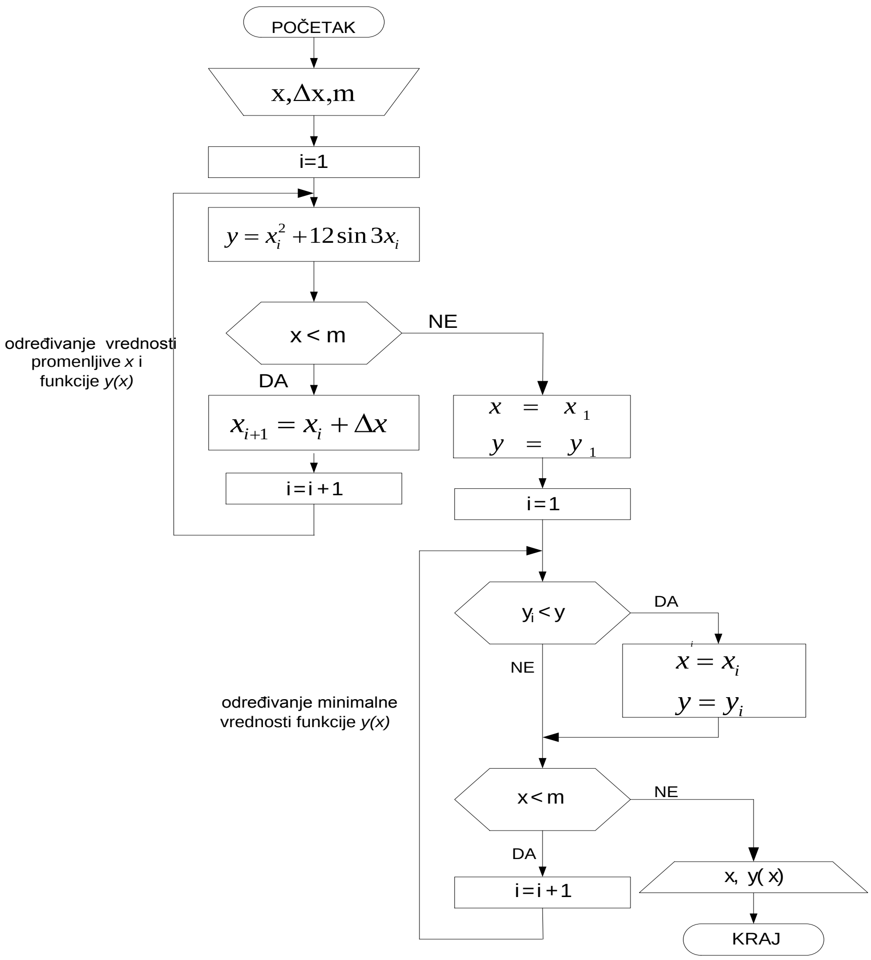
Primer 41. Nacrtati dijagram toka programa koji računa *m* vrednosti funkcije *y(x)*:

$$y(x) = x^2 + 12\sin 3x$$

ako se promenljiva *x* menja počev od vrednosti x_0 sa priraštajem Δx, a zatim nalazi minimalnu vrednost funkcije *y(x)* i vrednost promenljive *x*.

Rešenje:

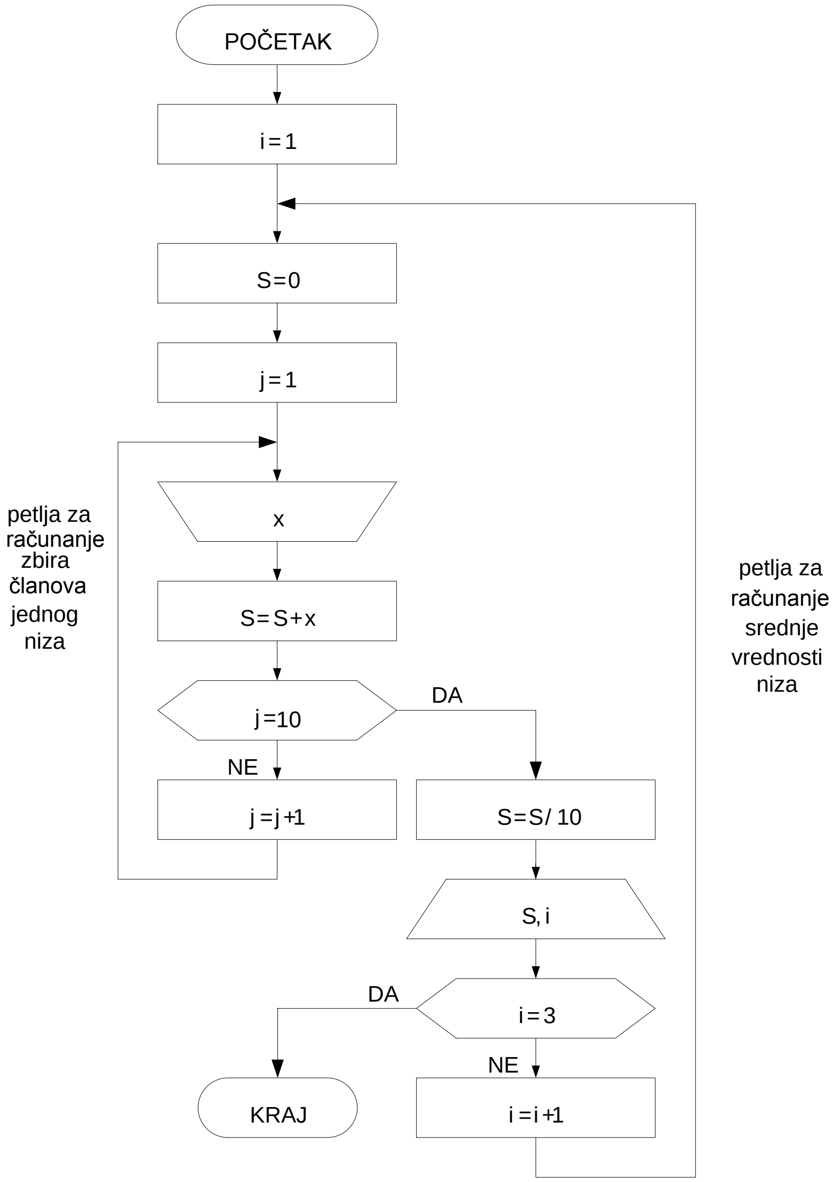
Dijagram toka sadrži dve nezavisne petlje. U prvoj se obavlja izračunavanje vrednosti funkcije *y*, i tada se dobija *m* parova (x_1, y_1), (x_2, y_2),...,(x_m, y_m). U drugoj petlji se određuje par (*x*, *y*) koji se sastoji od minimalne vrednosti funkcije *y* i odgovarajuću vrednosti promenljive *x*, pri čemu se polazi od pretpostavke da je prva izračunata vrednost funkcije y_1 dobijena za $x = x_1$ najmanja vrednost funkcije. Ukoliko se utvrdi da je neka vrednost funkcije y_i manja, onda se ova vrednost uzima kao nova najmanja, kojoj odgovara promenljiva x_i. Na kraju ispitivanja par (*x*, *y*) predstavlja traženu najmanju vrednost funkcije *y* i odgovarajuću vrednost promenljive *x*.



Primer 42. Neka su zadata tri niza brojeva, od kojih svaki sadrži po 10 brojeva. Nacrtati dijagram toka za program koji izračunava i prikazuje srednju vrednost svakog niza.

Rešenje:

Dijagram toka treba da sadrži dve koncentrične petlje. U unutrašnjoj petlji se učitava 10 članova niza i računa njihov zbir, dok se u spoljašnjoj petlji računa srednja vrednost niza i prikazuje u izveštaju. Spoljašnja petlja se izvršava tri puta, nakon čega se program završava.



Primer 43. Definisati algoritam koji za zadato *x* izračunava $\sqrt{x}$ po Njutnovoj iterativnoj formuli:

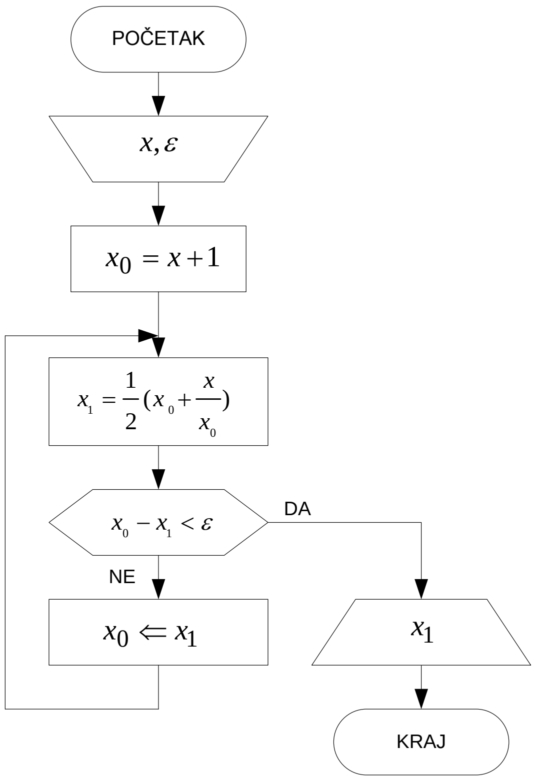
$$x_{i+1} = \frac{1}{2}(x_i + \frac{x}{x_i}) \quad i = 0,1,2,...$$

gde je $x_0 = x + 1$. Proces izračunavanja se prekida kada se dostigne zadata tačnost ε, tako da je $x_i - x_{i+1} < \varepsilon$.

Rešenje:

Ulazne veličine su x i ε, dok je izlazna veličina izračunat kvadratni koren iz *x*. Prema primenjenoj metodi to će biti x_{i+1} koji se izračunava u poslednjoj iteraciji. Za iterativno računanje mogu da se koriste samo dve promenljive x_0 (prehodna vrednost) i x_1 (naredna vrednost).

Ovde se koristi iterativni ciklus, što znači da je broj ponavljanja ciklusa nepoznat pre izvršenja programa. On zavisi od brzine konvergencije iterativnog postupka i zadate tačnosti (ε).



ZADACI ZA SAMOSTALAN RAD

1. Napisati dijagram toka programa koji izračunava srednju vrednost n brojeva.
2. Napisati algoritam koji izračunava m-ti stepen celog broja a.
3. Napisati dijagram toka stanja koji proverava da li je broj deljiv sa 15.
4. Nacrtati dijagram toka programa koji određuje rešenja sistema jednačina za N zadatih vrednosti promenljivih x i y:

$$a_1 x_i + b_1 y_i = c_1$$
$$a_2 x_i + b_2 y_i = c_2$$

5. Sastaviti algoritam koji za dato x izračunava y prema navedenoj formuli:

$$y = \begin{cases} -4, \quad x < 0 \\ x + 2, \quad 0 \leq x < 1 \\ 3x - 1{,}1, \quad 1 \leq x < 5 \\ 2x, \quad x \gg 5 \end{cases}$$

6. Nacrtati dijagram toka programa koji računa vrednost funkcije:

$$z = \sin(x+5) + \cos 2x$$

ako se vrednosti promenljive x nalaze u opsegu [x_0, x_m], pri čemu se x menja sa korakom Δx. Program se završava kada je izračunato K vrednosti funkcije ili ako je promenljiva x dostigla svoju maksimalnu vrednost x_m.

7. Realizovati dijagram toka programa koji u zavisnosti od vrednosti kontrolne promenljive C i poznatog poluprečnika kružnice r određuje obim kruga O (C = 1), površinu kružne površi P_K (C = 2) ili površinu kvadrata oko koga je opisana data kružnica S (C = 3). Program testira u petlji vrednost kontrolne promenljive C koja se učitava sa tastature i završava se izveštajem o brojnoj vrednosti tražene izračunate veličine kada promenljiva C dobije jednu od definisanih vrednosti.
8. Odrediti vrednosti funkcije Z_i (i = 1,...,8):

$$Z_i(x) = \sqrt{\ln(x_i)\sin^2 x_i}$$

Pri promeni indeksa i u intervalu $1 \leq i \leq 4$ argument x se povećava od vrednosti x_1 = 1.2 za $\Delta x = 0.3$, dok se u intervalu $5 \leq i \leq 8$ povećava za $\Delta x = 0.2$.

9. Na osnovu zadate formule

$$y_{i+1} = -y_{i-1} + y_i(2 - a^2 \Delta x^2)$$

odrediti vrednost parametra y_i ako su zadati početni uslovi:

$$y_0 = 1 \text{ i } y_1 = 1$$
$$0 \leq x \leq 1$$

za učitane podatke $a = 0.8$ i $\Delta x = 0.05$.

10. Neka je $A_3A_2A_1A_0$ četvorocifren dekadni broj. Napisati dijagram toka koji izračunava koliko postoji četvorocifrenih brojeva kod kojih je zbir prve dve cifre jednak zbiru sledeće dve cifre, tj. $A_3 + A_2 = A_1 + A_0$.
11. Napraviti algoritam za izračunavanje zbira elemenata svake vrste i svake kolone dvodimenzionalnog niza X (n, m).

12. Napraviti algoritam koji određuje da li se krug može prekriti kvadratom ili kvadrat krugom, ako su date površine kvadrata i kruga.
13. Napravi algoritam kojim se učitava ceo broj n i realni broj a, a zatim se izračunava a^n. Algoritam mora da radi $\forall a \in$ R i $\forall\ n \in$Z.
14. Naprraviti algoritam koji određuje maksimum za n unetih brojeva.
15. Napravi algoritam koji učitava cele brojeve a i b i za njih izračunava NZS i NZD.
16. Napravi algoritam kojim se najpre učitava broj n, koji predstavlja broj članova niza, određuje se indeks najvećeg i najmanjeg člana niza, a zatim se štampaju ti indeksi i ti članovi niza.
17. Napraviti algoritam kojim se učitava broj x i realna greška **ε**, a zatim na bazi razvoja u Maklorenov red funkcije $\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - ...$ izračunati vrednost sin x sa greškom manjom od zadatog **ε.**
18. Napraviti algoritam kojim se učitava broj x i realna greška **ε**, a zatim na bazi razvoja u Maklorenov red funkcije $\cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - ...$ izračunati vrednost sin x sa greškom manjom od zadatog **ε.**
19. Napraviti algoritam kojim se učitava prirodan broj n, a zatim se proverava da li je taj broj prost, i štampa odgovarajuća poruka.
20. Napraviti algoritam koji učitava današnji datum u obliku d,m,g (dan, mesec, godina), a zatim određuje i štampa sutrašnji datum.
21. Napraviti algoritam koji određuje koliko od unetih 10 brojeva je veće od 100.
22. Napraviti algoritam koji učitava elemente kvadratne matrice (m=n) vrstu po vrstu, nalazi i štampa najmanji element ispod glavne dijagonale.
23. Napraviti algoritam kojim se ušitavaju brojevi m i n koji predstavljaju dimenzije matrice A, a zatim se učitavaju elementi matrice a_{mn} i određuje:
    a) Suma elementa na glavnoj dijagonali;
    b) Vrednost najvećeg elementa iznad glavne dijagonale;
    c) Po apsolutnoj vrednosti najveći element u sporednoj dijagonali.

LITERATURA

1. Obradović S. "Osnovi računarske tehnike", *VIŠER*, Beograd, 2014.
2. Obradović S. "Veštine dobrog programiranja", *Galeb*, Beograd, 1997.
3. Mijalković M. "Programiranje MSC96 serije mikrokontrolera", *Viša elektrotehnička škola*, Beograd, 2002.
4. Daniels J. " Digital design from zero to one", *Johan Willey&Sons, Inc.*, New York 1996.
5. Lipschutz S. "Shaum`s outline of theory and problems of essential computer mathematics", *Mc Graw-Hill Publishing Company*, New York, 1982.
6. Parezanović N. " Računari i programiranje", *Naučna knjiga*, Beograd, 1980.