

Dragana Prokin
Vera V. Petrović
Milan Mijalković

**ZBIRKA ZADATAKA IZ
OSNOVA RAČUNARSKE TEHNIKE**

Visoka škola elektrotehnike i računarstva strukovnih studija
Beograd, 2013.

Autori: dr Dragana Prokin
dr Vera V. Petrović
dr Milan Mijalković

Recezeni: dr Slobodan Obradović
dr Zoran Banjac

Izdavač: Visoka škola elektrotehnike i računarstva strukovnih studija, Beograd
Vojvode Stepe 283

Tehnička obrada: Gabrijela Dimić
Divna Mičić

Korice: Kristijan Kuk
Gabrijela Dimić

Tiraž: 50

Štampa: Školski servis Gajić

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. – 2. neizmenjeno izd. –
Beograd: Visoka škola elektrotehnike i
računarstva strukovnih studija, 2013 (Beograd
: Školski servis Gajić). – 148 str.: graf.
prikazi; 24 cm

Tiraž 50. – Bibliografija: str.148.

ISBN 978-86-7982-096-9

1.Петровић, Вера В., 1965 – [аутор] 2.

Мијалковић, Милан, 1957-[аутор]

а) Рачунарство – Задаци

COBISS.SR – ID 201821964

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:

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

S = osnova (baza) brojnog sistema

S^i = težina cifre u brojnom sistemu

i = pozicija cifre ($R, R-1, \dots, 1, 0, -1, \dots, -P$)

$a_R, a_{R-1}, \dots, a_1, a_0, a_{-1}, \dots, a_{-P}$ su cifre broja koje pripadaju skupu $\{ 0, 1, \dots, S-1 \}$

Sažeti oblik prikazivanja broja x :

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

Decimalni brojni sistem (DEC) je težinski.

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

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

$S = 10$ osnova (baza) brojnog sistema

$a_R, a_{R-1}, \dots, a_1, a_0, a_{-1}, \dots, a_{-P}$ su cifre broja koje pripadaju skupu $\{ 0, 1, \dots, 9 \}$

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

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

$S = 16$ osnova (baza) brojnog sistema

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

• Konverzija iz OCT u DEC brojni sistem

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

$S = 8$ osnova (baza) brojnog sistema

$a_R, a_{R-1}, \dots, a_1, a_0, a_{-1}, \dots, a_{-P}$ su cifre broja koje pripadaju skupu $\{ 0, 1, \dots, 7 \}$

• Konverzija iz BIN u DEC brojni sistem

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

$S = 2$ osnova (baza) brojnog sistema

$a_R, a_{R-1}, \dots, a_1, a_0, a_{-1}, \dots, a_{-P}$ 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:

$$\begin{aligned}x_{(16)} &= 2E3A_{(16)} \\x_{(10)} &= 2 \cdot 16^3 + 14 \cdot 16^2 + 3 \cdot 16^1 + 10 \cdot 16^0 = 11834_{(10)}\end{aligned}$$

$$2E3A_{(16)} \rightarrow 11834_{(10)}$$

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

Rešenje:

$$\begin{aligned}x_{(8)} &= 321_{(8)} \\x_{(10)} &= 3 \cdot 8^3 + 2 \cdot 8^2 + 1 \cdot 8^1 = 1672_{(10)}\end{aligned}$$

$$321_{(8)} \rightarrow 1672_{(10)}$$

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

Rešenje:

$$\begin{aligned}x_{(2)} &= 10111011_{(2)} \\x_{(10)} &= 1 \cdot 2^7 + 0 \cdot 2^6 + 1 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 187_{(10)}\end{aligned}$$

$$10111011_{(2)} \rightarrow 187_{(10)}$$

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

- metodom **sukcesivnih deljenja** celobrojnog dela sa osnovom brojnog sistema
- metodom **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	$\uparrow$	$0.375 \cdot 2 = 0.75$	0	$\downarrow$
$120 : 2 = 60$	0		$0.75 \cdot 2 = 1.5$	1	
$60 : 2 = 30$	0		$0.5 \cdot 2 = 1.0$	1	
$30 : 2 = 15$	0				
$15 : 2 = 7$	1				
$7 : 2 = 3$	1				
$3 : 2 = 1$	1				
$1 : 2 = 0$	1				

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

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	$\uparrow$	$0.237 * 2 = 0.474$	0	$\downarrow$
$2429 : 2 = 1214$	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				

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

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	$\uparrow$	$0.136 * 8 = 1.088$	1	$\downarrow$
$545 : 8 = 68$	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				

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

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	$\uparrow$	$0.218 * 8 = 1.744$	1	$\downarrow$
$86 : 8 = 10$	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	

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

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

Rešenje:

$$\begin{array}{r|l}
 845 : 16 = 52 & 13 = D \uparrow \\
 52 : 16 = 3 & 4 \\
 3 : 16 = 0 & 3
 \end{array}
 \qquad
 \begin{array}{r|l}
 0.631 * 16 = 10.096 & A \downarrow \\
 0.096 * 16 = 1.536 & 1 \\
 0.536 * 16 = 8.576 & 8
 \end{array}$$

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

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

Rešenje:

$$\begin{array}{r|l}
 674 : 16 = 42 & 2 \uparrow \\
 42 : 16 = 2 & 10=A \\
 2 : 16 = 0 & 2
 \end{array}
 \qquad
 \begin{array}{r|l}
 0.574 * 16 = 9.184 & 9 \downarrow \\
 0.184 * 16 = 2.944 & 2 \\
 0.944 * 16 = 15.104 & 15=F \\
 0.104 * 16 = 1.664 & 1
 \end{array}$$

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

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

Rešenje:

$$\begin{array}{r|l}
 3428 : 16 = 214 & 4 \uparrow \\
 214 : 16 = 13 & 6 \\
 13 : 16 = 0 & 13=D
 \end{array}
 \qquad
 \begin{array}{r|l}
 0.435 * 16 = 6.95 & 6 \downarrow \\
 0.95 * 16 = 15.2 & 15=F \\
 0.2 * 16 = 3.2 & 3 \\
 0.2 * 16 = 3.2 & 3
 \end{array}$$

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

- 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:

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

$$11001110,01011_{(2)} \rightarrow 316,26_{(8)}$$

Primer 12. Binarni broj $x_{(2)} = 11110010110,010101111_{(2)}$ pretvoriti u heksadecimalni, $x_{(2)} \rightarrow x_{(16)}$.

Rešenje:

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

$$11110010110,0101011111_{(2)} \rightarrow 796,57C_{(16)}$$

- 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, $x_{(8)} \rightarrow x_{(2)}$.

Rešenje:

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

$$34752,423601_{(8)} \rightarrow 011100111101010, 100010011110000001_{(2)}$$

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

Rešenje:

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

$$E1B3C6,D4F8_{(16)} \rightarrow 11100001101100111100 \mid 0110, 1101010011111000_{(2)}$$

- Konverzija brojeva iz **OCT** u **HEX** brojni sistem vrši se preko binarnog brojnog sistema:
OCT $\rightarrow$ BIN $\rightarrow$ 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 $\rightarrow$ BIN $\rightarrow$ 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, $x_{(8)} \rightarrow x_{(16)}$.

Rešenje:

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

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

$$5716,043_{(8)} \rightarrow BCE,118_{(16)}$$

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

Rešenje:

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

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

$$4127,153_{(8)} \rightarrow 857,358_{(16)}$$

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

Rešenje:

$$D5C,13F_{(16)} \rightarrow 1101\ 0101\ 1100,0001\ 0011\ 1111_{(2)}$$

$$D5C,13F_{(16)} \rightarrow 110\ 101\ 011\ 100,000\ 100\ 111\ 111_{(2)} \rightarrow 6534,0477_{(8)}$$

$$D5C,13AF_{(16)} \rightarrow 6534,0477_{(8)}$$

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

Rešenje:

$$1CB,81A_{(16)} \rightarrow 0001\ 1100\ 1011,1000\ 0001\ 1010_{(2)}$$

$$1CB,81A_{(16)} \rightarrow 000\ 111\ 001\ 011,100\ 000\ 011\ 010_{(2)} \rightarrow 713,4032_{(8)}$$

$$1CB,81A_{(16)} \rightarrow 713,4032_{(8)}$$

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:

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

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

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

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

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

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

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

$$h) \quad 0, A4_{(16)} = 10 \cdot 16^{-1} + 4 \cdot 16^{-2} = 0,640625_{(10)}$$

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

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

a) 0000 - BFFF

b) C000 - CFFF

c) D000 - FFFF

Koje će decimalne vrednosti u bajtovima imati veličine ovih segmenata?

Rešenje:

$$a) \quad (0000 - BFFF)_{(16)} = (0 - 49151)_{(10)} \Rightarrow \text{veličina segmenta je } 49151$$

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

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

Primer 21. U decimalnom brojnom sistemu izračunati zbir:

$$22120_{(3)} + 1531_{(6)} + 677_{(9)} + 358_{(14)} + 10B_{(26)} + 9E_{(35)}.$$

Rešenje:

$$\begin{aligned} & 22120_{(3)} + 1531_{(6)} + 677_{(9)} + 358_{(14)} + 10B_{(26)} + 9E_{(35)} = \\ & = 231_{(10)} + 415_{(10)} + 556_{(10)} + 666_{(10)} + 687_{(10)} + 329_{(10)} = 2884_{(10)} \end{aligned}$$

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

Rešenje:

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

Primer 23. Razlomak $4/5$ prevesti u binarni i heksadecimalni zapis.

Rešenje:

$$4/5 = 0,110011001100_{(2)} = 0,CCC_{(16)}$$

Primer 24. Razlomak $1/7$ napisati kao oktalni, binarni i heksadecimalni broj.

Rešenje:

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

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

a) $q = 4$

b) $q = 5$

- c) $q = 7$
- d) $q = 9$.

Rešenje:

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

Primer 26. Mapa operativne memorije nekog računara prikazana je u dekadnoj notaciji: 0 – 191; 192 – 199; 200- 207; 208 -255. Odrediti ekvivalentnu heksadecimalnu notaciju.

Rešenje:

00 – BF; C0 – C7; C8 – CF; D0 – FF.

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) $363,64_{(8)}$
- b) $18E,3C_{(16)}$
- c) $110001100,111001000001_{(2)}$
- d) $101100100101,101000100001_{(2)}$
- e) $1653,206_{(8)}$.

Primer 28. Sledeće decimalne brojeve $x_{(10)}$:

- a) 124,567 d) 246,357 g) 258,401
- b) 456,764 e) 0,286 h) 0,025
- c) 645,780 f) 876 i) 1250,67

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 $x_{(10)}$:

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

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

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

- | | | | |
|------------|------------|-------------|-------------|
| a) A6B,5C4 | g) B,4CDE | m) 9888,65 | t) 1230,ABC |
| b) F3ED3 | h) 2ABC,D | n) 34A34B | u) 0,5DF |
| c) 99,ABCD | i) 0,FEBC | o) ABC,DEF | v) ED34,57 |
| d) 2,ABB | j) 891,435 | p) 3F2E,BAD | x) 5A6B,DF |
| e) CC5 | k) 672,21C | r) 578,226 | y) 1AB,AB3 |
| f) CFA | l) 70448 | s) 9177CF | z) 16276,1. |

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)-invertovanjem svako bita polaznog binarnog broja ($0 \rightarrow 1$ i $1 \rightarrow 0$).
- **Komplement dvojke** (drugi komplement)-dodavanjem jedinice na prvi komplement.

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

Rešenje:

$$\begin{aligned}10^0 - 0,4567_{(10)} &= 1 - 0,4567_{(10)} = \mathbf{0,5433_{(10)}} \\10^2 - 34,639_{(10)} &= 100 - 34,639_{(10)} = \mathbf{65,361_{(10)}}\end{aligned}$$

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

Rešenje:

$$\begin{aligned}10^0 - 10^{-4} - 0,4567_{(10)} &= (1 - 0,0001) - 0,4567_{(10)} \\&= 0,9999_{(10)} - 0,4567_{(10)} \\&= \mathbf{0,5432_{(10)}} \\10^2 - 10^{-3} - 34,639_{(10)} &= (100 - 0,001) - 34,639_{(10)} \\&= 99,999_{(10)} - 34,639_{(10)} \\&= \mathbf{65,360_{(10)}}\end{aligned}$$

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:

$$x_{(10)} = 2^7 b_7 + 2^6 b_6 + \dots + 2^1 b_1 + 2^0 b_0 \qquad 0_{(10)} \leq x_{(10)} \leq 255_{(10)}$$

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

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

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:

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

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

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

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:

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

$$c) 83_{(16)} = 00000000 \underset{2^7}{1} 00000 \underset{2^1 2^0}{11}_{(2)}$$

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 osmобitnom 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):

$$\begin{array}{r} 1111\ 1111\ 1101\ 0110\ (2) \\ +\ 0000\ 0000\ 0000\ 0001\ (2) \\ \hline -41_{(10)} = 1111\ 1111\ 1101\ 0111\ (2) \end{array}$$

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

$$-41_{(10)} = 1111\ 1111\ 1101\ 0111\ (2) = \text{FFD7}_{(16)}$$

! 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 označenog broja, 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 brojnem 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.

Primer 11. 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 što svakako mora biti **označeni** broj, moramo poći od šesnaestobitnog binarnog zapisa broja +3 kao **označenog** broja. Binarni kod broja +3 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 jedan.*

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

! 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:

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

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:

$$-512_{(10)} = 1111\ 1110\ 0000\ 0000_{(2)} = \mathbf{FE00}_{(16)}$$

Primer 14. U šesnaestobitnom zapisu predstavi sledeće decimalne brojeve:

- a) -6
- b) -33
- c) 2049.

Rešenje:

$$\begin{array}{r} \text{a) } 6_{(10)} \rightarrow 0000000000000110_{(2)} \\ \quad 1111111111111001_{(2)} \quad (1. \text{ komplement}) \\ \quad \quad \quad +1 \\ \hline \quad 1111111111111010 \quad (2. \text{ komplement}) \end{array}$$

$$\begin{array}{r} \text{b) } 33_{(10)} \rightarrow 0000000000100001_{(2)} \\ \quad 1111111111011110_{(2)} \quad (1. \text{ komplement}) \\ \quad \quad \quad +1 \\ \hline \quad 1111111111011111 \quad (2. \text{ komplement}) \end{array}$$

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

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:

$$\begin{array}{r} \text{a) } \quad 10100011_{(2)} \\ \quad + 00111010_{(2)} \\ \hline \quad 11011101_{(2)} \quad \text{zbir je } 221_{(10)} \end{array}$$

b) Postupak isti kao pod a). *Uvek je isti.* Zbir je $-35_{(10)}$

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:

$$\begin{array}{r} \text{a) } \quad 01101001_{(2)} \\ \quad + 10001010_{(2)} \\ \hline \quad 11110011_{(2)} \quad \text{zbir je } 243_{(10)} \end{array}$$

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

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.

Primer 17. U osmobitnoj aritmetici binarno sabrati brojeve 129 i 131. Koji decimalni broj je rezultat sabiranja? Komentarisati odgovor. Kakvo će biti stanje zastavica V, N, C posle sabiranja?

Rešenje:

$$\begin{array}{r} 10000001 \\ + 10000011 \\ \hline 00000100 \end{array} \text{ zbir je } 4$$

U ovom slučaju, stanje flegova će biti sledeće: C=1, N=0 (najznačajniji bit rezultata je nula, pa je broj tretiran kao pozitivan), V=1 (fleg V se postavlja pod pretpostavkom da su ulazni podaci označeni brojevi. Prvi broj (10000001), posmatran kao označen je -127, a drugi (10000011) je -125. Sabiranjem ova dva broja dobija se broj van opsega označenog osmobitnog broja (-128 do 127) pa se V fleg postavlja.

Zbog pojave prekoračenja rezultat nije aritmetički tačan.

Primer 18. U osmobitnoj aritmetici binarno sabrati brojeve 126 i – 44. Koji decimalni broj je rezultat sabiranja? Komentarisati odgovor. Kakvo će biti stanje zastavica V, N, C posle sabiranja ?

Rešenje:

$$\begin{array}{r} 01111110 \\ + 11010100 \\ \hline 01010010 \end{array} \text{ zbir je } 82$$

U ovom slučaju, stanje flegova će biti sledeće: **C=1**, **N=0** (najznačajniji bit 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 i kada se tretira kao označen i kao neoznačen, a drugi je -44 kada se tretira kao označen. Sabiranjem ova dva broja dobija se broj u opsegu označenog osmobitnog broja (-128 do 127) pa se V fleg briše.

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

Primer 19. Ilustrovati kako bi se u binarnom sistemu sabrali brojevi 143 i 114:

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:

$$\begin{array}{r} \text{a)} \quad 143 = 10001111_{(2)} \\ + 114 = 01110010_{(2)} \\ \hline + 00000001_{(2)} \end{array} \quad \boxed{\text{8-bitni rezultat}}$$

Rezultat 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 već -113 (binarni kod 10001111₍₂₎ je 143 ako se tretira kao neoznačeni a -113 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 i -113 i dobija se rezultat 1 koji je u opsegu označenih osmobiitnih brojeva (od -128 do +127).

$$\begin{array}{r} \text{b)} \quad 143 = 0000\ 0000\ 1000\ 1111_{(2)} \\ + 114 = 0000\ 0000\ 0111\ 0010_{(2)} \\ \hline 0000\ 0001\ 0000\ 0001_{(2)} \end{array}$$

Rezultat sabiranja u mikroračunaru sa 16-bitnom ALU je **257**.

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, i **C** i **V** i **N** su u ovom slučaju nule. Nema prekoračenja, nema prenosa u sedamnaesti bit i najznačajniji bit je nula pa je broj pozitivan.

Primer 20. Ilustrovati kako bi osmobiitni računar binarno sabrao decimalne brojeve 66 i 254 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:

$$\begin{array}{r} \text{a)} \quad 01000010 = 66 \\ + 11111110 = 254 \\ \hline + 01000000 \end{array}$$

b) **Postupak isti kao pod a).** *Uvek je isti.* Rezultat je **64** (deveti, najznačajniji bit je odsečen) i pod a) i pod b)

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

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

Primer 21. Ilustrovati kako bi osmobarbitni računar binarno sabrao brojeve 34 i 253 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) $00100010 = 34$
 $+ 11111101 = 253$
 ± 00011111

b) **Postupak isti kao pod a).** *Uvek je isti.* Rezultat je **31** (deveti bit je odsečen) i pod a) i pod b)

Rezultat je tačan za označene jer je $11111101_{(2)} = -3$ 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 brojnem sistemu.

Rešenje:

$$\begin{array}{rcl}
 100 : 2 = 50 & | & 0 \\
 50 : 2 = 25 & | & 0 \\
 25 : 2 = 12 & | & 1 \\
 12 : 2 = 6 & | & 0 \\
 6 : 2 = 3 & | & 0 \\
 3 : 2 = 1 & | & 1 \\
 1 : 2 = 0 & | & 1
 \end{array}$$

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

$$\begin{array}{rcl}
 56 : 2 = 28 & | & 0 \\
 28 : 2 = 14 & | & 0 \\
 14 : 2 = 7 & | & 0 \\
 7 : 2 = 3 & | & 1 \\
 3 : 2 = 1 & | & 1 \\
 1 : 2 = 0 & | & 1
 \end{array}$$

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

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

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

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

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:

$$\begin{array}{rcl}
 111 : 2 = 55 & | & 1 \\
 55 : 2 = 27 & | & 1 \\
 27 : 2 = 13 & | & 1 \\
 13 : 2 = 6 & | & 1 \\
 6 : 2 = 3 & | & 0 \\
 3 : 2 = 1 & | & 1 \\
 1 : 2 = 0 & | & 1
 \end{array}$$

$$\begin{array}{rcl}
 87 : 2 = 43 & | & 1 \\
 43 : 2 = 21 & | & 1 \\
 21 : 2 = 10 & | & 1 \\
 10 : 2 = 5 & | & 0 \\
 5 : 2 = 2 & | & 1 \\
 2 : 2 = 1 & | & 0 \\
 1 : 2 = 0 & | & 1
 \end{array}$$

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

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

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

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

$$\begin{array}{r}
 01101111 \\
 + 10101000 \\
 \hline
 \textcircled{1}00010111 \\
 \xrightarrow{+1} \\
 00011000_{(2)} \rightarrow 2^4 + 2^3 = 24_{(10)}.
 \end{array}$$

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:

$$\begin{array}{r|l}
 113 : 2 = 56 & 1 \\
 56 : 2 = 28 & 0 \\
 28 : 2 = 14 & 0 \\
 14 : 2 = 7 & 0 \\
 7 : 2 = 3 & 1 \\
 3 : 2 = 1 & 1 \\
 1 : 2 = 0 & 1
 \end{array}
 \uparrow$$

$$\begin{array}{r|l}
 66 : 2 = 33 & 0 \\
 33 : 2 = 16 & 1 \\
 16 : 2 = 8 & 0 \\
 8 : 2 = 4 & 0 \\
 4 : 2 = 2 & 0 \\
 2 : 2 = 1 & 0 \\
 1 : 2 = 0 & 1
 \end{array}
 \uparrow$$

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

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

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

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

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

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:

$$\begin{array}{r|l}
 124 : 2 = 62 & 0 \\
 62 : 2 = 31 & 0 \\
 31 : 2 = 15 & 1 \\
 15 : 2 = 7 & 1 \\
 7 : 2 = 3 & 1 \\
 3 : 2 = 1 & 1 \\
 1 : 2 = 0 & 1
 \end{array}
 \uparrow$$

$$\begin{array}{r|l}
 91 : 2 = 45 & 1 \\
 45 : 2 = 22 & 1 \\
 22 : 2 = 11 & 0 \\
 11 : 2 = 5 & 1 \\
 5 : 2 = 2 & 1 \\
 2 : 2 = 1 & 0 \\
 1 : 2 = 0 & 1
 \end{array}
 \uparrow$$

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

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

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

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

$$\begin{array}{r}
 01011011 \\
 + 10000011 \\
 \hline
 11011110 \rightarrow 10100001 \text{ (1. komplement)} = -(2^5 + 2^0) = -33_{(10)}.
 \end{array}$$

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:

$$\begin{array}{r|l}
 45 : 2 = 22 & 1 \\
 22 : 2 = 11 & 0 \\
 11 : 2 = 5 & 1 \\
 5 : 2 = 2 & 1 \\
 2 : 2 = 1 & 0 \\
 1 : 2 = 0 & 1
 \end{array}
 \uparrow$$

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

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

$$\begin{array}{r} 73_{(10)} \rightarrow 01001001_{(2)} \\ -73_{(10)} \rightarrow 10110110_{(2)} \quad \text{1.komplement} \\ \quad \quad \quad + 1 \\ \hline \quad \quad \quad 10110111_{(2)} \quad \text{2.komplement} \end{array}$$

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

$$\begin{array}{r} 00101101_{(2)} \\ + 10110111_{(2)} \\ \hline 11100100_{(2)} \rightarrow -1*2^7 + 1*2^6 + 1*2^5 + 1*2^2 = -28_{(10)} \end{array}$$

ili

$$\begin{array}{r} 11100100 \\ 10011011 \quad (\text{negativan br. 1.komplement}) \\ \quad + 1 \quad (\text{korekcija}) \\ \hline \end{array}$$

$$10011100_{(2)} \rightarrow -(1*2^4 + 1*2^3 + 1*2^2) = -28_{(10)}.$$

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:

$$\begin{array}{r|l} 59 : 2 = 29 & 1 \uparrow \\ 29 : 2 = 14 & 1 \\ 14 : 2 = 7 & 0 \\ 7 : 2 = 3 & 1 \\ 3 : 2 = 1 & 1 \\ 1 : 2 = 0 & 1 \end{array}$$

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

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

$$\begin{array}{r} 100_{(10)} \rightarrow 01100100_{(2)} \\ -100_{(10)} \rightarrow 10011011_{(2)} \quad \text{1. komplement} \\ \quad \quad \quad + 1 \\ \hline \quad \quad \quad 10011100_{(2)} \quad \text{2. komplement} \end{array}$$

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

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

ili

$$\begin{array}{r} 11010111 \\ 10101000 \quad (\text{negativan br. 1.komplement}) \\ \quad + 1 \quad (\text{korekcija}) \\ \hline \end{array}$$

$$10101001_{(2)} \rightarrow -(1*2^5 + 1*2^3 + 1*2^0) = -41_{(10)}.$$

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	$\uparrow$
$61 : 2 = 30$	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	$\uparrow$
$47 : 2 = 23$	1	
$23 : 2 = 11$	1	
$11 : 2 = 5$	1	
$5 : 2 = 2$	1	
$2 : 2 = 1$	0	
$1 : 2 = 0$	1	

$$\begin{array}{rcl}
 122_{(10)} & \rightarrow & 01111010_{(2)} \\
 - 122_{(10)} & \rightarrow & 10000101_{(2)} \quad \text{1. komplement} \\
 & & \underline{+ 1} \\
 & & 10000110_{(2)} \quad \text{2. komplement}
 \end{array}$$

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

$$\begin{array}{rcl}
 & & 01011110_{(2)} \\
 + & & \underline{10000110_{(2)}} \\
 11100100_{(2)} & \rightarrow & 10011011_{(2)} \quad \text{(1. komplement)} \\
 & & \underline{+ 1} \\
 & & 10011100_{(2)} \quad \text{(2. komplement)} = - (2^4 + 2^3 + 2^2) = -28_{(10)}.
 \end{array}$$

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	$\uparrow$
$53 : 2 = 26$	1	
$26 : 2 = 13$	0	
$13 : 2 = 6$	1	
$6 : 2 = 3$	0	
$3 : 2 = 1$	1	
$1 : 2 = 0$	1	

$63 : 2 = 31$	1	$\uparrow$
$31 : 2 = 15$	1	
$15 : 2 = 7$	1	
$7 : 2 = 3$	1	
$3 : 2 = 1$	1	
$1 : 2 = 0$	1	

$$\begin{array}{rcl}
 107_{(10)} & \rightarrow & 01101011_{(2)} \\
 - 63_{(10)} & \rightarrow & 11000000_{(2)} \quad \text{1. komplement} \\
 & & \underline{+ 1} \\
 & & 11000001_{(2)} \quad \text{2. komplement}
 \end{array}$$

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

$$\begin{array}{rcl}
 & & 01101011_{(2)} \\
 + & & \underline{11000001_{(2)}} \\
 1001101100_{(2)} & = & 2^5 + 2^3 + 2^2 = 44_{(10)}
 \end{array}$$

Primer 30. Koristeći 8-bitne označene binarne brojeve izračunati primenom:

a) prvog komplementa

b) drugog komplementa

Razliku brojeva i rezultat prikazati 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) $31_{(10)} - 112_{(10)}$

ñ) $14_{(10)} - 105_{(10)}$

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 od decimalne tačke definiše opseg brojeva koji mogu da se predstavljaju tim formatom.
- Broj cifara desno od decimalne tačke 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

Simbol za format **neoznačenih brojeva** sa nepokretnom decimalnom tačkom, ako je binarni zapis dužine N je:

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

$N - R$ = broj cifara ispred decimalne tačke

R = broj cifara iza decimalne tačke (radix)

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

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

Simbol za format **označenih brojeva** sa nepokretnom decimalnom tačkom, ako je binarni zapis dužine N je:

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

$N - R$ = broj cifara ispred decimalne tačke, uključujući 1 bit za znak (MSB)

R = broj cifara iza decimalne tačke (radix)

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

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

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 $Q_{N-R,R}$ pri čemu je $N = 8$, a $R \in \{2, 3, 4, 5\}$
- c) u formatu $Q^S_{N-R,R}$ 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 $x = b_R b_{R-1} \dots b_1 b_0$, gde su $b_R, b_{R-1}, \dots, b_1, b_0 \in \{0, 1\}$ cifre u binarnom zapisu. Za konverziju u decimalni brojni sistem $x_{(2)} \rightarrow x_{(10)}$, koristi se opšta formula

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

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

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

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

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

Pošto je $R = 2$,

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

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:

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

Pošto je $R = 3$,

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

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 $Q^S_{N-R,R}$ 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:

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

Pošto je $R = 2$,

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

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:

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

Pošto je $R = 3$,

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

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:

$$x_{(2)} = \underbrace{11001}_{5 \text{ cifara}}, \underbrace{001}_{3 \text{ cifre}} = 2^4 + 2^3 + 2^0 + 2^{-3} = \mathbf{25.125}_{(10)}$$

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 koristi 6 binarnih cifara, a za decimalni deo 2 binarne cifre:

$$x_{(2)} = \underbrace{011010}_{6 \text{ cifara}}, \underbrace{11}_{2 \text{ cifre}} = 2^4 + 2^3 + 2^1 + 2^{-1} + 2^{-2} = \mathbf{26.75}_{(10)}$$

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:

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

Primer 5. 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) $97_{(16)} \rightarrow 1001\ 0111_{(2)}$; ako je format $Q_{7,1} \Rightarrow \mathbf{1001011,1}_{(2)} \rightarrow \mathbf{75,5}_{(10)}$.
b) $97_{(16)} \rightarrow 1001\ 0111_{(2)}$; ako je format $Q_{7,1} \Rightarrow \mathbf{1001011,1}_{(2)} \rightarrow \mathbf{-52,5}_{(10)}$.

Primer 6. 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) $0000\ 1011_{(2)} \rightarrow 0,34375_{(10)}$

Pri određivanju **optimalnog formata** treba obezbediti:

- **Dovoljan broj bita** levo od decimalne tačke za **celobrojni deo**
- **Maksimalan broj bita** desno decimalne tačke za **razlomljeni deo**

Promenom formata menjaju se tačnost prikaza brojne vrednosti i opseg

Primer 7. Odrediti optimalan format $Q_{8-R, R}$ za čuvanje decimalnog broja 4,563.

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:

$$\begin{array}{rcl}
 4_{(10)} = 100_{(2)} & 0.563 * 2 = 1.126 & \left| \begin{array}{c} 1 \\ 0 \\ 0 \\ 1 \\ 0 \end{array} \right. \\
 & 0.126 * 2 = 1.252 & \\
 & 0.252 * 2 = 0.504 & \\
 & 0.504 * 2 = 1.008 & \\
 & 0.008 * 2 = 0.016 & \downarrow
 \end{array}$$

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

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

Primer 8. 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:

$$\begin{array}{rcl}
 2_{(10)} = 10_{(2)} & 0.872 * 2 = 1.744 & \left| \begin{array}{c} 1 \\ 1 \\ 0 \\ 1 \\ 1 \\ 1 \end{array} \right. \\
 & 0.744 * 2 = 1.488 & \\
 & 0.488 * 2 = 0.976 & \\
 & 0.976 * 2 = 1.952 & \\
 & 0.952 * 2 = 1.904 & \\
 & 0.904 * 2 = 1.808 & \downarrow
 \end{array}$$

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

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

Primer 9. 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 $\mathbf{Q_{3,5}^S}$. 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:

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

Ekvivalentna decimalna vrednost označenog broja u optimalnom formatu je:

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

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

Primer 10. 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.

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

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:

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

Primer 11. Kako broj π ($3,1416_{(10)}$) predstaviti u binarnom brojnom sistemu u osmobitnom zapisu, ako je:

a) broj neoznačen,

b) broj označen,

i koji je optimalni format?

Rešenje:

- a) $3,1416_{(10)}$ (neoznačen) $\rightarrow 11001001_{(2)}$ u formatu $Q_{2,6}$
 - b) $3,1416_{(10)}$ (označen) $\rightarrow 01100100_{(2)}$ u formatu $Q^s_{3,5}$.
-

Primer 12. Kako bi u memoriji osmobiitnog računara izgledao optimalan zapis decimalnog broja $66,6_{(10)}$?
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) $\rightarrow 10000101_{(2)}$ u formatu $Q_{7,1}$
 - b) $66,6_{(10)}$ (označen) $\rightarrow 01000010_{(2)}$ u formatu $Q^s_{8,0}$
-

Primer 13. Koji heksadecimalan sadržaj predstavlja optimalni zapis decimalnog broja $9,87_{(10)}$ u osmobiitnom sistemu:
a) neoznačenih brojeva
b) označenih brojeva
Naznačiti format u kome je broj zapisan.

Rešenje:

- a) $9,87_{(10)}$ (neoznačeni) $\rightarrow 10011101_{(2)} = 9D_{16}$ u formatu $Q_{4,4}$
 - b) $9,87_{(10)}$ (označeni) $\rightarrow 01001110_{(2)} = 4E_{16}$ u formatu $Q_{5,3}$
-

Primer 14. Kako bi u memoriji osmobiitnog 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) $\rightarrow 10001010_{(2)}$ u formatu $Q_{6,2}$
 - b) $34,6_{(10)}$ (označeni) $\rightarrow 01000101_{(2)}$ u formatu $Q^s_{7,1}$
-

Primer 15. Koji heksadecimalan sadržaj predstavlja optimalni zapis decimalnog broja $6,283_{(10)}$ u osmobiitnom sistemu:
a) neoznačenih brojeva
b) označenih brojeva
Naznačiti format u kome je broj zapisan.

Rešenje:

- a) $6,283_{(10)}$ (neoznačeni) $\rightarrow 11001001_{(2)} = \mathbf{C9}_{(16)}$ u formatu $\mathbf{Q}_{3,5}$
b) $6,283_{(10)}$ (označeni) $\rightarrow 01100100_{(2)} = \mathbf{64}_{(16)}$ u formatu $\mathbf{Q}_{4,4}^s$
-

Primer 16. U šesnaestobitnom zapisu sa fiksnim zarezom, koristeći format koji je optimalan sa gledišta tačnosti zapisa, predstaviti decimalni broj 1025,608 kao (a) neoznačen i kao (b) označen broj. Odgovore dati kao dva šesnaestobitna (četvorocifrena) heksadecimalna broja.

Rešenje:

Pretvaranjem u binarni sistem celobrojnog dela dobijamo

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

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

$$0,608 = 0,100110111010\dots_{(2)} \quad (\text{zapis beskonačne dužine})$$

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

Dakle,

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

- 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:

$$1025,608_{(10)} = \dots 000001\mathbf{0000000001},\mathbf{100110111010}\dots_{(2)}$$

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):

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

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 toliko 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 vrednosti 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. Predstavljen sa 12 bita,

$$1025_{(10)} = 0100\ 0000\ 0001_{(2)} \quad (\text{zapis dužine 12 bita})$$

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

$$0,608_{(10)} = 0,\underbrace{1001}_{\text{Četiri bita}}10111010\dots$$

Četiri bita mogu da uđu u zapis (12 je neophodno za zapis)

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

$$1025,608_{(10)} = \dots 0000010000000001,100110111010\dots_{(2)}$$

Uporedite ovo rešenje sa rešenjem pod a) i podsetite se zašto je nužna nula na mestu najznačajnijeg bita kada se pozitivan broj prikazuje kao označen.

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

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

Primer 17. 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 $+71,75 = 1000111,11_{(2)}$

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 $Q_{13,3}^S$ kao i $Q_{12,4}^S$ i tako dalje sve do formata $Q_{8,8}^S$.

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 $Q_{14,2}^S$

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

Negativan broj se dobija kao drugi komplement pozitivnog broja iste apsolutne vrednosti. Drugi komplement se dobija pojedinačnim invertovanjem svakog bita (što je prvi komplement) i dodavanjem jedinice na bit najmanje pozicione vrednosti (bez obzira gde je položaj zareza). Zato je :

$$-71,75_{(10)} = 1111\ 1110\ 1110\ 0001_{(2)} = \mathbf{FEE1}_{(16)} (Q_{14,2}^S)$$

Podjednako dobro rešenje je i, na primer:

$$-71,75_{(10)} = 1011\ 1000\ 0100\ 0000_{(2)} = \mathbf{B840}_{(16)} (Q_{8,8}^S)$$

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 $-71_{(10)} = 1111\ 1111\ 1011\ 1001_{(2)}$ (kao drugi komplement zapisa za $+71_{(10)}$)

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

Kombinacijom ova dva dolazi se do pogrešnog rezultata:

$$-71,75_{(10)} = 11\ 1111\ 1011\ 1001,11_{(2)}$$

Rezultat **očigledno nije tačan**. Uporedite rezultat i postupak sa rešenjem. Najčistije je razlomljeni broj tretirati kao celinu. Ako se tako ne tretira, moramo paziti da delovi sa kojima radimo nezavisno zaista daju polazni broj. Tako, broj **-71,75 nije zbir brojeva -71 i 0,75**. Ma kako očigledno delovala ova prosta aritmetička činjenica, zbog iznenađujuće velikog procenta pojave ove greške, na nju treba obratiti posebnu pažnju. Greška je naročito

Primer 18. 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 $Q_{1,15}^S$?

Rešenje:

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

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

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

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

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

$01000000 = 64,87_{(10)} (Q^s_{8,0})$

$+ 11111101 = -3,14_{(10)} (Q^s_{8,0})$

$\pm 00111101 = 61_{(10)} (Q^s_{8,0})$

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

Primer 23. Opisati binarno sabiranje brojeva $32,17_{(10)}$ i $-9,81_{(10)}$ u osmobitnoj aritmetici fiksnog zareza. Koji decimalni broj je rezultat ovakvog sabiranja?

Rešenje:

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

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

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

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

$01000000 = 32,17_{(10)} (Q^s_{7,1})$

$+ 11101101 = -9,81_{(10)} (Q^s_{7,1})$

$\pm 00101101 = 22,5_{(10)} (Q^s_{7,1})$

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

**** 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 **Gray**-ov kod.

- 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.

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

Gray-ov 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	Gray-ov 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

Primer 24. U memoriji šesnaestobitnog računara nalazi se podatak 8C38₍₁₆₎. O kom decimalnom broju je reč ako se zna da je zapisan kao:

- Neoznačen ceo broj
- Označen ceo broj
- Četvorocifreni broj u kodu “više 3” (viši bajt je prvi)
- Neoznačeni broj u formatu Q_{12,4}
- Dva označena osmobiitna broja u formatu Q_{4,4}
- Dva neoznačena osmobiitna broja u formatu Q_{2,6}

Rešenje:

- $2^{15} + 2^{11} + 2^{10} + 2^5 + 2^4 + 2^3 = \mathbf{35896}_{(10)}$
- $-2^{15} + 2^{11} + 2^{10} + 2^5 + 2^4 + 2^3 = \mathbf{-29640}_{(10)}$
- $\mathbf{5905}_{(10)}$

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 2⁰ do 2¹¹, a desno od zareza rastu u negativnu stranu od 2⁻¹ do 2⁻⁴.

$$8C38_{(16)} (Q_{12,4}) = \underset{2^{11}}{1} \underset{2^7}{0} \underset{2^6}{0} \underset{2^5}{0} \underset{2^4}{1} \underset{2^3}{1} \underset{2^2}{0} \underset{2^1}{0} \underset{2^0}{0} \underset{2^{-1}}{1} \underset{2^{-2}}{1} \underset{2^{-3}}{0} \underset{2^{-4}}{0} \underset{2^{-5}}{0} \underset{2^{-6}}{0} \underset{(2)}{0}$$

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:

$8C_{38(16)}$ se direktno pretvara u decimalni kôd, a dobijeni rezultat treba da se podeli sa 2^4 (odnosno da se pomnoži sa $1/2^4$):

$$8C_{38(16)} = 2^{-4} (2^{15} + 2^{11} + 2^{10} + 2^5 + 2^4 + 2^3) = 35896/16 = \mathbf{2243,5_{(10)}}$$

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:*

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

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

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 25. 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 osmобitna broja u formatu $Q_{4,4}$
- f) Dva neoznačena osmобitna 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 26. U dve uzastopne lokacije osmобitnog 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) $105_{(10)}$ i $138_{(10)}$
- b) $105_{(10)}$ i $-118_{(10)}$
- c) $69_{(10)}$ i $57_{(10)}$
- d) $3,28125_{(10)}$ i $4,3125_{(10)}$
- e) $52,5_{(10)}$ i $69,0_{(10)}$

Primer 27. 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) $163_{(10)}$ i $58_{(10)}$
- b) $-93_{(10)}$ i $58_{(10)}$
- c) $7007_{(10)}$
- d) $10,1875_{(10)}$ i $3,625_{(10)}$
- e) $81,5_{(10)}$ i $29,0_{(10)}$

Primer 28. 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 $Q_{8,0}^S$; $Q_{5,3}$; $Q_{7,1}$; $Q_{4,4}$; $Q_{6,2}^S$; $Q_{3,5}$; $Q_{2,6}$ i $Q_{1,7}^S$.

Primer 29. 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 $Q_{N-R,R}^S$ pri čemu je $N = 8$, a $R \in \{2, 3, 4, 5\}$.

- | | | |
|-------------|-------------|---------------|
| 1) 11100011 | 5) 00011011 | 9) 00111110 |
| 2) 10001010 | 6) 11101111 | 10) 01010101 |
| 3) 01000111 | 7) 11001100 | 11) 01011100 |
| 4) 11110110 | 8) 01001101 | 12) 00110100. |

4. SABIRANJE PRIMENOM KODOVA “8421” I “VIŠE 3”

Sabiranje brojeva koji su predstavljeni kodovima “8421” i “više 3” ne može da se izvrši direktnom primenom pravila binarne aritmetike

- Neophodna je hardverska ili softverska korekcija.
- Sporija je obrada od binarne aritmetike.
- Programer vodi računa o pravilnoj interpretaciji kodiranih brojeva.

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 $< 1010_{(2)}$ nema korekcije.
- Ako je broj $\geq 1010_{(2)}$ i nema prenosa u sledeću tetradu, dodaje se $6_{(10)}(0110_{(2)})$
- Ako postoji prenos u sledeću tetradu, 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 se pojavi prenos u sledeću tetradu, 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 $-3_{(10)}$, odnosno $1101_{(2)}$).

Primer 1. Prikazati dekadni broj $15,147_{(10)}$ u kodu “8421” i kodu “više 3”.

Rešenje:

$$\begin{aligned} 15,147_{(10)} &= \mathbf{0001\ 0101\ ,\ 0001\ 0100\ 0111}_{(8421)} \\ 15,147_{(10)} &= \mathbf{0100\ 1000\ ,\ 0100\ 0111\ 1010}_{(“\text{više } 3”)} \end{aligned}$$

Primer 2. Dati su brojevi:

a) $100100,00111$ u kodu “8421”

b) $111011,01111$ u kodu “više 3”.

Napisati ove brojeve u decimalnom brojnom sistemu.

Rešenje:

$$\begin{aligned} \text{a)} \quad 100100\ ,\ 00111_{(8421)} &= 0010\ 0100\ ,\ 0011\ 1000 = \mathbf{24,38}_{(10)} \\ \text{b)} \quad 111011\ ,\ 01111_{(“\text{više } 3”)} &= 0011\ 1011\ ,\ 0111\ 1000 = \mathbf{8,45}_{(10)}. \end{aligned}$$

Primer 3. Prikazati u kodu “8421” brojeve $6187_{(10)}$ i $2495_{(10)}$, a zatim izračunati njihov zbir i vratiti rezultat u decimalni brojni sistem.

Rešenje:

$$\begin{array}{r}
 6187_{(10)} \quad 0110\ 0001\ 1000\ 0111_{(8421)} \\
 2495_{(10)} \quad + \underline{0010\ 0100\ 1001\ 0101_{(8421)}} \\
 \quad 1000\ 0110\ 0001\ 1100 \\
 \quad + \quad \underline{0110\ 0110} \quad 1. \text{ korekcija} \\
 \quad \quad \quad \mathbf{1000\ 0110\ 1000\ 0010 = 8682_{(10)}}
 \end{array}$$

Primer 4. Prikazati u kodu “8421” brojeve $7531_{(10)}$ i $1484_{(10)}$, a zatim izračunati njihov zbir i vratiti rezultat u decimalni brojni sistem.

Rešenje:

$$\begin{array}{r}
 7531_{(10)} \quad 0111\ 0101\ 0011\ 0001_{(8421)} \\
 1484_{(10)} \quad + \underline{0001\ 0100\ 1000\ 0100_{(8421)}} \\
 \quad 1000\ 1001\ 1011\ 0101 \\
 \quad + \quad \underline{0110} \quad 1. \text{ korekcija} \\
 \quad 1000\ 1010\ 0001\ 0101 \\
 \quad + \quad \underline{0110} \quad 2. \text{ korekcija} \\
 \quad \quad \quad \mathbf{1001\ 0000\ 0001\ 0101_{(8421)} = 9015_{(10)}}
 \end{array}$$

Primer 5. Prikazati u kodu “8421” brojeve $6187_{(10)}$ i $2495_{(10)}$, a zatim izračunati njihov zbir i vratiti rezultat u decimalni brojni sistem.

Rešenje:

$$\begin{array}{r}
 6187_{(10)} \quad 0110\ 0001\ 1000\ 0111_{(8421)} \\
 2495_{(10)} \quad + \underline{0010\ 0100\ 1001\ 0101_{(8421)}} \\
 \quad 1000\ 0110\ 0001\ 1100 \\
 \quad + \quad \underline{0110\ 0110} \quad 1. \text{ korekcija} \\
 \quad \quad \quad \mathbf{1000\ 0110\ 1000\ 0010 = 8682_{(10)}}
 \end{array}$$

Primer 6. Prikazati u kodu “8421” brojeve $7531_{(10)}$ i $1484_{(10)}$, a zatim izračunati njihov zbir i vratiti rezultat u decimalni brojni sistem.

Rešenje:

$$\begin{array}{r}
 7531_{(10)} \quad 0111\ 0101\ 0011\ 0001_{(8421)} \\
 1484_{(10)} \quad + \underline{0001\ 0100\ 1000\ 0100_{(8421)}} \\
 \quad 1000\ 1001\ 1011\ 0101 \\
 \quad + \quad \underline{0110} \quad 1. \text{ korekcija} \\
 \quad 1000\ 1010\ 0001\ 0101 \\
 \quad + \quad \underline{0110} \quad 2. \text{ korekcija} \\
 \quad \quad \quad \mathbf{1001\ 0000\ 0001\ 0101_{(8421)} = 9015_{(10)}}
 \end{array}$$

Primer 7. Prikazati u kodu “8421” brojeve $5324_{(10)}$ i $1768_{(10)}$, a zatim izračunati njihov zbir i vratiti rezultat u decimalni brojni sistem.

Rešenje:

$$\begin{array}{r}
 5324_{(10)} \quad 0101\ 0011\ 0010\ 0100_{(8421)} \\
 1768_{(10)} \quad + \underline{0001\ 0111\ 0110\ 1000}_{(8421)} \\
 \quad \quad 0110\ 1010\ 1000\ 1100 \\
 \quad \quad + \underline{\quad 0110 \quad \quad 0110 \quad} \quad \text{korekcija} \\
 \quad \quad \quad \mathbf{0111\ 0000\ 1001\ 0010}_{(8421)} = \mathbf{7092}_{(10)}
 \end{array}$$

Primer 8. Prikazati u kodu “8421” brojeve $3712_{(10)}$ i $1456_{(10)}$, a zatim izračunati njihov zbir i vratiti rezultat u decimalni brojni sistem.

Rešenje:

$$\begin{array}{r}
 3712_{(10)} \quad 0011\ 0111\ 0001\ 0010_{(8421)} \\
 1456_{(10)} \quad + \underline{0001\ 0100\ 0101\ 0110}_{(8421)} \\
 \quad \quad 0100\ 1011\ 0110\ 1000 \\
 \quad \quad + \underline{\quad 0110 \quad} \quad \text{korekcija} \\
 \quad \quad \quad \mathbf{0101\ 0001\ 0110\ 1000}_{(8421)} = \mathbf{5168}_{(10)}.
 \end{array}$$

Primer 9. Prikazati u kodu “8421” brojeve $1357_{(10)}$ i $5468_{(10)}$, a zatim izračunati njihov zbir i vratiti rezultat u decimalni brojni sistem.

Rešenje:

$$\begin{array}{r}
 1357_{(10)} \quad 0001\ 0011\ 0101\ 0111_{(8421)} \\
 5468_{(10)} \quad + \underline{0101\ 0100\ 0110\ 1000}_{(8421)} \\
 \quad \quad 0110\ 0111\ 1011\ 1111 \\
 \quad \quad + \underline{\quad 0110\ 0110 \quad} \quad \text{korekcija} \\
 \quad \quad \quad \mathbf{0110\ 1000\ 0010\ 0101}_{(8421)} = \mathbf{6825}_{(10)}.
 \end{array}$$

Primer 10. Prikazati u kodu “8421” brojeve $2875_{(10)}$ i $6943_{(10)}$, a zatim izračunati njihov zbir i vratiti rezultat u decimalni brojni sistem.

Rešenje:

$$\begin{array}{r}
 2875_{(10)} \quad 0010\ 1000\ 0111\ 0101_{(8421)} \\
 6943_{(10)} \quad + \underline{0110\ 1001\ 0100\ 0011}_{(8421)} \\
 \quad \quad 1001\ 0001\ 1011\ 1000 \\
 \quad \quad + \underline{\quad 0110\ 0110 \quad} \quad \text{korekcija} \\
 \quad \quad \quad \mathbf{1001\ 1000\ 0001\ 1000}_{(8421)} = \mathbf{9818}_{(10)}.
 \end{array}$$

Primer 11. Prikazati u kodu “8421” brojeve $1337_{(10)}$ i $2468_{(10)}$, a zatim izračunati njihov zbir i vratiti rezultat u decimalni brojni sistem.

Rešenje:

$$\begin{array}{r}
 1337_{(10)} \quad 0001\ 0011\ 0011\ 0111_{(8421)} \\
 2468_{(10)} \quad + \underline{0010\ 0100\ 0110\ 1000}_{(8421)} \\
 \quad 0011\ 0111\ 1001\ 1111 \\
 \quad + \underline{\quad\quad\quad 0110\quad\quad\quad} \quad \text{korekcija} \\
 \quad 0011\ 0111\ 1010\ 0101 \\
 \quad + \underline{\quad\quad\quad 0110\quad\quad\quad} \quad \text{korekcija} \\
 \quad \mathbf{0011\ 1000\ 0000\ 0101}_{(8421)} = \mathbf{3805}_{(10)}.
 \end{array}$$

Primer 12. Prikazati u kodu “8421” brojeve $2075_{(10)}$ i $5943_{(10)}$, a zatim izračunati njihov zbir i vratiti rezultat u decimalni brojni sistem.

Rešenje:

$$\begin{array}{r}
 2075_{(10)} \quad 0010\ 0000\ 0111\ 0101_{(8421)} \\
 5943_{(10)} \quad + \underline{0101\ 1001\ 0100\ 0011}_{(8421)} \\
 \quad 0111\ 1001\ 1011\ 1000 \\
 \quad + \underline{\quad\quad\quad 0110\quad\quad\quad} \quad \text{korekcija} \\
 \quad 0111\ 1010\ 0001\ 1000 \\
 \quad + \underline{\quad\quad\quad 0110\quad\quad\quad} \quad \text{korekcija} \\
 \quad \mathbf{1000\ 0000\ 0001\ 1000}_{(8421)} = \mathbf{8018}_{(10)}.
 \end{array}$$

Primer 13. Prikazati u kodu “8421” brojeve $2369_{(10)}$ i $1653_{(10)}$, a zatim izračunati njihov zbir i vratiti rezultat u decimalni brojni sistem.

Rešenje:

$$\begin{array}{r}
 2369_{(10)} \quad 0010\ 0011\ 0110\ 1001_{(8421)} \\
 1653_{(10)} \quad + \underline{0001\ 0110\ 0101\ 0011}_{(8421)} \\
 \quad 0011\ 1001\ 1011\ 1100 \\
 \quad + \underline{\quad\quad\quad 0110\ 0110\quad\quad\quad} \quad \text{korekcija} \\
 \quad 0011\ 1010\ 0010\ 0010 \\
 \quad + \underline{\quad\quad\quad 0110\quad\quad\quad} \quad \text{korekcija} \\
 \quad \mathbf{0100\ 0000\ 0010\ 0010}_{(8421)} = \mathbf{4012}_{(10)}.
 \end{array}$$

Primer 14. Prikazati u kodu “više 3” brojeve $8153_{(10)}$ i $1298_{(10)}$, a zatim izračunati njihov zbir i vratiti rezultat u decimalni brojni sistem.

Rešenje:

$$\begin{array}{r}
 8153_{(10)} = 1011\ 0100\ 1000\ 0110_{(“\text{više } 3”)} \\
 1298_{(10)} = + \underline{0100\ 0101\ 1100\ 1011}_{(“\text{više } 3”)} \\
 \quad 1111\ 1010\ 0101\ 0001 \\
 \quad + \underline{1101\ 1101\ 0011\ 0011} \quad \text{korekcija} \\
 \quad \mathbf{1100\ 0111\ 1000\ 0100}_{(“\text{više } 3”)} = \mathbf{9451}_{(10)}
 \end{array}$$

Primer 15. Prikazati u kodu “više 3” brojeve $5947_{(10)}$ i $3106_{(10)}$, a zatim izračunati njihov zbir i vratiti rezultat u decimalni brojni sistem.

Rešenje:

$$\begin{array}{r}
 5947_{(10)} \quad 1000 \ 1100 \ 0111 \ 1010 \text{ ("više 3") } \\
 3106_{(10)} \quad + \underline{0110 \ 0100 \ 0011 \ 1001} \text{ ("više 3") } \\
 \quad \quad 1111 \ 0000 \ 1011 \ 0011 \\
 \quad \quad + \underline{1101 \ 0011 \ 1011 \ 0011} \text{ korekcija} \\
 \quad \quad \mathbf{1100 \ 0011 \ 1000 \ 0110} \text{ ("više 3") } = \mathbf{9053_{(10)}}
 \end{array}$$

Primer 16. Prikazati u kodu "više 3" brojeve $2875_{(10)}$ i $6943_{(10)}$, a zatim izračunati njihov zbir i vratiti rezultat u decimalni brojni sistem.

Rešenje:

$$\begin{array}{r}
 2875_{(10)} \quad 0101 \ 1011 \ 10101000 \text{ ("više 3") } \\
 6943_{(10)} \quad + \underline{1001 \ 1100 \ 0111 \ 0110} \text{ ("više 3") } \\
 \quad \quad 1111 \ 1000 \ 0001 \ 1110 \\
 \quad \quad + \underline{1101 \ 0011 \ 0011 \ 1101} \text{ korekcija} \\
 \quad \quad \mathbf{1100 \ 1011 \ 0100 \ 1011} \text{ ("više 3") } = \mathbf{9818_{(10)}}
 \end{array}$$

Primer 17. Prikazati u kodu "više 3" brojeve $5324_{(10)}$ i $1768_{(10)}$, a zatim izračunati njihov zbir i vratiti rezultat u decimalni brojni sistem.

Rešenje:

$$\begin{array}{r}
 5324_{(10)} \quad 1000 \ 0110 \ 0101 \ 0111 \text{ ("više 3") } \\
 1768_{(10)} \quad + \underline{0100 \ 1010 \ 1001 \ 1011} \text{ ("više 3") } \\
 \quad \quad 1101 \ 0000 \ 1111 \ 0010 \\
 \quad \quad + \underline{1101 \ 0011 \ 1101 \ 0011} \text{ korekcija} \\
 \quad \quad \mathbf{1010 \ 0011 \ 1100 \ 0101} \text{ ("više 3") } = \mathbf{7092_{(10)}}
 \end{array}$$

Primer 18. Prikazati u kodu "više 3" brojeve $3712_{(10)}$ i $1456_{(10)}$, a zatim izračunati njihov zbir i vratiti rezultat u decimalni brojni sistem.

Rešenje:

$$\begin{array}{r}
 3712_{(10)} \quad 0110 \ 1010 \ 0100 \ 0101 \text{ ("više 3") } \\
 1456_{(10)} \quad + \underline{0100 \ 0111 \ 1000 \ 1001} \text{ ("više 3") } \\
 \quad \quad 1011 \ 0001 \ 1100 \ 1110 \\
 \quad \quad + \underline{1101 \ 0011 \ 1101 \ 1101} \text{ korekcija} \\
 \quad \quad \mathbf{1000 \ 0100 \ 1001 \ 1011} \text{ ("više 3") } = \mathbf{5168_{(10)}}
 \end{array}$$

Primer 19. Prikazati u kodu "više 3" brojeve $5318_{(10)}$ i $2471_{(10)}$, a zatim izračunati njihov zbir i vratiti rezultat u decimalni brojni sistem.

Rešenje:

$$\begin{array}{r}
 5318_{(10)} \quad 1000 \ 0110 \ 0100 \ 1011 \text{ ("više 3") } \\
 2471_{(10)} \quad + \underline{0101 \ 0111 \ 1010 \ 0100} \text{ ("više 3") } \\
 \quad \quad 1101 \ 1101 \ 1110 \ 1111 \\
 \quad \quad + \underline{1101 \ 1101 \ 1101 \ 1101} \text{ korekcija} \\
 \quad \quad \mathbf{1010 \ 1010 \ 1011 \ 1100} \text{ ("više 3") } = \mathbf{7789_{(10)}}
 \end{array}$$

Primer 20. Prikazati navedene brojeve u kodu:

a) **8421**

b) **“više 3”**

a zatim izračunati njihov zbir i vratiti rezultat u decimalni 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 21. Prikazati navedene brojeve u kodu:

a) **8421**

b) **“više 3”**

a zatim izračunati njihov zbir i vratiti rezultat u decimalni 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 $S = \{x, y, z, \dots\}$ koji sadrži najmanje dva različita elementa, i neka su na ovom skupu definisana dva binarna operanda: $+$ (logičko sabiranje, ILI) i $\cdot$ (logičko množenje, I), i jedan unarni operand $\bar{}$ (negacija, NE).

Bulova algebra sadrži dva specijalna elementa **0** i **1**, takva da sve promenljive $x, y, z, \dots$ uzimaju vrednost iz skupa $\{0, 1\}$. Da bi ovaj skup S , i operacije $+$, $\cdot$ i $\bar{}$ sačinjavali Bulovu algebru, neophodno je da budu zadovoljene **aksiome Hantingtona**:

A-1: Binarne operacije $+$ i $\cdot$ su komutativne na skupu S , i međusobno su distributivne tako da za svako x, y, z , koji pripadaju skupu S , važi:

$$x + y = y + x$$

$$x \cdot y = y \cdot x$$

A-2: Binarne operacije $+$ i $\cdot$ 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:

$$x + 0 = 0 + x = x$$

$$x \cdot 1 = 1 \cdot x = x.$$

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 :

$$x + \bar{x} = 1$$

$$x \cdot \bar{x} = 0.$$

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: $A + \bar{A} \cdot B = A + B$

Rešenje:

$$\begin{aligned}A + \bar{A} \cdot B &= A \cdot (1 + \bar{A} + B) + \bar{A} \cdot B = A + A \cdot \bar{A} + A \cdot B + \bar{A} \cdot B = \\ &= A \cdot A + A \cdot \bar{A} + A \cdot B + \bar{A} \cdot B = \\ &= A \cdot (A + B) + \bar{A} \cdot (A + B) = \\ &= (A + B) \cdot (A + \bar{A}) = A + B\end{aligned}$$

Primer 4. Dokazati identitet: $A + B + \bar{A} \cdot \bar{B} = 1$

Rešenje:

$$A + B + \bar{A} \cdot \bar{B} = (A + B) + (\overline{A + B}) = 1$$

Primer 5. Dokazati identitet: $(A + B) \cdot (A + \bar{B}) = A$

Rešenje:

$$\begin{aligned}(A + B) \cdot (A + \bar{B}) &= A \cdot A + A \cdot \bar{B} + A \cdot B + B \cdot \bar{B} \\ &= A + A \cdot (B + \bar{B}) = A + A = A\end{aligned}$$

Primer 6. Uprostiti zadatu logičku funkciju:

$$F(A, B, C) = A \cdot \bar{B} \cdot \bar{C} + A \cdot \bar{B} \cdot C + A \cdot B \cdot \bar{C} + A \cdot B \cdot C$$

Rešenje:

$$F(A, B, C) = A \cdot \bar{C} \cdot (B + \bar{B}) + A \cdot C \cdot (B + \bar{B}) = A \cdot \bar{C} + A \cdot C = A \cdot (C + \bar{C}) = A$$

Primer 7. Uprošćavanjem izraza za zadatu logičku funkciju dokazati da vrednost funkcije F ne zavisi od vrednosti promenljive B :

$$F = A \cdot (B \cdot C + D) + A \cdot C$$

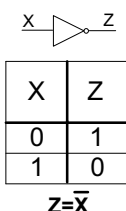
Rešenje: $F = A \cdot (C + D)$

Osnovne logičke operacije nad binarnim ciframa

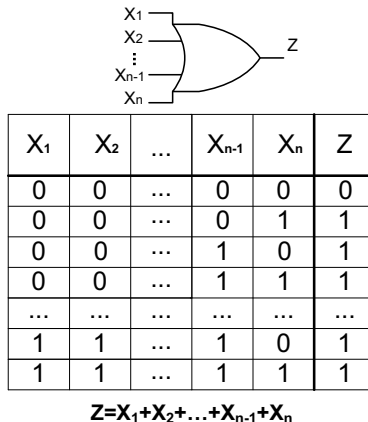
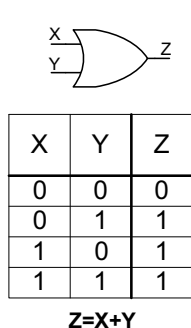
Postoje dve vrste logičkih operacija, zavisno od broja operanada koje u njima učestvuju, i to su:

- **unarne**, logičke operacije nad jednim operandom (negacija)
- **binarne**, logičke operacije nad dva operanda (sve druge operacije)

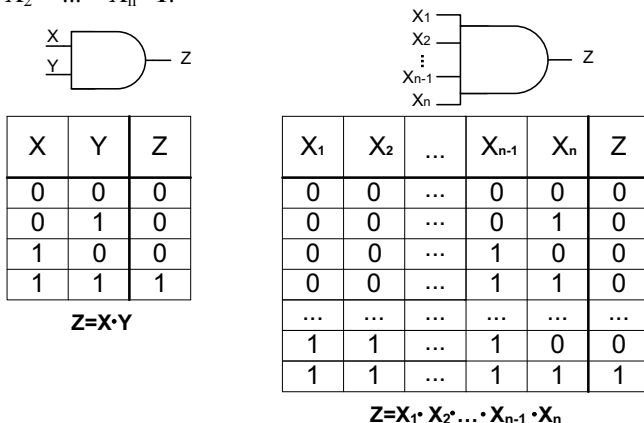
Negacija (NE, NOT) je najprostija logička operacija koja se obavlja nad jednim operandom. Zove se još inverzija ili komplementiranje. Operacija **NE** konvertuje vrednost 1 u vrednost 0 i obrnuto. U tabeli istinitosti za **NE** operaciju X je ulazna veličina (operand), a Z je izlazna veličina (rezultat).



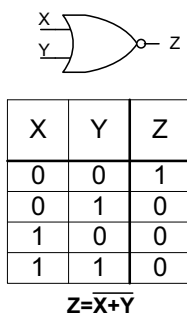
ILI operacija (OR) se vrši nad dve ili više ulaznih vrednosti, a naziva se još i logičko sabiranje (disjunkcija). Da bi rezultat **ILI** operacije imao vrednost **1** mora bar jedna ulazna veličina da ima vrednost **1**. U tabeli istinitosti za ILI operaciju nad dve ulazne vrednosti X i Y, kao i tabeli istinitosti za **n** ulaznih vrednosti $X_1, \dots, X_n$ dobija se rezultat $Z = 1$ kada su jedna ili više ulaznih vrednosti istovremeno jednake 1.



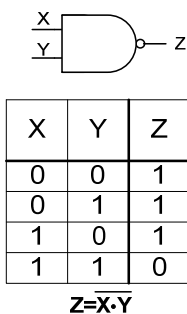
Operacija I (AND) se vrši nad dve ili više ulaznih vrednosti, a naziva se još i logičko množenje (konjunkcija). Rezultat I operacije je jednak **0**, ako je bar jedna ulazna vrednost jednaka nuli. Da bi rezultat I operacije bio $Z = 1$, moraju svi ulazni signali istovremeno da budu jednaki jedinici: $X = Y = 1$ tj. $X_1 = X_2 = \dots = X_n = 1$.



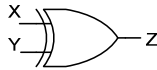
NILI operacija (NOR) daje rezultat **1** samo kada se na svim ulazima nalazi vrednost **0**.



NI operacija (NAND) daje rezultat **0** samo ako se na svim ulazima nalazi vrednost **1**.



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.



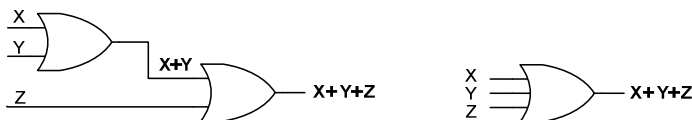
X	Y	Z
0	0	1
0	1	0
1	0	0
1	1	1

$$Z = \overline{X \oplus Y}$$

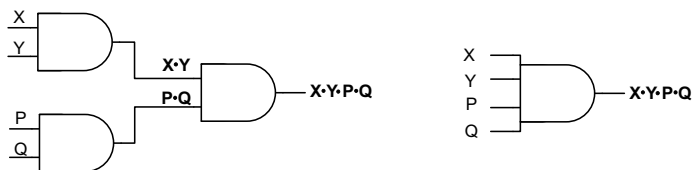
Viševalazna logička kola mogu da se realizuju primenom dvoulaznih kola

Primer:

- **ILI** kolo sa tri ulaza (troulazno kolo) može da se realizuje primenom dva dvoulazna ILI kola

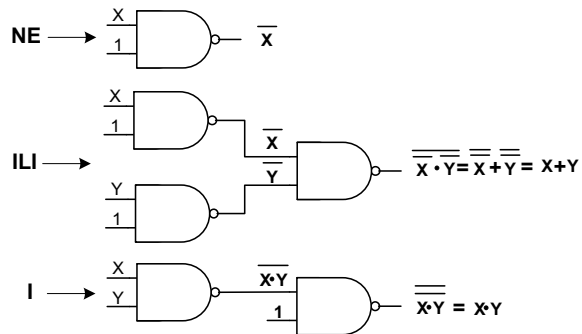


- **I** kolo sa četiri ulaza (četvoulazno kolo) može da se realizuje primenom tri dvoulazna I kola

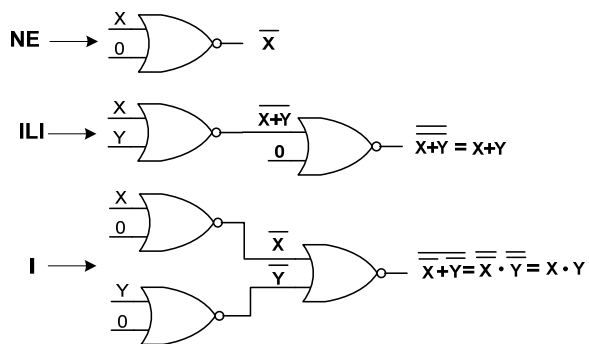


Osnovne logičke operacije mogu da se realizuju primenom samo jednog tipa logičkih kola

- Realizacija NE, ILI i I logičke operacije primenom **NI** kola



- Realizacija NE, ILI i I logičke operacije primenom **NILI** kola



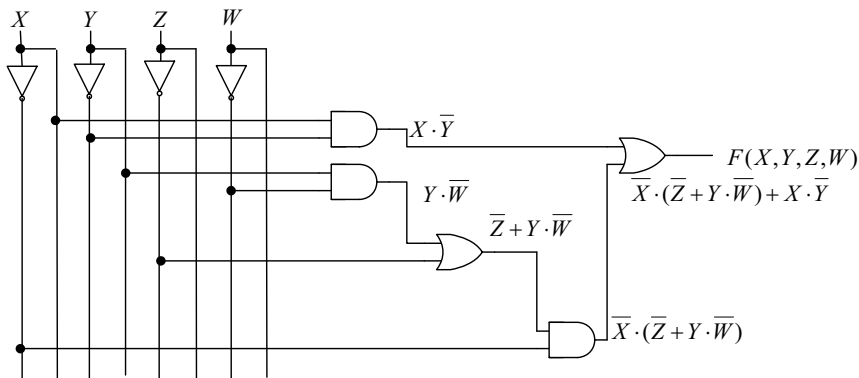
Primer 8. Nacrtati šemu logičke mreže kojom se data funkcija

$$F(X,Y,Z,W) = \overline{X} \cdot \overline{Z} + X \cdot \overline{Y} + \overline{X} \cdot Y \cdot \overline{W}$$
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:

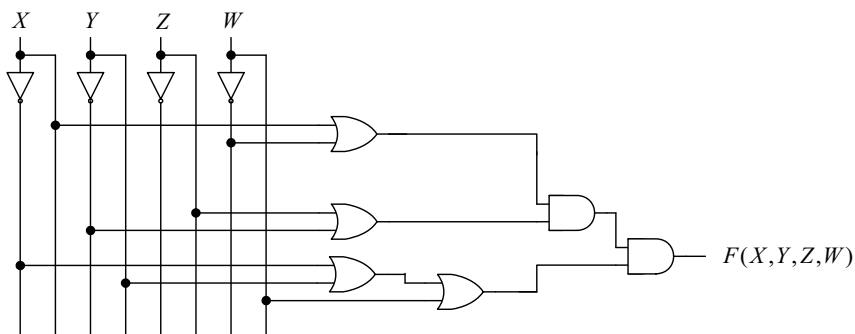
$$F(X,Y,Z,W) = \overline{X} \cdot \overline{Z} + X \cdot \overline{Y} + \overline{X} \cdot Y \cdot \overline{W} = \overline{X} \cdot (\overline{Z} + Y \cdot \overline{W}) + X \cdot \overline{Y}$$



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.

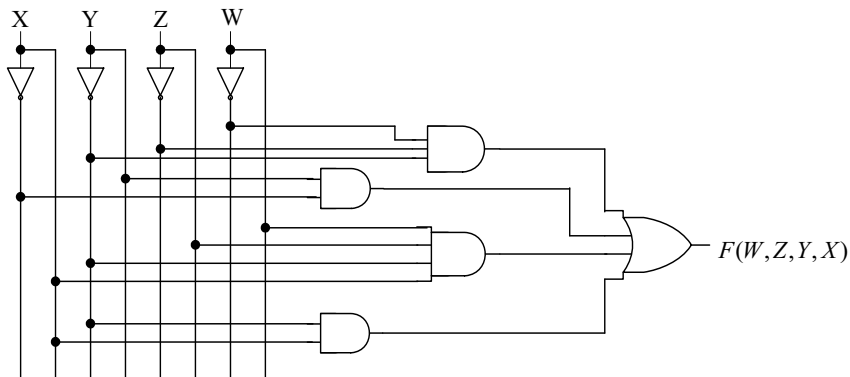
Rešenje:



Primer 10. Primenom **NE** kola i višoulaznih **ILI** i **I** logičkih kola nacrtati šemu logičke mreže kojom se realizuje funkcija F data izrazom:

$$F(W,Z,Y,X) = \overline{W} \cdot \overline{Z} \cdot \overline{Y} + Y \cdot \overline{X} + W \cdot Z \cdot \overline{Y} \cdot X + \overline{Y} \cdot X$$

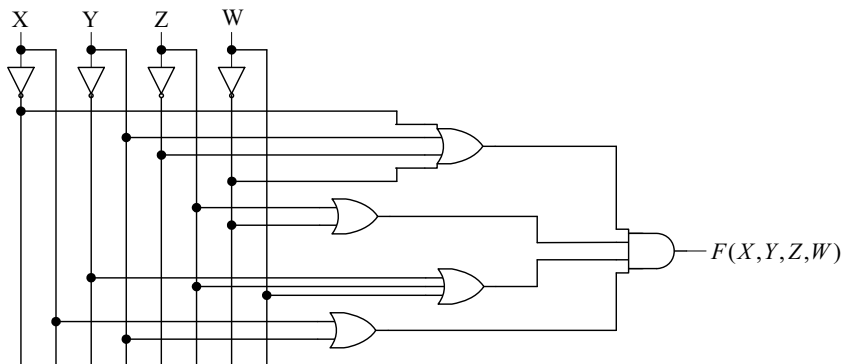
Rešenje:



Primer 11. Primenom **NE** kola i višeulaznih **ILI** i **I** logičkih kola nacrtati šemu logičke mreže kojom se realizuje funkcija F data izrazom:

$$F(X, Y, Z, W) = (\bar{X} + Y + \bar{Z} + \bar{W}) \cdot (Z + \bar{W}) \cdot (\bar{Y} + Z + W) \cdot (X + Y)$$

Rešenje:



Primer 12. Nacrtati šemu logičke mreže kojom se realizuje funkcija data izrazom:

$$F(A, B, C, D) = (A + \bar{B} + C) \cdot (B + \bar{C} + D) \cdot (\bar{A} + \bar{C} + \bar{D})$$

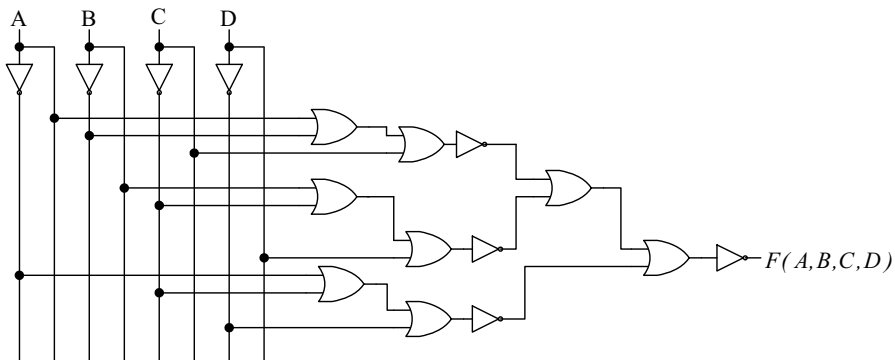
primenom **NE** kola i dvoulaznih **ILI** kola.

Rešenje:

$$F(A, B, C, D) = \overline{(A + \bar{B} + C) \cdot (B + \bar{C} + D) \cdot (\bar{A} + \bar{C} + \bar{D})}$$

$$F(A, B, C, D) = \overline{(A + \bar{B} + C) \cdot (B + \bar{C} + D) + (\bar{A} + \bar{C} + \bar{D})}$$

$$F(A, B, C, D) = \overline{(A + \bar{B} + C) + (B + \bar{C} + D) + (\bar{A} + \bar{C} + \bar{D})}$$



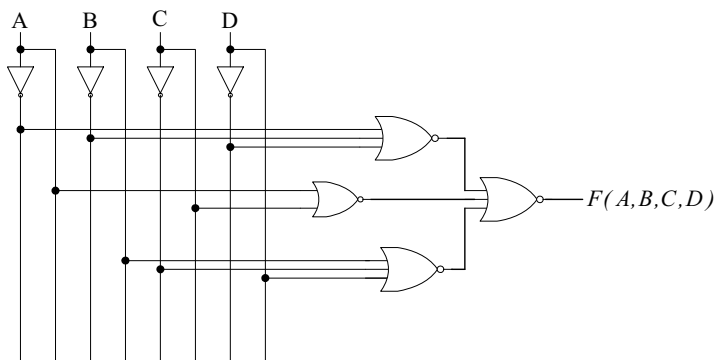
Primer 13. Primenom **NE** kola i višeuilaznih **NLI** kola nacrtati šemu logičke mreže kojom se realizuje funkcija F data izrazom:

$$F(A, B, C, D) = (\bar{A} + \bar{B} + \bar{D}) \cdot (A + C) \cdot (B + \bar{C} + D)$$

Rešenje:

$$F(A, B, C, D) = \overline{(\bar{A} + \bar{B} + \bar{D}) \cdot (A + C) \cdot (B + \bar{C} + D)}$$

$$F(A, B, C, D) = \overline{(\bar{A} + \bar{B} + \bar{D}) + (A + C) + (B + \bar{C} + D)}$$

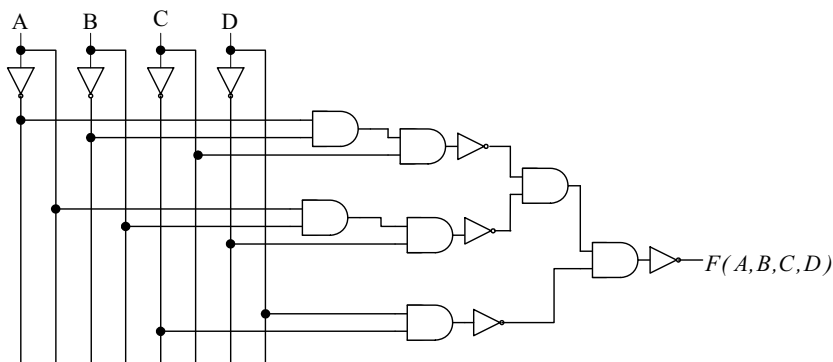


Primer 14. Primenom **NE** kola i dvoulaznih **I** kola nacrtati šemu logičke mreže kojom se realizuje funkcija F data izrazom:

$$F(A, B, C, D) = (\bar{A} \cdot \bar{B} \cdot C) + (A \cdot B \cdot \bar{D}) + (D \cdot \bar{C})$$

Rešenje:

$$F(A, B, C, D) = \overline{(\bar{A} \cdot \bar{B} \cdot C) \cdot (A \cdot B \cdot \bar{D}) \cdot (D \cdot \bar{C})}$$



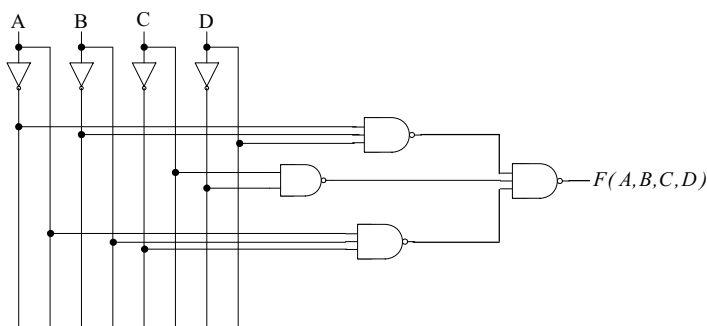
Primer 15. Primenom **NE** kola i višeulaznih **NI** kola nacrtati šemu logičke mreže kojom se realizuje funkcija F data izrazom:

$$F(A, B, C, D) = (\bar{A} \cdot \bar{B} \cdot D) + (C \cdot \bar{D}) + (A \cdot B \cdot \bar{C})$$

Rešenje:

$$F(A, B, C, D) = (\bar{A} \cdot \bar{B} \cdot D) + (C \cdot \bar{D}) + (A \cdot B \cdot \bar{C})$$

$$F(A, B, C, D) = (\bar{A} \cdot \bar{B} \cdot D) \cdot (C \cdot \bar{D}) \cdot (A \cdot B \cdot \bar{C})$$



Primer 16. Logička funkcija je zadata izrazom:

$$F(A, B, C, D) = A \cdot D + \bar{B} \cdot C + \bar{A} \cdot B \cdot \bar{C} \cdot D + B \cdot C$$

- Odrediti vrednost funkcije ako se na ulaz A dovede niz logičkih nivoa 1010011, na ulaz B: 11010101, na ulaz C: 10101000 i na ulaz D: 10101011.
- Nacrtati šemu logičke mreže kojom se realizuje data funkcija, koristeći **NE** i dvoulazna **I** kola.

Rešenje:

- Da bi se lakše odredila vrednost funkcije, može se primeniti analitičko uprošćavanje datog izraza primenom identiteta iz Primera 3.

$$\begin{aligned}
 F(A,B,C,D) &= A \cdot D + \bar{B} \cdot C + \bar{A} \cdot B \cdot \bar{C} \cdot D + B \cdot C = D \cdot (A + \bar{A} \cdot B \cdot \bar{C}) + C \cdot (B + \bar{B}) = \\
 &= D \cdot (A + B \cdot \bar{C}) + C = A \cdot D + C + \bar{C} \cdot B \cdot D = \\
 &= A \cdot D + C + B \cdot D = C + D \cdot (A + B)
 \end{aligned}$$

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

$$F(A,B,C,D)=\mathbf{10101011}$$

- 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:

$$F(X,Y,Z,W) = (\bar{Y} + \bar{Z}) \cdot (X + W) \cdot (\bar{Y} + Z + \bar{W}) \cdot (X + W)$$

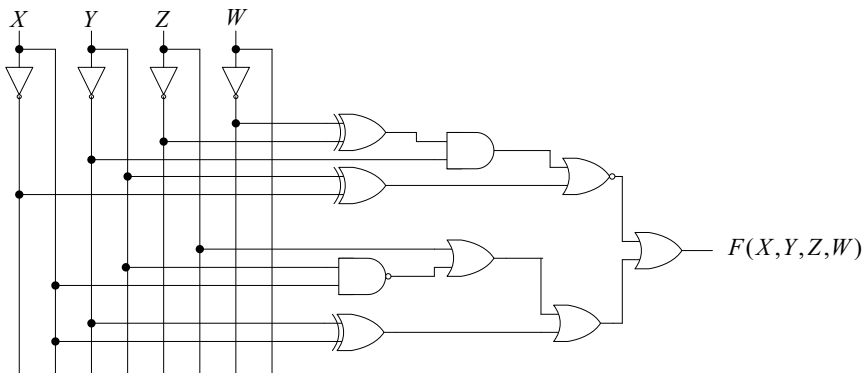
ako su vrednosti ulaznih promenljivih:

- a) $X=0, Y=1, Z=1, W=1$
b) $X=1, Y=0, Z=0, W=0$

Rešenje:

- a) $F(X,Y,Z,W) = F(0,1,1,1) = (\bar{1} + \bar{1}) \cdot (0 + 1) \cdot (\bar{1} + 1 + \bar{1}) \cdot (0 + 1) = 0 \cdot 1 \cdot 1 \cdot 1 = 0$
b) $F(X,Y,Z,W) = F(1,0,0,0) = (\bar{0} + \bar{0}) \cdot (1 + 0) \cdot (\bar{0} + 0 + \bar{0}) \cdot (1 + 0) = 1 \cdot 1 \cdot 1 \cdot 1 = 1$

Primer 18. a) Odrediti izraz za funkciju $F(X, Y, Z, W)$ koja opisuje rad kombinacione mreže prikazane na slici:



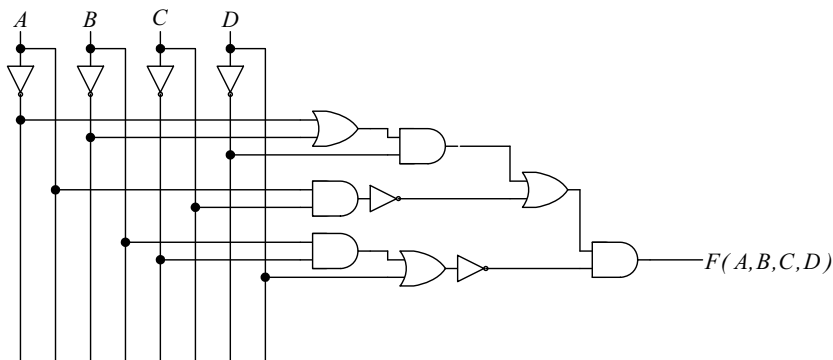
- b) Odrediti vrednost funkcije F ako su vrednosti promenljivih $X=1, Y=1, Z=0, W=0$

Rešenje:

a) $F(X, Y, Z, W) = (\overline{Z \oplus W}) \cdot \overline{Y} + (\overline{X \oplus Y}) + \overline{X \cdot Y} + Z + X \oplus \overline{Y}$

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:



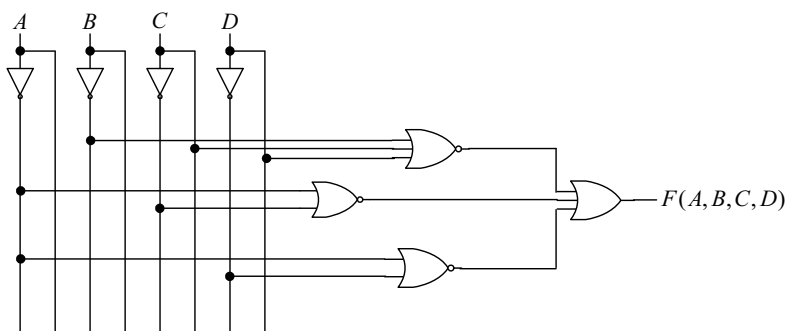
b) Odrediti vrednost funkcije F ako su vrednosti promenljivih $A = 0, B = 0, C = 1, D = 1$

Rešenje:

a) $F(A, B, C, D) = ((\overline{A} + \overline{B}) \cdot \overline{D} + \overline{A \cdot C}) \cdot (\overline{B \cdot C} + D)$

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 višezlaznih NI logičkih kola.

Rešenje:

a) $F(A, B, C, D) = \overline{\overline{B} + C + D} + \overline{\overline{A} + C} + \overline{\overline{A} + D}$

b) Primenom De Morganove teoreme u izrazu dobijenom u a) i dvostrukom negacijom izraza, dobija se:

$$F(A, B, C, D) = \overline{B \cdot C \cdot D} \cdot \overline{A \cdot C} \cdot \overline{A \cdot D}$$

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: $F(A, B, C) = \overline{\overline{A} \cdot B} + C$.

- 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) $F(A, B, C) = \overline{\overline{A} \cdot B} + C = A + B + C$, 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 višoulaznih **I** i **ILI** logičkih kola.
- b) Primenom **NE** kola i višoulaznih **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 višoulaznih **I** i **ILI** logičkih kola.
- b) Primenom **NE** kola i višoulaznih **NI** kola.

Primer 28. Nacrtati šeme kombinacionih mreža kojima se realizuju funkciju datu 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 višoulaznih **ILI** i **I** logičkih kola.
- b) Primenom **NE** kola i višoulaznih **ILI** kola.

Primer 29. Nacrtati šemu kombinacione mreže kojom se realizuju funkcije date izrazima:

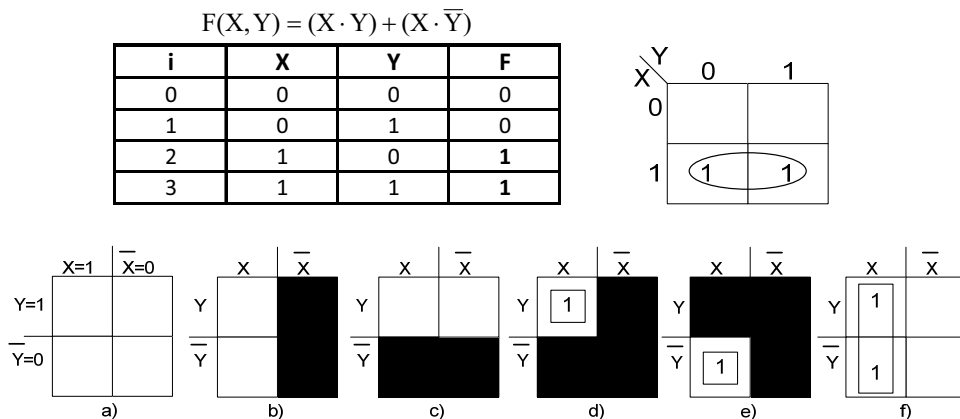
$$F(A, B, C, D) = \overline{B} \cdot \overline{C} \cdot D + A \cdot B \cdot C + B \cdot \overline{C} \cdot \overline{D} + A \cdot D$$

- a) Primenom **NE** kola i višoulaznih **ILI** i **I** logičkih kola.
- b) Primenom **NE** kola i višoulaznih **I** kola.

6. MINIMIZACIJA LOGIČKIH FUNKCIJA PRIMENOM KARNOOVIH MAPA

Određivanje minimalne disjunktivne forme (MDF)

- **Karnoova mapa za funkciju dve promenljive X i Y** slika 1. a) sastoji se od 4 polja. Svako od tih polja rezervisano je za 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 preslikava funkcija dve promenljive na odgovarajuću Karnoovu mapu. Na primer, ako je data funkcija $F(X,Y) = (X \cdot Y) + (X \cdot \bar{Y})$, u njoj se pojavljuju dve konjunkcije: $X \cdot Y$ i $X \cdot \bar{Y}$. Mesto konjunkcije $X \cdot Y$ u mapi određuje se na sledeći način:
 - Pošto konjunkcija $x \cdot y$ sadrži promenljivu x (bez negacije) onda se ona mora nalaziti u oblasti gde je $X=1$ (a osenčimo oblast $X=0$, slika 1. b)), a kako sadrži i promenljivu Y to mora biti i u oblasti gde je $Y=1$ (neosenčena oblast na slici 1. c)). Dakle, konjunkcija $X \cdot Y$ se nalazi na polju koje je u preseku ove dve oblasti, u koje upisujemo 1 (slika 1.d)). Na isti način se određuje polje koje sadrži konjunkciju $X \cdot \bar{Y}$ slika 1. e), a popunjena Karnoova mapa za čitavu funkciju F data je na slici 1. f)



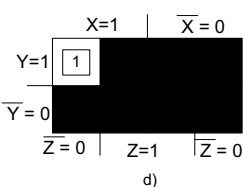
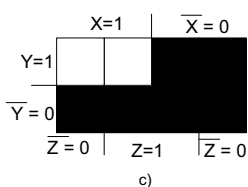
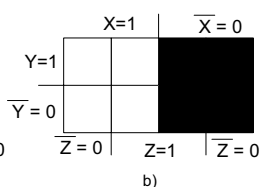
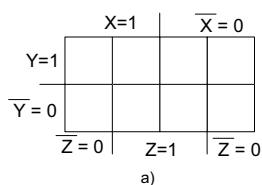
Slika 1. Karnoova mapa za funkciju sa dve promenljive

- **Karnoova mapa za funkciju tri promenljive X, Y i Z**, ima izgled prikazan na Slici 2.a). I u ovom slučaju svako polje mape rezervisano je za jednu tačno određenu konjunkciju promenljivih ili njihovih negacija. Kako se za zadatu konjunkciju pronalazi odgovarajuće polje? Pronalaženje odgovarajućeg polja ide postepeno, eliminacijom polja koja ne zadovoljavaju. Neka je data konjunkcija $X \cdot Y \cdot \bar{Z}$.
 - Pošto se u zadatom izrazu promenljiva x javlja u afirmativnom obliku, u obzir dolaze 4 leva polja (desna polovina mape se osenči jer ta polja ne dolaze u obzir, slika 2. b)). Druga promenljiva je y , pa se 4 leva polja smanjuju samo na 2, i to u gornjoj vrsti, a donja vrsta se osenči (slika 2. c)). Konačno, treći element je $\bar{z}$, pa treba osenčiti polja u sredini u kojima je $Z=1$ (slika 2. d)). Preostalo, neosenčeno polje određuje polje datog izraza i u njega upisujemo 1, slika 2. d). Analognim postupkom može se naći polje za bilo koju konjunkciju 3 promenljive.

$$F(X, Y, Z) = (X \cdot Y \cdot \bar{Z})$$

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

YZ	00	01	11	10
X=0				
X=1				1



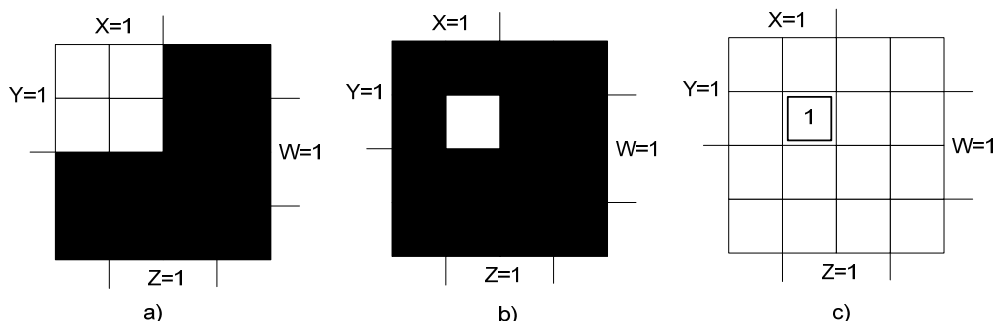
Slika 2. Karnoova mapa za tri promenljive

- **Karnoova mapa za funkciju sa četiri promenljive X, Y, Z i W**, ima izgled prikazan na slici 3. a). Istim postupkom eliminacije odredićemo položaj konjunkcije: $X \cdot Y \cdot Z \cdot W$.
 - Pošto su sve promenljive bez negacije, treba osenčiti polja u kojima promenljive uzimaju vrednost 0 (slika 3. a) i b)), a preostalo neosenčeno polje je polje u koje se upisuje 1 koja odgovara ovoj konjunkciji, slika 3. c).
 -

$$F(X, Y, Z) = X \cdot Y \cdot Z \cdot W$$

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

ZW	00	01	11	10
XY=00				
XY=01				
XY=11			1	
XY=10				



Slika 3. Karnoova mapa za četiri promenljive

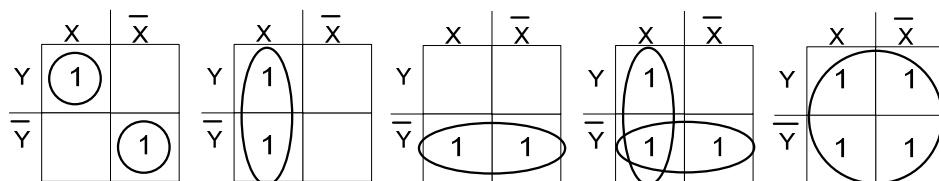
Proces minimizacije u cilju dobijanja analitičkog izraza u **minimalnoj disjunktivnoj formi (MDF)** sastoji se u grupisanju jedinica u okviru mape u veće celine. Pri tome se nastoji da ove celine budu što veće, ali one ne smeju da sadrže polja koja nemaju 1. Kada se grupišu polja sa treba primeniti sledeća opšta pravila:

- Treba grupisati po 1, 2, 4, 8, ili 16 jedinica. Grupe sa brojem jedinica koji nije oblika 2^N nisu dozvoljene.
- Treba se truditi da grupe budu što veće i po cenu preklapanja dela grupe sa drugom.
- Preklapanje grupa ne smeta ako se time prave veće grupe.
- Treba se truditi da ukupan broj grupa bude što manji.
- Moraju se grupisati sve jedinice.
- Treba formirati minimalan broj što većih grupa tako da se obuhvate 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.

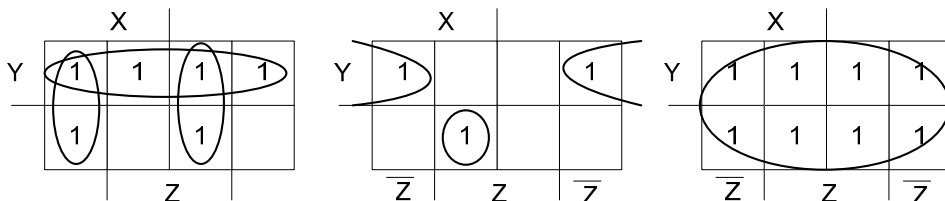
Minimalna disjunktivna forma ima onoliko članova koliko je grupa napravljeno. Svaki član je logički proizvod (logička I funkcija) do četiri ulaza (za funkciju 4 promenljive) ili njihovih invertovanih vrednosti. Ako grupu čine dve jedinice, član ima tri elementa; ako grupu čine četiri jedinice, član ima dva elementa, a samo jedan element ako smo grupisali osam jedinica. Članovi su međusobno povezani logičkom ILI funkcijom.

Primeri grupisanja polja u **Karnoovoj mapi sa 2 promenljive** dati su na slici 4.



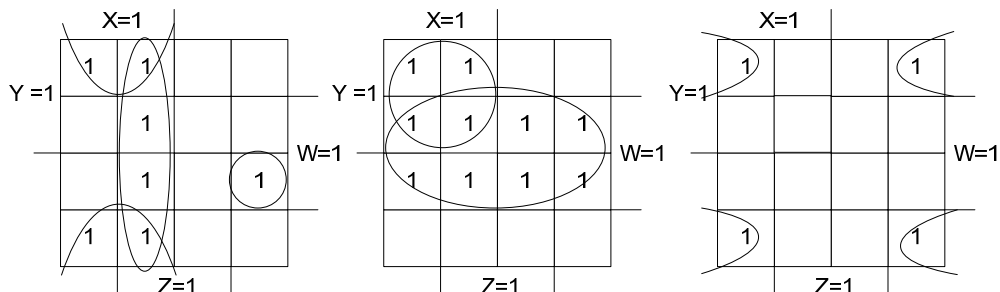
Slika 4. Grupisanje jedinica u Karnoovoj mapi sa dve promenljive

U Karnoovoj mapi sa 3 promenljive može se uočiti da se područje nezavisno promenljive $\bar{Z}$ proteže direktno sleva na desnu stranu tabele. To se može zamisliti kao da je tabela nacrtana na valjku, čiji je omotač po visini razvijen u ravan. Primeri grupisanja polja u Karnoovoj mapi sa četiri promenljive prikazani su na Slici 5.



Slika 5. Grupisanje jedinica u Karnoovoj mapi sa tri promenljive

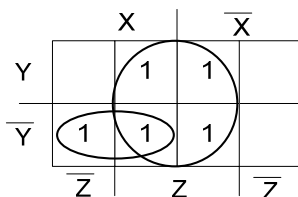
Primeri pravilnog grupisanja polja u **Karnoovim mapama sa 4 promenljive** dati su na Slici 6. Kao i u prethodnim primerima, primenjeno je pravilo da treba formirati minimalan broj grupa tako da budu obuhvaćene sve jedinice u Karnoovoj mapi.



Slika 6. Grupisanje jedinica u Karnoovoj mapi sa četiri promenljive

Određivanje analitičkog oblika minimalne disjunktivne forme logičke funkcije (MDF) na osnovu formiranih grupa 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 disjunktija ovih pojedinačnih članova. Primer mape sa 3 promenljive, gde su već formirane grupe prikazan je na slici 7



Slika 7. Određivanje analitičkog izraza za grupu jedinica

Na osnovu grupisanih polja sa slike 7 :

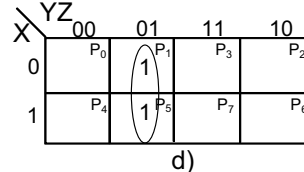
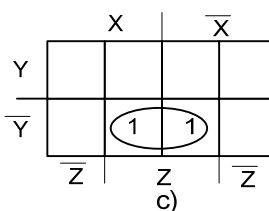
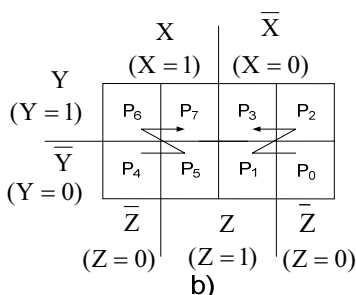
- Prvo definišemo odgovarajuću konjunkciju za grupu od 4 srednje jedinice. U ovom slučaju grupa ima jedinice u poljima X ($X=1$) i $\bar{X}$ ($X=0$), pa ta varijabla neće figurisati u rezultujućem izrazu. U datoj grupi ima polja i sa $Y=1$ i sa $Y=0$, pa ni ova varijabla ne utiče na rezultujuću konjunkciju. Grupa je definisana u svim poljima sa $Z=1$ (oblast Z), tako da varijabla Z ulazi u konjunkciju. Konačno, prvi faktor minimalne logičke funkcije je Z (nije vezan konjunkcijom jer je samo jedna promenljiva).
- Druga grupa sastoji se od dva polja u donjem levom uglu mape. Vrednost varijable X je jedinstvena u ovoj grupi, pa ulazi u konjunkciju. Zatim, u grupi postoji polje sa varijablom Z , ali i sa $\bar{Z}$, pa, prema tome, ova varijabla ne ulazi u konačni izraz za ovu grupu. Kako je za ovu grupu i varijabla $\bar{Y}$ jedinstvena, to i ona ulazi u izraz, pa je konačan oblik konjunkcije za ovu grupu: $X \cdot \bar{Y}$. Minimalni oblik ove funkcije dobija se povezivanjem konjunkcija za pojedine grupe operatorom disjunktije : $F(X, Y, Z) = Z + X \cdot \bar{Y}$

Prilikom minimizacije logičkih funkcija moguće je početi od tablice istinitosti, a ne samo od disjunktivne normalne forme, slika 8. **Da bi se tablica istinitosti preslikala na Karnoovu mapu**, posmatraju se svi redovi gde je funkcija jednaka **1** i odgovarajuća kombinacija **1** i **0** promenljivih definiše 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.
- Međutim, ako vrednost funkcije zavisi od vrednosti promenljivih X , Y i Z kao što je prikazano na slici 8 a), onda se Karnoova mapa može popuniti po šablonu, jer svakoj vrsti iz tabele uvek odgovara tačno određeno mesto u Karnoovoj mapi, kao što je prikazano na Slici 8 b). Kako je položaj konjunkcija fiksiran, to se samo umesto $P_0, P_1, \dots, P_7$ upišu jedinice ili nule (Slika 8.c i d)).

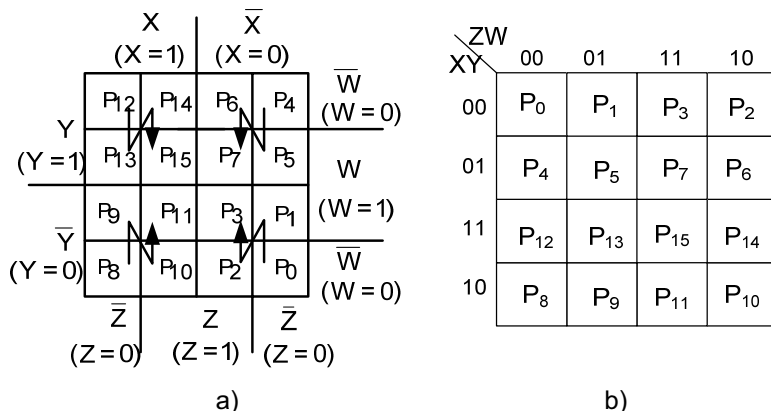
0	0	0	0	P_0	0
1	0	0	1	P_1	1
2	0	1	0	P_2	0
3	0	1	1	P_3	0
4	1	0	0	P_4	0
5	1	0	1	P_5	1
6	1	1	0	P_6	0
7	1	1	1	P_7	0

a)



Slika 8. Popunjavanje Karnoove mape i minimizacija funkcije F

Raspored polja u Karnoovim mapama za četiri ulazne promenljive, koje su prikazane na Slici 9, treba zamisliti kao da su nacrtana na torusu, pa razvijena 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.

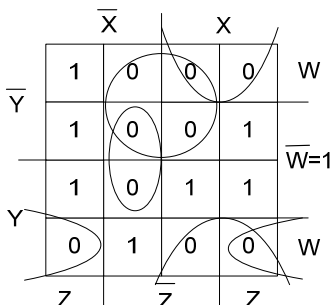


Slika 9. Karnoove mape za četiri promenljive

Određivanje minimalne konjunktivne forme (MKF)

- **Za nalaženje minimalne konjunktivne forme (MKF) funkcije**, treba najpre popuniti Karnoovu mapu i to na isti način kao kod disjunktivne forme. Zatim treba grupisati pojedinačna polja, parove, kvartete i oktete nula i to na isti način kao što se grupišu jedinice kod disjunktivne forme. Na kraju treba odrediti analitički izraz za minimalne disjunktivne. Drugim rečima, polja u oblasti gde je **promenljiva jednaka nuli** uzimaju se kao afirmacija, npr. oblast $X=0$ je oblast X , a oblast $X=1$ je oblast $\bar{X}$. Tabela na Slici 10 pokazuje Karnoovu mapu za dobijanje minimalne konjunktivne forme funkcije.
- **MKF** ima onoliko logičkih suma (disjunktija) koliko u tabeli postoji zaokruženih polja, parova, kvarteta i okteta nula. Finalni oblik MKF dobija se kada se napravi logički proizvod (konjunktija) svih logičkih suma (disjunktija):

$$F = (\bar{Y} + \bar{Z}) \cdot (\bar{X} + \bar{Z} + \bar{W}) \cdot (X + W) \cdot (Y + Z + W)$$



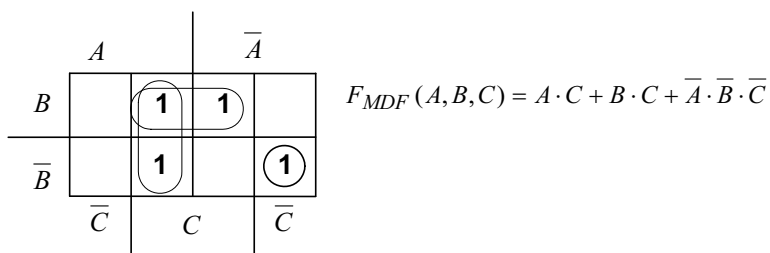
Slika 10. Primer Karnoove mape za funkciju sa četiri promenljive

Primer 1. Primenom Karnoove mape odrediti minimalnu disjunktivnu formu (MDF) funkcije koja je data izrazom:

$$Y(A, B, C, D) = A \cdot B \cdot C + A \cdot \bar{B} \cdot C + \bar{A} \cdot B \cdot C + \bar{A} \cdot \bar{B} \cdot \bar{C}$$

Rešenje:

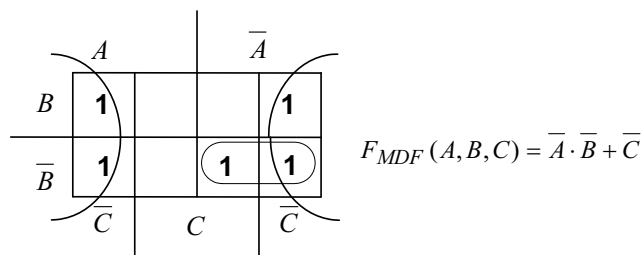
Izraz za funkciju Y sastoji se od sume potpunih proizvoda, odnosno može da se predstavi u obliku: $Y(A, B, C) = \sum(0, 3, 5, 7)$



Primer 2. Primenom Karnoove mape odrediti minimalnu disjunktivnu (MDF) funkcije F koja je zadata kao suma potpunih proizvoda:

$$F(A, B, C) = \sum(0, 1, 2, 4, 6)$$

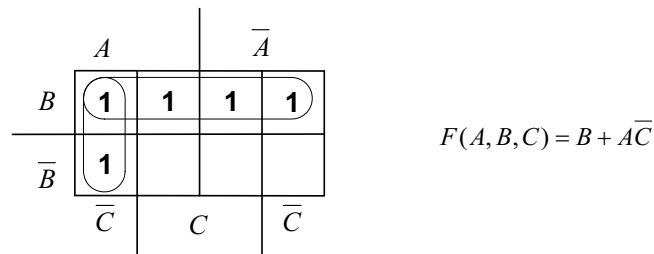
Rešenje:



Primer 3. Primenom Karnoove mape odrediti minimalnu disjunktivnu (MDF) funkcije F koja je zadata kao zadata na sledeći način:

$$F(A, B, C) = \bar{A}\bar{B}\bar{C} + A\bar{C} + BC$$

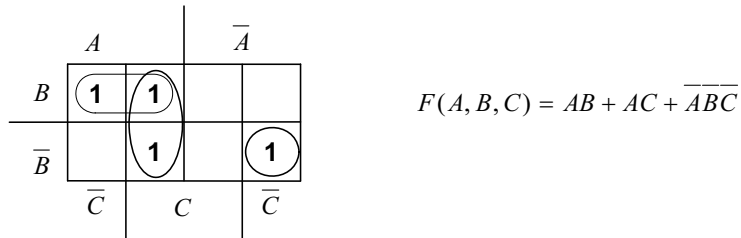
Rešenje:



Primer 4. Primenom Karnoove mape izvršiti minimizaciju funkcije F minimalnom disjunktivnom formulom (**MDF**), ako je funkcije F zadata na sledeći način:

$$F(A, B, C) = \overline{A}\overline{B}C + \overline{A}BC + AB + AC$$

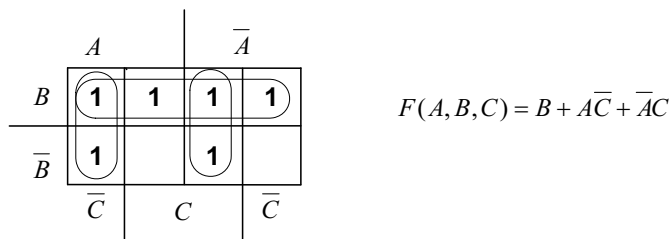
Rešenje:



Primer 5. Primenom Karnoove mape izvršiti minimizaciju funkcije F minimalnom disjunktivnom formulom (**MDF**), ako je funkcije F zadata na sledeći način:

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

Rešenje:



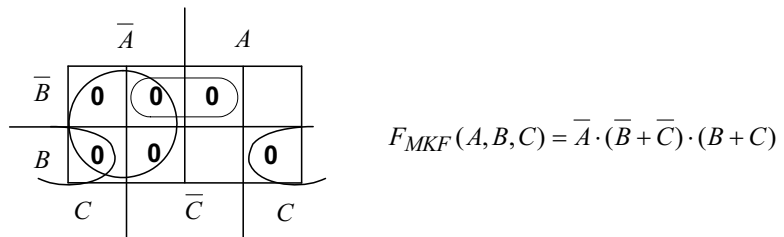
Primer 6. Primenom Karnoove mape odrediti minimalnu konjuktivnu formu (MKF) funkcije Y koja je zadata kao proizvod potpunih suma:

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

Rešenje:

Izraz za funkciju Y sastoji se od proizvoda potpunih suma, odnosno može da se predstavi u obliku:

$$Y(A, B, C) = \prod (0, 3, 4, 5, 6, 7)$$



Primer 7. Primenom Karnoove mape odrediti minimalnu konjunktivnu formu (MKF) funkcije F koja je zadata kao suma potpunih proizvoda:

$$F(X, Y, Z) = \prod (0, 1, 4, 5, 6)$$

Rešenje:

	$\bar{X}$		X	
$\bar{Y}$	0			
Y	0	0	0	0
	Z	$\bar{Z}$	Z	

$$F_{MKF}(X, Y, Z) = Y \cdot (\bar{X} + Z)$$

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.

$$X(A, B, C) = \bar{A}\bar{B}\bar{C} + AC[\bar{B} + A(B + \bar{C})]$$

Rešenje:

Izraz za funkciju $X(A, B, C)$ transformiše se na sledeći način:

$$\begin{aligned} X(A, B, C) &= \bar{A} \cdot B \cdot \bar{C} + A \cdot C[\bar{B} + A \cdot (B + \bar{C})] = \bar{A} \cdot B \cdot \bar{C} + A \cdot C \cdot [\bar{B} + A \cdot B + A \cdot \bar{C}] = \\ &= \bar{A} \cdot B \cdot \bar{C} + A \cdot C \cdot \bar{B} + A \cdot C \cdot A \cdot B + A \cdot C \cdot A \cdot \bar{C} = \bar{A} \cdot B \cdot \bar{C} + A \cdot \bar{B} \cdot C + A \cdot B \cdot C \end{aligned}$$

Odavde sledi da funkcija $X(A, B, C)$ može da se predstavi kao suma potpunih proizvoda, odnosno u disjunktivnoj normalnoj formi: $X_{DNF}(A, B, C) = \sum (2, 5, 7)$. Na osnovu toga se popunjava Karnoova mapa.

	A		$\bar{A}$	
B		1		1
$\bar{B}$		1		
	$\bar{C}$	C	$\bar{C}$	

$$X_{MDF}(A, B, C) = \bar{A} \cdot B \cdot \bar{C} + A \cdot C$$

Primer 9. Data je logička funkcija

$$F = \bar{A} \cdot C \cdot D + \bar{B} \cdot \bar{C} \cdot \bar{D} + \bar{A} \cdot \bar{B} \cdot C + \bar{A} \cdot \bar{B} \cdot \bar{C} \cdot D + A \cdot \bar{B} \cdot C \cdot \bar{D}$$

Koristeći Karnoovu mapu pronaći minimalnu disjunktivnu formu ove funkcije

Rešenje:

Da se podsetimo:

Karnoovu mapu 4×4 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:

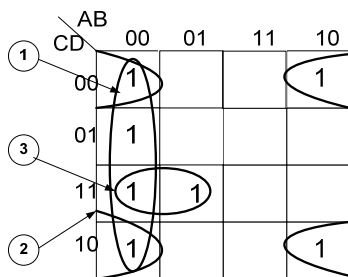
CD \ AB	AB			
	00	01	11	10
00	0	4	12	8
01	1	5	13	9
11	3	7	15	11
10	2	6	14	10

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 funkciju 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 odgovara prvom članu $\bar{A} \cdot C \cdot D$ 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

CD \ AB	AB			
	00	01	11	10
00				
01				
11				
10				

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 $\bar{A} \cdot C \cdot D$ može se doći i analitički, odnosno proširivanjem nepotpunog logičkih proizvoda: $\bar{A}CD = A(B + \bar{B})CD = \bar{A}BCD + \bar{A}\bar{B}CD$, 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:



$$F = \underline{\bar{B} \cdot \bar{D}} + \underline{\bar{A} \cdot C \cdot D} + \underline{\bar{A} \cdot \bar{B}}$$

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
<u>0</u>	<u>0</u>	<u>1</u>	<u>0</u>
<u>A</u>	<u>B</u>	-	-

Ulaz A je za celu grupu na nuli, ulaz B je za celu grupu na nuli, a ulazi C 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 $\bar{A} \cdot \bar{B}$.

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
<u>1</u>	<u>0</u>	<u>1</u>	<u>0</u>
-	<u>B</u>	-	<u>D</u>

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: $\bar{B} \cdot \bar{D}$.

Grupa označena sa ③ nastala je od jedinica za sledeće ulaze:

$$\begin{array}{cccc} \mathbf{A} & \mathbf{B} & \mathbf{C} & \mathbf{D} \\ 0 & 0 & 1 & 1 \\ \hline 0 & 1 & 1 & 1 \\ \hline \overline{A} & - & C & D \end{array}$$

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 između prve i poslednje su izlazi po binarnom redu. **x** je neodređen izlaz koji može biti bilo nula ili jedinica. Napisati logički izraz koji predstavlja:

- Minimalnu disjunktivnu formu ove funkcije.
- 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

AB \ CD	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:

AB \ CD	00	01	11	10
00	1	0	1	1
01	0	0	1	X
11	0	0	0	0
10	0	X	1	1

$$F = \underline{AC} + \underline{AD} + \underline{B \bar{C} \bar{D}}$$

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 - $\bar{C}$ -

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 $\underline{AC}$.

Grupu označenu sa ② koja spaja dve jedinice 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:

$$\begin{array}{cccc} \mathbf{A} & \mathbf{B} & \mathbf{C} & \mathbf{D} \\ 0 & 0 & 0 & 0 \\ \hline 1 & 0 & 0 & 0 \\ \hline - & \overline{\mathbf{B}} & \overline{\mathbf{C}} & \overline{\mathbf{D}} \end{array}$$

Ulaz B je za celu grupu na jedinici, ulazi C i D takođe, a ulaz A 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:

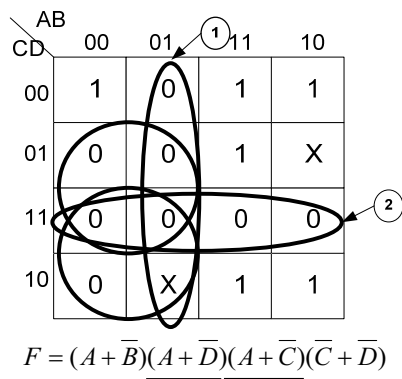
$$\begin{array}{cccc} \mathbf{A} & \mathbf{B} & \mathbf{C} & \mathbf{D} \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ \hline 1 & 0 & 1 & 0 \\ \hline \mathbf{A} & - & - & \overline{\mathbf{D}} \end{array}$$

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 $\overline{\mathbf{A}}\overline{\mathbf{D}}$.

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.



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
-	-	$\bar{C}$	$\bar{D}$

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 $\bar{C} + \bar{D}$.

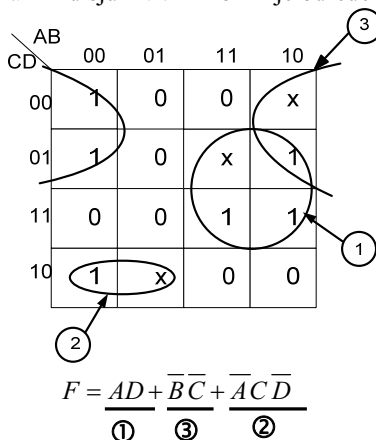
Postupak određivanja ostalih članova je isti. Funkcija koja se na ovaj način dobija kao minimalna disjunktivna forma je sledeća (članovi su podvučeni linijom kakvom je označena odgovarajuća grupa)

Primer 11. 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 između 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:

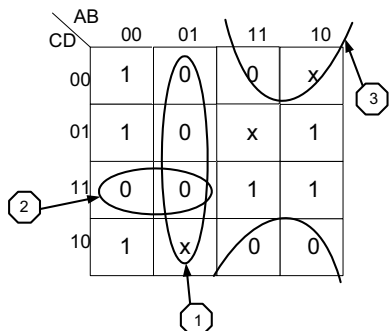
- Minimalnu disjunktivnu formu ove funkcije.
- Minimalnu konjunktivnu formu ove funkcije.

Rešenje:

- Jedna od mogućih minimalnih disjunktivnih formi je određena na osnovu tabele:



- b) Minimalna konjunktivna forma može da se odredi pomoću iste Karnoove mape, grupisanjem polja u kojima se nalaze nule.

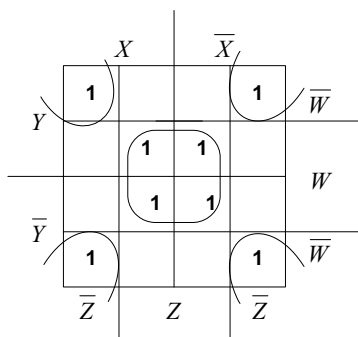


$$F = \underbrace{(A + \bar{B})}_{\textcircled{1}} \underbrace{(A + \bar{C} + \bar{D})}_{\textcircled{2}} \underbrace{(\bar{A} + D)}_{\textcircled{3}}$$

Primer 12. 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:

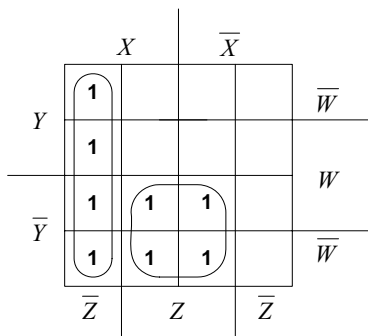


$$F_{MDF}(X, Y, Z, W) = ZW + \bar{Z}\bar{W}$$

Primer 13. 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:

		X		$\bar{X}$		
Y		1			1	$\bar{W}$
		1	1	1	1	W
$\bar{Y}$		1			1	$\bar{W}$
		1			1	W
		$\bar{Z}$		Z	$\bar{Z}$	

$$F_{MDF}(X, Y, Z, W) = \bar{Z} + YW$$

Primer 15. Primenom Karnoove mape izvršiti minimizaciju funkcije F minimalnom disjunktivnom formulom (**MDF**), ako je funkcije F zadata na sledeći način:

$$F(X, Y, Z, W) = YW + \bar{X}Z + X\bar{Y}W + \bar{X}Y\bar{Z} + \bar{X}Y\bar{Z}$$

Rešenje:

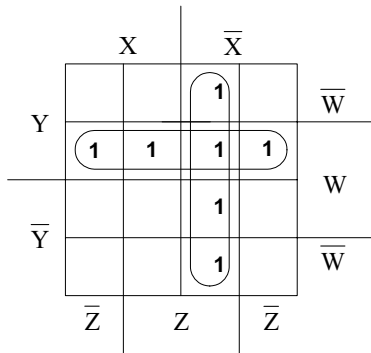
		X		$\bar{X}$		
Y				1	1	$\bar{W}$
		1	1	1	1	W
$\bar{Y}$		1	1	1	1	$\bar{W}$
				1	1	W
		$\bar{Z}$		Z	$\bar{Z}$	

$$F_{MDF}(X, Y, Z, W) = \bar{X} + W$$

Primer 16. Primenom Karnoove mape izvršiti minimizaciju funkcije F minimalnom disjunktivnom formulom (**MDF**), ako je funkcije F zadata na sledeći način:

$$F(X, Y, Z, W) = XYW + \bar{X}Z + \bar{X}Y\bar{Z}W$$

Rešenje:



$$F_{MDF}(X, Y, Z, W) = \bar{X}Z + YW$$

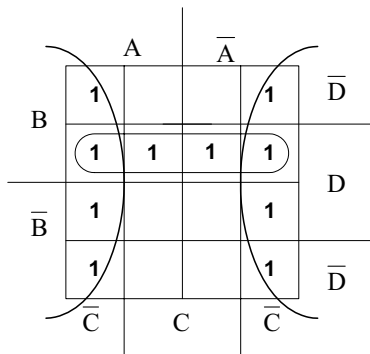
Primer 17. Za funkciju:

$$F(A, B, C, D) = A \cdot \bar{C} + B \cdot C \cdot D + \bar{A} \cdot \bar{C} \cdot D + \bar{C} \cdot \bar{D} + A \cdot B \cdot \bar{C} \cdot D$$

- Odrediti minimalnu disjunktivnu formu funkcije (MDF) primenom Karnoove mape.
- 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.
- Nacrtati šemu kombinacione mreže kojom se realizuje MDF date funkcije F, koristeći samo dva **NE** kola i dva **I** kola.

Rešenje:

a)



$$F_{MDF}(A, B, C, D) = \bar{C} + B \cdot D$$

b)

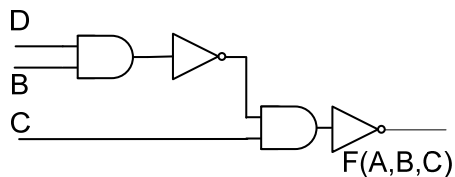
A	B	C	D	$\bar{C}$	$B \cdot D$	$\bar{C} + B \cdot D$
0	0	0	1	1	0	1
1	0	1	1	0	0	0
0	1	0	0	1	0	1
1	1	1	0	0	0	0
0	1	0	1	1	1	1
1	0	1	0	0	0	0
1	0	0	1	1	0	1
1	1	1	0	0	0	0

$$F(A, B, C, D) = 10101010$$

c)

$$F(A, B, C, D) = \overline{C} + B \cdot D$$

$$F(A, B, C, D) = \overline{\overline{\overline{C} + B \cdot D}} = \overline{\overline{C} \cdot \overline{B \cdot D}} = \overline{\overline{C} \cdot \overline{B} \cdot \overline{D}}$$



Primer 18. Za funkciju zadatu logičkim izrazom:

$$F(A, B, C, D) = A \cdot D + \overline{B} \cdot C + \overline{A} \cdot B \cdot \overline{C} \cdot D + B \cdot C$$

- 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.
- Odrediti izraz za minimalnu disjunktivnu formu (MDF) primenom Karnoove mape.

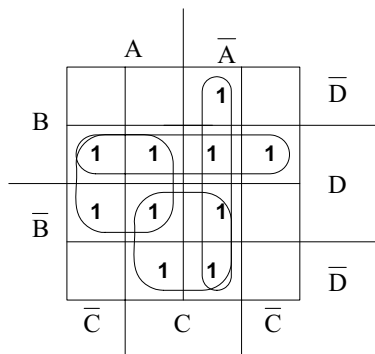
Rešenje:

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)

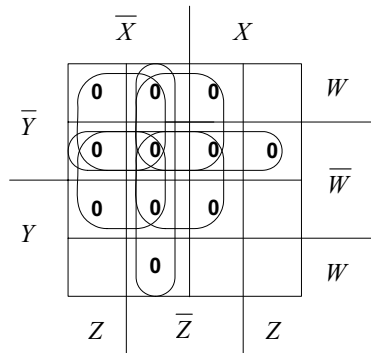


$$F(A, B, C, D) = AD + BD + \overline{C}\overline{B} + C\overline{A} = D(A + B) + C(\overline{A} + \overline{B})$$

Primer 19. Primenom Karnoove mape odrediti minimalnu konjunktivnu 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:

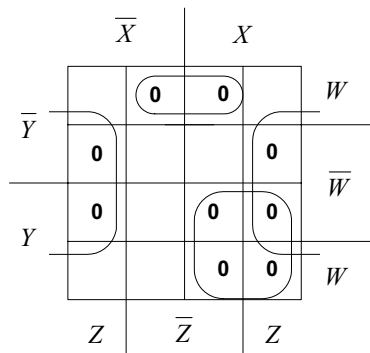


$$F_{MKF}(X, Y, Z, W) = (\bar{Z} + \bar{W}) \cdot (\bar{X} + \bar{W}) \cdot (\bar{Y} + \bar{W}) \cdot (\bar{X} + \bar{Z}) \cdot (\bar{X} + \bar{Y}) \cdot (\bar{Y} + \bar{Z})$$

Primer 20. Primenom Karnoove mape odrediti minimalnu konjunktivnu formu (**MKF**) funkcije F koja je zadata kao proizvod potpunih suma:

$$F(X, Y, Z, W) = \Pi(0, 1, 2, 3, 5, 6, 9, 13, 14)$$

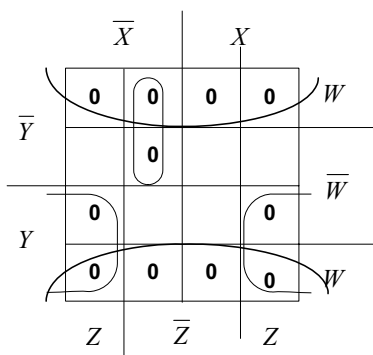
Rešenje:



$$F_{MKF}(X, Y, Z, W) = (X + Y) \cdot (Z + \bar{W}) \cdot (\bar{Y} + \bar{Z} + W)$$

Primer 21. Primenom Karnoove mape izvršiti minimizaciju funkcije F minimalnom konjunktivnom 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 $X, Y, Z, W = 1, 1, 1, 1$.

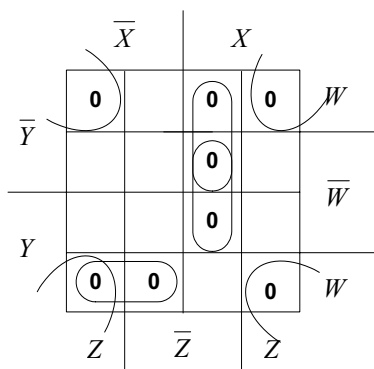
Rešenje:



Primer 22. Primenom Karnoove mape odrediti minimalnu konjunktivnu 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:



$$F_{MKF}(X, Y, Z, W) = (Z + W) \cdot (\bar{X} + Y + W) \cdot (X + \bar{Y} + \bar{Z}) \cdot (X + \bar{Z} + \bar{W})$$

Primer 23. 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 između 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:

- Minimalnu disjunktivnu formu ove funkcije.
- Minimalnu konjunktivnu formu ove funkcije.

Rešenje:

a) Minimalna disjunktivna forma:

AB \ CD	00	01	11	10
00	0	1	1	x
01	x	0	0	0
11	0	1	x	0
10	0	1	1	x

$$F = \underline{C \cdot B} + \underline{B \cdot \bar{D}}$$

b) Minimalna konjunktivna forma

AB \ CD	00	01	11	10
00	0	1	1	x
01	x	0	0	0
11	0	1	x	0
10	0	1	1	x

$$F = \underline{(C + \bar{D})} \underline{(\bar{A} + \bar{D})} \underline{(A + B)}$$

Primer 24. Za datu funkciju četiri logičke promenljive $F(X,Y,Z,W)$, čiji je skup vrednosti $F(0,1,1,x,0,1,x,1,0,1,0,x,0)$, primenom Karnoovih mapa odrediti njenu minimalnu disjunktivnu formulu (**MDF**), ako x predstavlja nedefinisano stanje funkcije.

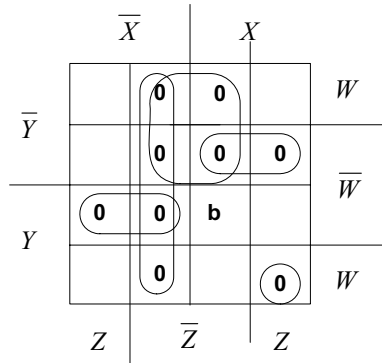
Rešenje:

	X		$\bar{X}$		
Y	$\bar{Z}$	Z	$\bar{Z}$	Z	$\bar{W}$
					W
$\bar{Y}$	1	x	1	1	1
Y	1	x	1	1	1
			1		$\bar{W}$

$$F_{MDF}(X,Y,Z,W) = \bar{X}Z + \bar{Z}W + YZ\bar{W}$$

Primer 25. Primenom Karnoove mape izvršiti minimizaciju funkcije F minimalnom konjunktivnom formulom (**MKF**), ako je funkcija $F(X,Y,Z,W)$ zadata u formi $F(011b100010001100)$, a sa **b** su označena nedefinisana stanja funkcije:

Rešenje:

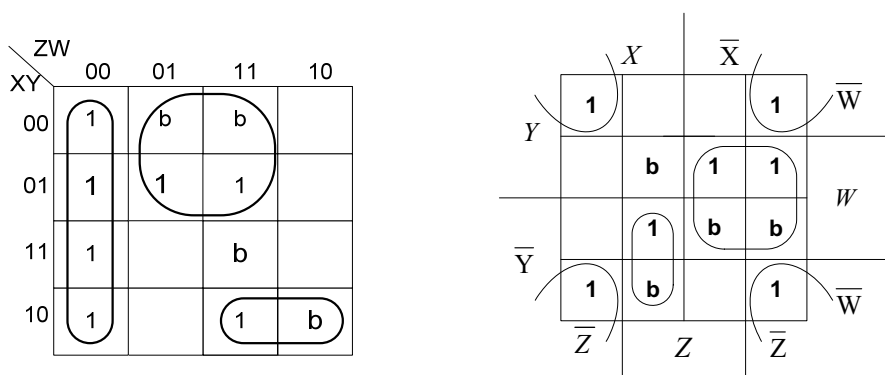


$$F_{MKF}(X,Y,Z,W) = (\bar{X} + \bar{Z}) \cdot (\bar{Y} + \bar{Z}) \cdot (X + \bar{Y} + \bar{W}) \cdot (\bar{X} + Y + \bar{W}) \cdot (X + Y + Z + W)$$

Primer 26. 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:



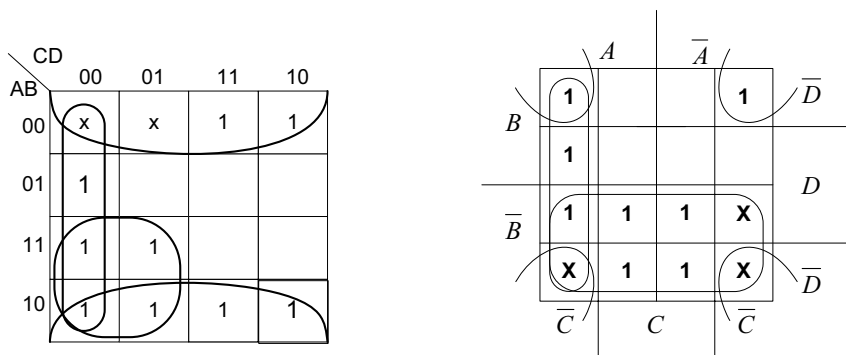
$$F_{MDF}(X, Y, Z, W) = \bar{X}W + \bar{Z}\bar{W} + XY\bar{Z}$$

Primer 27. Primenom Karnooove 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:



Primer 28. Primenom Karnooove 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:

CD \ AB	00	01	11	10
00	x			1
01	1	1	x	1
11	x			1
10	1			1

	A	$\bar{A}$	
B	x	1	$\bar{D}$
$\bar{B}$			D
	$\bar{C}$	C	$\bar{C}$
	1	1	$\bar{D}$

$$F_{MDF}(A, B, C, D) = \bar{D} + \bar{A}B$$

Primer 29. Primenom Karnoove mape odrediti minimalnu konjunktivnu 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:

CD \ AB	00	01	11	10
00		x		0
01		0		
11		x	0	0
10		0	0	0

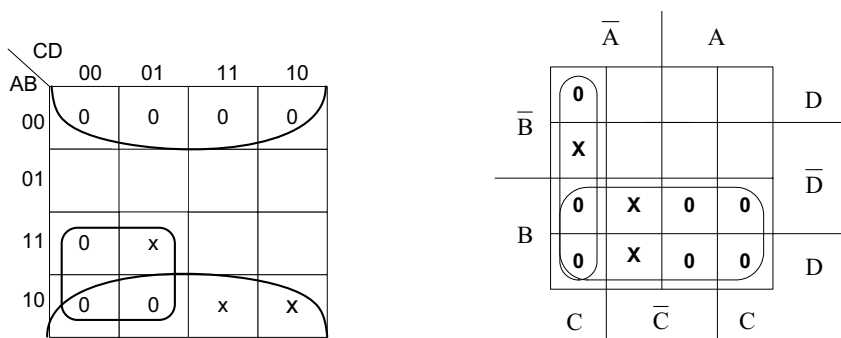
	$\bar{A}$	A	
$\bar{B}$	0		D
$\bar{D}$	x	0	0
B	0	0	x
D		0	0
	C	$\bar{C}$	C

$$F_{MKF}(A,B,C,D) = (\bar{A} + \bar{C}) \cdot (C + \bar{D}) \cdot (B + \bar{C} + D)$$

Primer 30. 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	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:

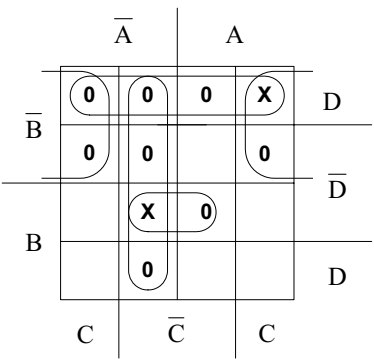


$$F_{MKF}(A, B, C, D) = (\bar{A} + C) \cdot B$$

Primer 31. Primenom Karnooove 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

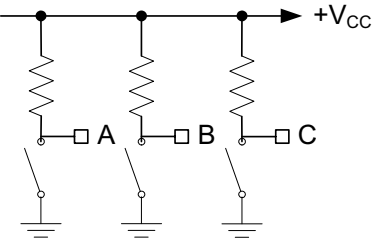
Rešenje:



$$F_{MKF}(A,B,C,D) = (\overline{A} + \overline{C}) \cdot (\overline{B} + D) \cdot (\overline{B} + C) \cdot (B + \overline{C} + \overline{D})$$

Primer 32. 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) Definirati funkciju $OUT(A, B, C)$ koja opisuje rad kombinacione mreže pomoću tabele istinitosti.
- b) Odrediti minimalnu disjunktivnu formu funkcije $OUT_{MDF}(A, B, C)$.

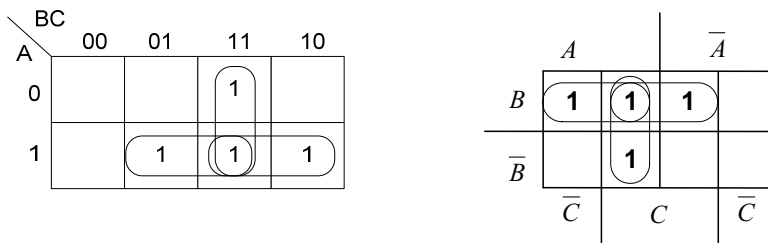


Rešenje:

- a) Pošto se na ulaz kombinacione mreže dovode tri signala A, B i C broj kombinacija logičkih vrednosti ovih signala je $2^3 = 8$.

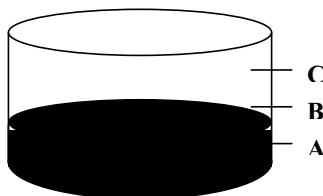
n	A	B	C	OUT
0	0	0	0	0
1	0	0	1	0
2	0	1	0	0
3	0	1	1	1
4	1	0	0	0
5	1	0	1	1
6	1	1	0	1
7	1	1	1	1

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:



- 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.
- Primenom Karnoove mape odrediti minimalnu disjunktivnu formu funkcije $F(C, B, A)$.

Rešenje:

- 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

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

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.

BA \ C	00	01	11	10
0	0	1	0	X
1	X	X	0	X

$$F_{MDF}(C, B, A) = \overline{BA}$$

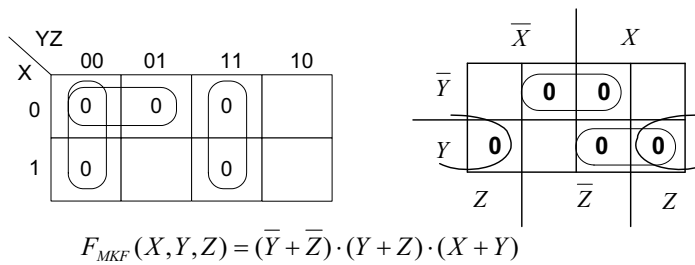
	C		$\bar{C}$	
B	X	0	0	X
$\bar{B}$	X	X	1	0
	$\bar{A}$	A		$\bar{A}$

- 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.
- Odrediti minimalnu konjunktivnu formu funkcije F primenom Karnoove mape.

a)

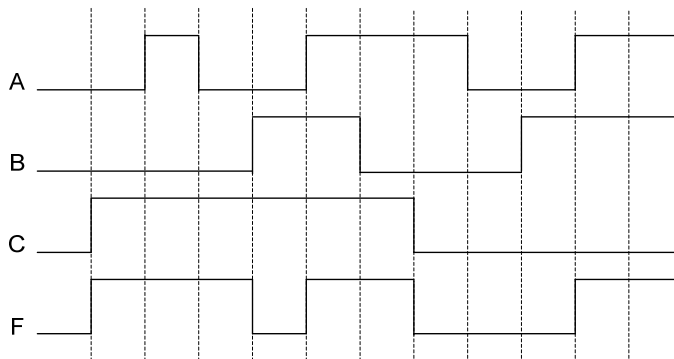
n	X	Y	Z	F
0	0	0	0	0
1	0	0	1	0
2	0	1	0	1
3	0	1	1	0
4	1	0	0	0
5	1	0	1	1
6	1	1	0	1
7	1	1	1	0

b)



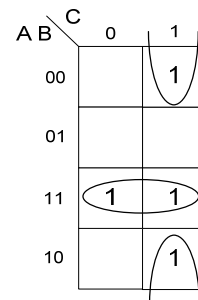
Primer 35. Na slici je prikazan vremenski dijagram promene vrednosti funkcije $F(A,B,C)$ u zavisnosti od promene ulaznih signala A , B i C . Na osnovu zadatih vremenskih dijagrama:

- Popuniti tabelu istinitosti.
- Odrediti analitički izraz za minimalnu disjunktivnu formu funkcije primenom Karnoove mape.
- 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

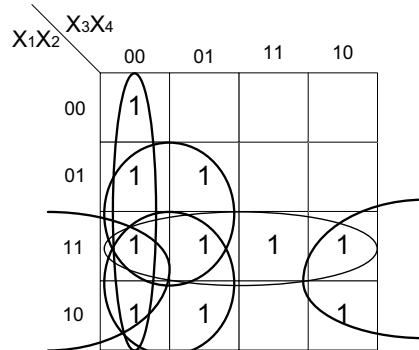


$$F(A,B,C) = A \cdot B + \bar{B} \cdot C$$

- 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



$$Y(X_1, X_2, X_3, X_4) = X_1 \cdot X_2 + \overline{X_3} \cdot \overline{X_4} + X_2 \cdot \overline{X_3} + X_1 \cdot \overline{X_3} + X_1 \cdot \overline{X_4}$$

Primer 37. Neka su A, B i C tri senzora, čiji su digitalni izlazi dovedeni na ulaz kombinacione mreže. Definirati tabelu istinitosti za mrežu čija logička funkcija $Y(A, B, C)$ ima vrednost 0:

- kada senzori A i C daju istu vrednost na izlazu
- kada senzori A i C daju različite vrednosti na izlazu
- odrediti minimalnu konjunktivnu formu (MKF) funkcije $Y(A, B, C)$ za primere a) i b).

Primer 38. Za funkciju koja je zadata kao proizvod potpunih suma (DNF)
 $F(D, C, B, A) = \Sigma(1, 2, 4, 6, 9, 10, 11, 13, 14, 15)$

- Odrediti minimalnu disjunktivnu formu funkcije (**MDF**).
- Odrediti minimalnu konjunktivnu formu funkcije (**MKF**).

Primer 39. 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 konjunktivnu formu funkcije (**MKF**).

Primer 40. 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 konjunktivnu formu funkcije (**MKF**).
- b) Odrediti minimalnu disjunktivnu formu funkcije (**MDF**).

Primer 41. 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 konjunktivnu formu funkcije (**MKF**).
- b) Odrediti minimalnu disjunktivnu formu funkcije (**MDF**).

Primer 42. 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 konjunktivnu formu funkcije (**MKF**).

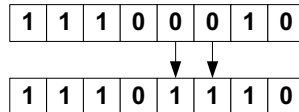
Primer 43. Za funkciju $F_{KNF}(X_1, X_2, X_3, X_4) = \Pi(0, 4, 7, 8, 12) = 0$, $F(b) = (1, 3, 9, 12, 14, 15)$,
gde je b nedefinisana vrednost funkcije:

- a) Odrediti minimalnu konjunktivnu formu funkcije (**MKF**).
- b) Odrediti minimalnu disjunktivnu formu funkcije (**MDF**).

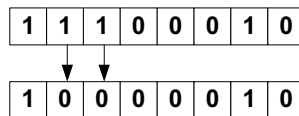
7. PRIMENA UNARNIH I BINARNIH LOGIČKIH OPERACIJA

Unarne operacije su instrukcije sa **jednim** operandom. Osnovni tipovi unarnih operacija su:

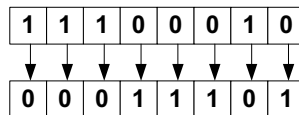
- **Postavljanje (SET)** - logička vrednost **0** nekog **bita** pretvara se u logičku vrednost **1**. Primenjuje se na bite u registru stanja, registrima CPU i memorijskim lokacijama.



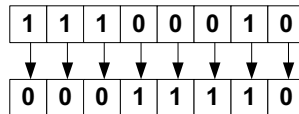
- **Brisanje (CLEAR, RESET)** - logička vrednost **1** željenog **bita** se pretvara u logičku vrednost **0**. Primenjuje se na bite u registru stanja, registrima CPU i memorijskim lokacijama.



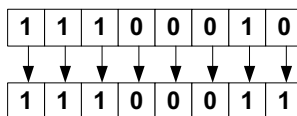
- **Komplementiranje (COMPLEMENT)** - sve logičke **0** u jednom broju se pretvaraju u **1** i obrnuto, tako da se dobija **prvi komplement**. Operacija se odnosi na sadržaj celog registra.



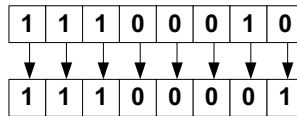
- **Negacija (NEGATE)** – kao rezultat se dobija drugi komplement broja. Operacija se odnosi na sadržaj celog registra.



- **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.



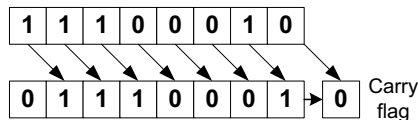
- **Dekrementiranje (DECREMENT)** - smanjuje vrednosti operanda za 1. Odnosi se na sadržaj celog registra. Kod nekih procesora se primenjuje samo na registre u okviru CPU, a kod drugih se primenjuje i na sadržaj memorijskih lokacija.



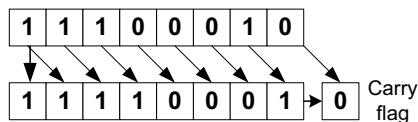
- **Pomeranje (SHIFT)** - pomeraju se biti u okviru jednog registra ulevo ili udesno. Bit koji napušta registar se gubi ili se upisuje u Carry flag. Kod mikroprocesora se obično u jednom taktu obavi pomeranje za jednu poziciju, a kod većih računara se u jednom taktu pomeri sadržaj za više pozicija. Upražnjena mesta se obično popunjavaju logičkim nulama (0).

- **Pomeranje udesno**

Primena: Ispitivanje sadržaja bita u registru
Deljenje sa 2^N , N = broj pomeranja

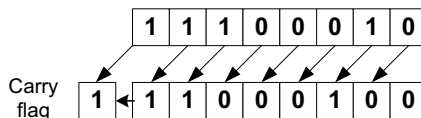


Aritmetičko pomeranje udesno - bit znaka ostaje na MSB poziciji, a pomeraju se svi ostali biti.



- **Pomeranje ulevo**

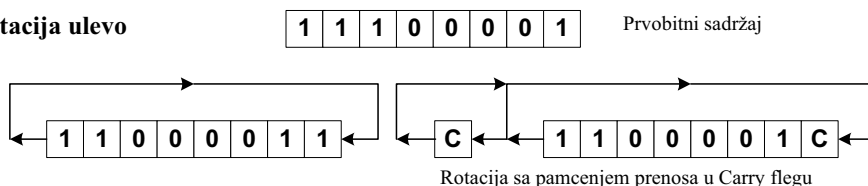
Primena: Ispitivanje sadržaja bita u registru
Množenje sa 2^N , N = broj pomeranja



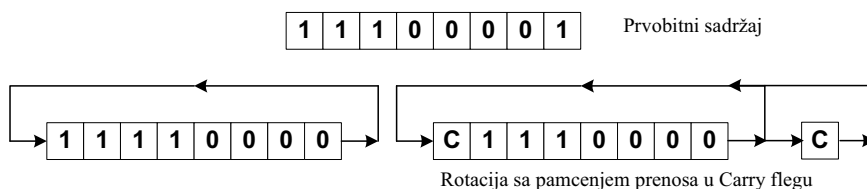
- **Rotacija (ROTATE) - rotiraju se biti u okviru jednog registra ulevo ili udesno.**

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 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 8-bitnom registru A nalazi se broj $7E_{(16)}$. Koji se HEX broj nalazi u registru posle:

- pomeranja prvobitnog sadržaja za 3 mesta ulevo
- rotacije prvobitnog sadržaja za 3 mesta ulevo
- pomeranja prvobitnog sadržaja za 3 mesta udesno
- rotacije prvobitnog sadržaja za 3 mesta udesno.

Rešenje:

Prvobitni sadržaj u registru A: $01111110_{(2)}$

- Posle pomeranja za 3 mesta ulevo, sadržaj u registru A je: **$11110000_{(2)}$**
- Posle rotacije za 3 mesta ulevo, sadržaj u registru A je: **$11110011_{(2)}$**
- Posle pomeranja za 3 mesta udesno, sadržaj u registru A je: **$00001111_{(2)}$**
- Posle rotacije za 3 mesta udesno, sadržaj u registru A je: **$11001111_{(2)}$**

Primer 2 Iz tabele sa **ASCII** kodovima pročitati kod za broj $9_{(10)}$ i sadržaj sačuvati u 8-bitnom registru A. Pokazati kako primenom operacija pomeranja i rotiranja za odgovarajući broj mesta možemo da postignemo da:

- viši nibl u registru A bude 0
- niži nibl u registru A bude 0
- viši nibl dođe na mesto nižeg nibla
- niži nibl dođe na mesto višeg nibla

Rešenje:

Prvobitni sadržaj u registru A je $00111001_{(2)} = 39_{(16)}$

- Posle pomeranja za 4 mesta udesno, sadržaj u registru A je **$00000011_{(2)} = 03_{(16)}$**

- b) Posle pomeranja za 4 mesta ulevo, sadržaj u registru A je $10010000_{(2)} = 90_{(16)}$
- c) Posle rotacije za 4 mesta udesno, sadržaj u registru A je $10010011_{(2)} = 93_{(16)}$
- d) Posle rotacije za 4 mesta ulevo, sadržaj u registru A je $10010011_{(2)} = 93_{(16)}$

Primer 3 Iz tabele sa **EBCDIC** kodovima pročitati kod za broj $6_{(10)}$ i sadržaj sačuvati u 8-bitnom 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 $00111101_{(2)} = 3D_{(16)}$
 - b) Posle pomeranja za 4 mesta ulevo, sadržaj u registru A je $01100000_{(2)} = 60_{(16)}$
 - c) Posle rotacije za 4 mesta udesno, sadržaj u registru A je $01101111_{(2)} = 6F_{(16)}$
 - d) Posle rotacije za 4 mesta ulevo, sadržaj u registru A je $01101111_{(2)} = 6F_{(16)}$
-

Primer 4 U registru A se nalazi HEX broj $B3_{(16)}$. Šta će se 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 $01111011_{(2)}$
 - b) Posle rotacije za 2 mesta ulevo, ako je $C = 1$ sadržaj u registru A je $11001111_{(2)}$
 - c) Posle rotacije za 4 mesta udesno, ako je $C = 0$ sadržaj u registru A je $01101011_{(2)}$
 - d) Posle rotacije za 2 mesta ulevo, sadržaj u registru A je $11001101_{(2)}$
-

Primer 5 Sadržaj akumulatora predstavljaju slova **Lt** (ASCII kod). Napraviti masku za komplementiranje parnih bitova.

Rešenje: **0001100100100001**

Primer 6 Sadržaj akumulatora predstavljaju slova **Cp** (ASCII kod). Napraviti masku za izdvajanje po tri krajnja bita.

Rešenje: **1100000000000111**

Primer 7 Sadržaj akumulatora predstavljaju slova **sP** (ASCII kod). Napraviti masku za izdvajanje 1.,3.,7.,9.,12. i 14. bita.

Rešenje: **0101001000000000**

Primer 8 Predstaviti u EBCDIC kodu slova **NM** kao sadržaj akumulatora A. Izvršiti uključivanje prva tri i poslednja tri prekidača, koji su direktno priključeni na akumulator A bez promene ostalih prekidača.

Rešenje: **1110010111010111**

Primer 9 U registru A se nalazi binarni zapis broja 6.

- Koji dekadni broj se nalazi u registru posle 5 pomeranja binarnog sadržaja ulevo?
- Koja aritmetička operacija je realizovana na ovaj način?

Rešenje:

Prvobitni sadržaj u registru A je $00000110_{(2)}$

- Posle pomeranja za 5 mesta ulevo sadržaj registra A je **$11000000_{(2)} = 192_{(10)}$**
 - Na ovaj način je izvršeno množenje broja 5 sa **$2^5 = 32$**
-

Primer 10 U 8-bitnom registru A nalazi se ASCII kod broja 9, a u 8-bitnom registru B se nalazi se ASCII kod broja 7.

- Primenom odgovarajuće logičke operacije i maske pretvoriti ASCII kod brojeva u registru A i B u raspakovani format BCD koda "8421".
- 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).
- 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 $00110111_{(2)}$

- 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":
 $A = 00111001_{(2)} \text{ AND } 00001111_{(2)} = \mathbf{00001001_{(2)}}$
 $B = 00110111_{(2)} \text{ AND } 00001111_{(2)} = \mathbf{00000111_{(2)}}$
 - 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 $A = A \text{ OR } B = \mathbf{01111001_{(2)}} = \mathbf{79_{(16)}}$
 - 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 $B = A \text{ OR } B = \mathbf{01111001_{(2)}} = \mathbf{79_{(16)}}$
-

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

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. 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. 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:

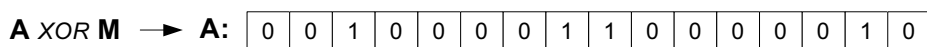
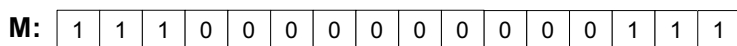
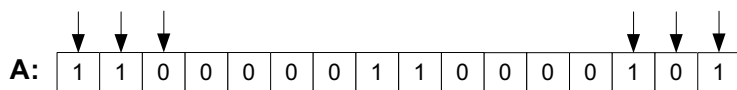
0	0	0	0	1	1	0	0	1	0	0	1	0	0	1	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

A XOR M → A:

1	0	1	1	1	0	1	0	0	1	1	1	1	0	1	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

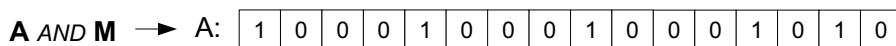
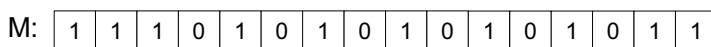
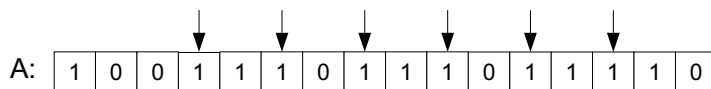
Primer 14 U akumulatoru A nalazi se sledeći 16-bitni podatak: 1100000110000101. Primenom odgovarajuće maske i logičke operacije **komplementirati** po tri krajnja stanja u ovom registru. Nakon izvršenja logičke operacije rezultat se nalazi u akumulatoru A.

Rešenje:



Primer 15 U akumulatoru A nalazi se sledeći podatak A: 1001110111011110, 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:



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

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
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Primer 17 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: 10101010101010. 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
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Primer 18 U akumulatoru A nalazi se sledeći podatak A : 1100100101001001, 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
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Primer 19 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

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

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:

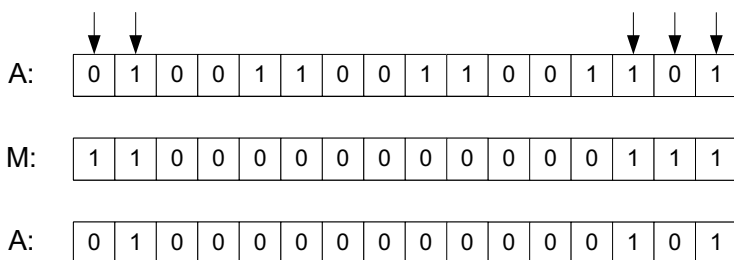
0	0	0	0	0	1	1	0	0	0	1	0	1	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

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. 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:



Primer 24. U registru A se nalazi binarni zapis broja 72.

- Koji dekadni broj se nalazi u registru posle 2 pomeranja binarnog sadržaja ulevo?
- Koja aritmetička operacija je realizovana na ovaj način?

Primer 25. U registru A se nalazi binarni zapis broja 9.

- Koji dekadni broj se nalazi u registru posle 4 pomeranja binarnog sadržaja ulevo?
- Koja aritmetička operacija je realizovana na ovaj način?

Primer 26. U registru A se nalazi binarni zapis broja 40.

- Kako primenom operacije pomeranja u odgovarajućem smeru broj može da se podeli sa 8?
- Šta se nalazi u registru nakon poslednje operacije pomeranja?

Primer 27. U registru A se nalazi binarni zapis broja 16.

- Kako primenom operacije pomeranja u odgovarajućem smeru ovaj broj može da se pomnoži sa 16?
- Šta se nalazi u registru nakon poslednje operacije pomeranja?

Primer 28. U 16-bitnom registru A nalazi se binarni broj 0011000011101011. Primenom odgovarajuće maske i logičke operacije **invertovati**:

- 1.,2.,5.,8.,10.,12. i 14. bit;
- niži bajt i najviši bit;
- po četiri krajnja bita;
- svaki neparni bit;
- 4.,5.,8.,9.,11.,13. i 15. bit.

Primer 29. U 16-bitnom registru A nalazi se binarni broj 0011010100001111. Primenom odgovarajuće maske i logičke operacije **isključiti**:

- 0.,2.,7.,10.,11.,12. i 15. bit;
- sve parne bitove;
- sve neparne bitove;
- viši bajt;
- 3.,4., 8.,9.,12. i 13. bit.

- Primer 30. U 16-bitnom registru A nalazi se binarni broj 1111000010101010. 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 31. U 16-bitnom registru A nalazi se binarni broj 1100110101011100. 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 32. U 16-bitnom registru A nalazi se binarni broj 1111100111101111. Primenom odgovarajuće maske i logičke operacije:
- a) i) 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 33. U 16-bitnom registru A nalazi se binarni broj 1000001100101000. Primenom odgovarajuće maske i logičke operacije:
- a) setovati najniža 3 bita
 - b) setovati 2., 3., 4., 5., 11., 12. i 14. bit
- Primer 34. U 16-bitnom registru A nalazi se binarni broj 1001100111101001. 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 35. 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 36. 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 37. 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 38. U 8 – bitnom registru A se nalazi HEX broj 4, a u registru B je HEX broj 5.

- a) Primenom odgovarajuće 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.

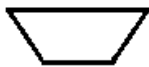
Definicija:

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.

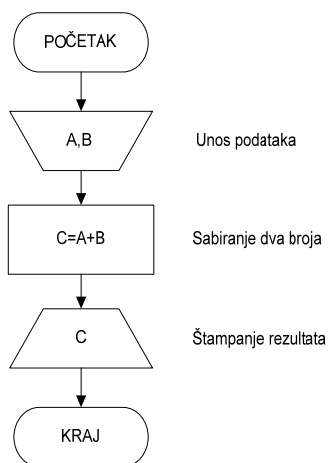
Međutim, ako se namučimo i osmislimo suštinski drugačiji algoritam (V) za uređivanje, on stvarno može biti reda složenosti $n \cdot \log(n)$. Algoritmi (A) i (B) su iste složenosti, jer se u notaciji sa velikim O obeležavaju sa $O(n^2)$, a u govoru se zovu 'algoritmi kvadratne složenosti', dok je algoritam (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*, *until*, *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.

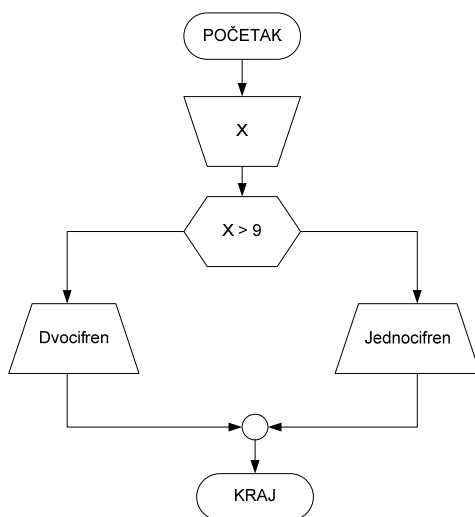
Primer 1. Napraviti algoritam za sabiranje dva broja.

Rešenje:



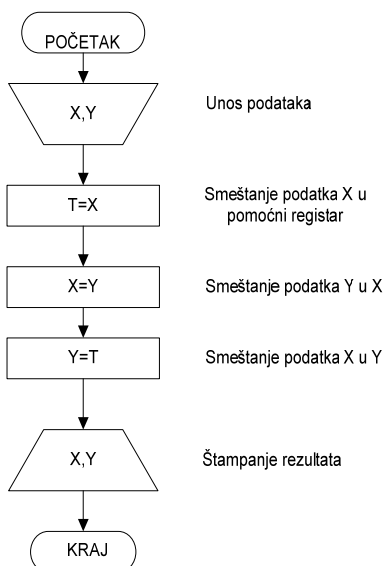
Primer 2. Napisati algoritam koji za uneti prirodni broj do 100 ispisuje da li je broj dvocifren ili jednocifren.

Rešenje:



Primer 3. Napraviti algoritam koji učitava brojeve X i Y i vrši zamenu njihovih vrednosti.

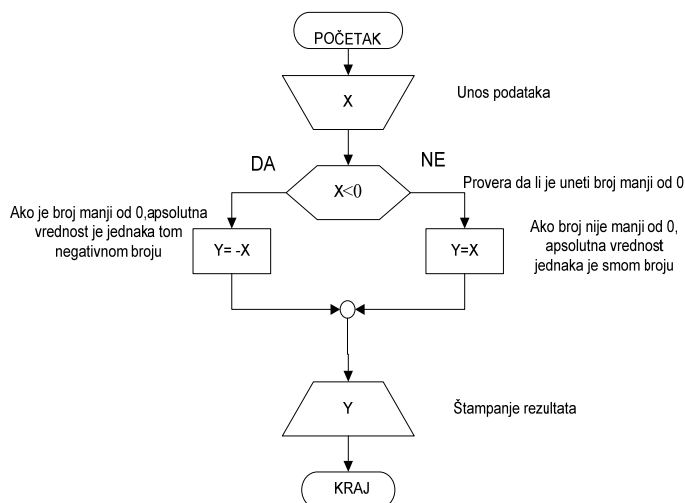
Rešenje:



Primer 4. Napraviti algoritam koji određuje apsolutnu vrednost broja X.

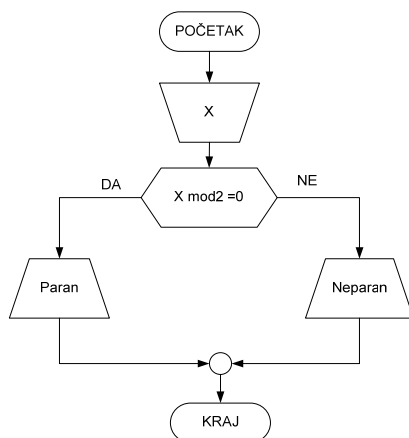
$$y = |x| = \begin{cases} -x; & x < 0 \\ x; & x \geq 0 \end{cases}$$

Rešenje:



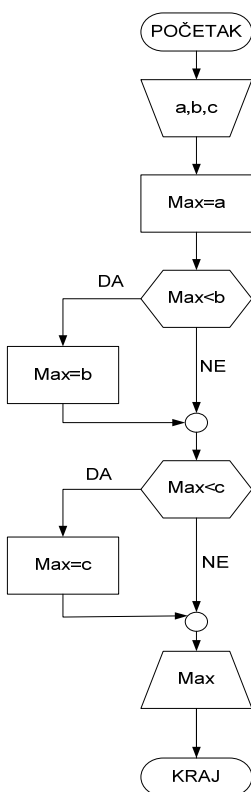
Primer 5. Napisati algoritam koji ispituje da li je broj paran ili neparan

Rešenje:



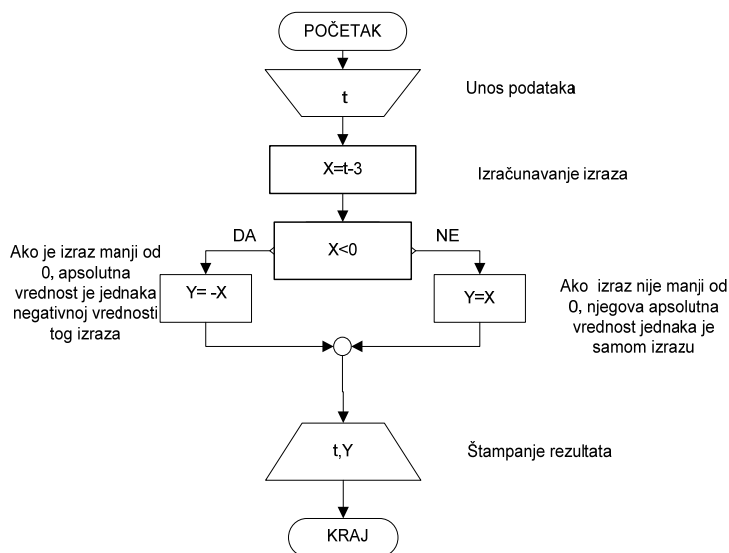
Primer 6. Napraviti algoritam koji promenljivoj max dodeljuje najveću vrednost tri zadata broja a, b, c

Rešenje:



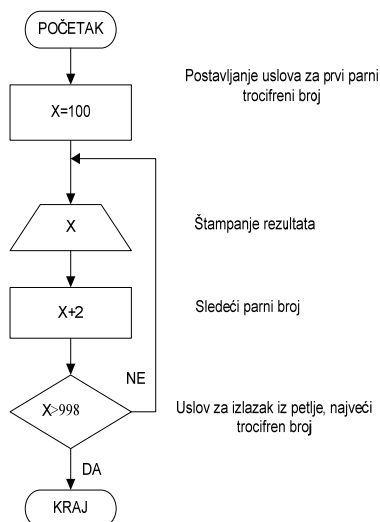
Primer 7. Napraviti algoritam koji izračunava izraz $|t-3|$ i štampa uneti podatak t i apsolutnu vrednost izraza .

Rešenje:



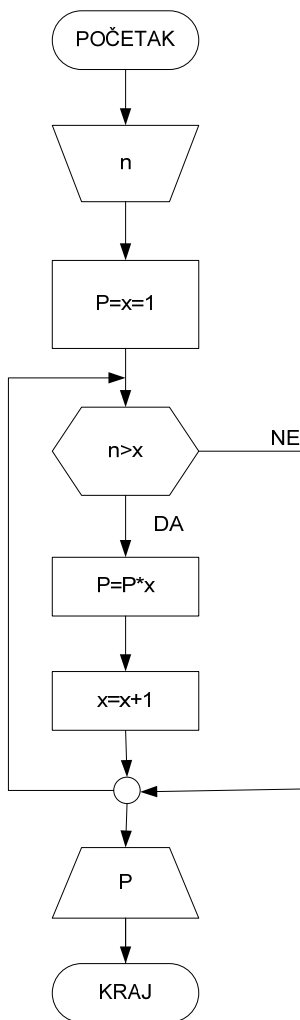
Primer 8. Napraviti algoritam koji štampa sve parne trocifrene brojeve.

Rešenje:



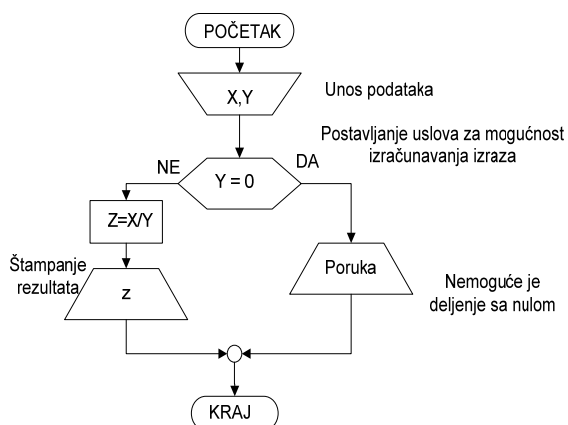
Primer 9. Napisati algoritam za izračunavanje faktoriala.

Rešenje:



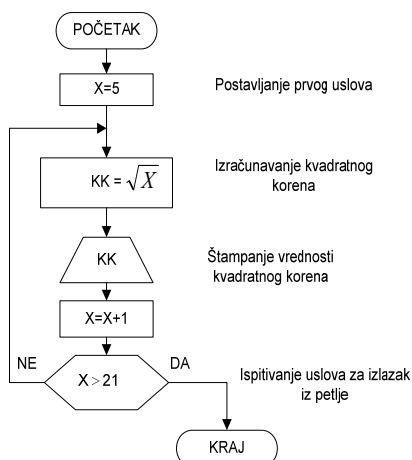
Primer 10. Napraviti algoritam za unošenje dva prirodna broja (x,y) i izračunavanje izraza $z = x/y$. Štampati vrednost izraza, z .

Rešenje:



Primer 11. Napraviti algoritam koji izračunava i štampa kvadratni koren brojeva od 5 do 21.

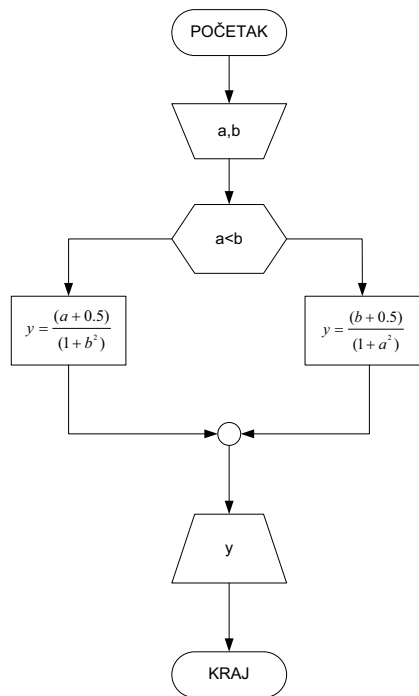
Rešenje:



Primer 12. Sastaviti algoritam kojim se za zadate realne brojeve a i b, izračunava funkcija y prema formuli:

$$y = \frac{\min(a, b) + 0,5}{1 + \max^2(a, b)}$$

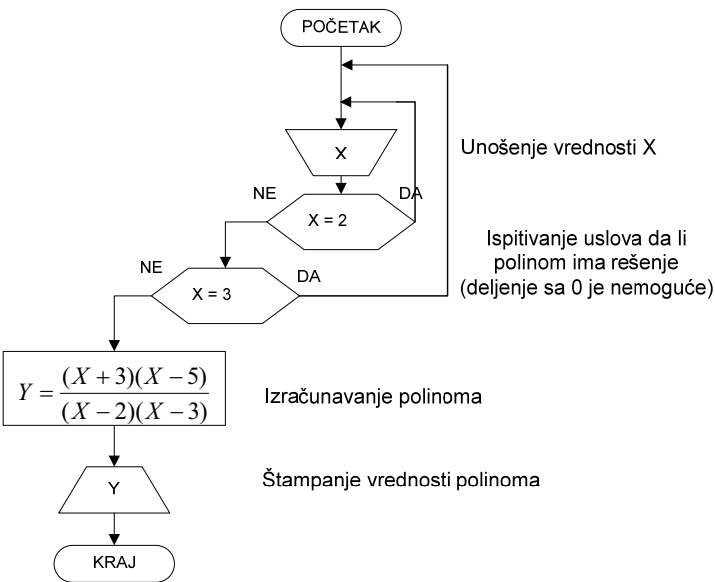
Rešenje:



Primer 13. Napraviti algoritam koji učitava vrednost x, izračunava i štampa vrednost polinoma y .

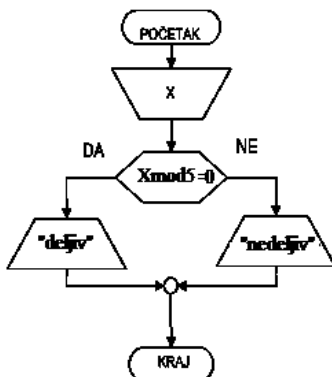
$$Y = \frac{(X + 3)(X - 5)}{(X - 2)(X - 3)}$$

Rešenje:



Primer 14. Napisati algoritam koji proverava da li je broj deljiv sa 5.

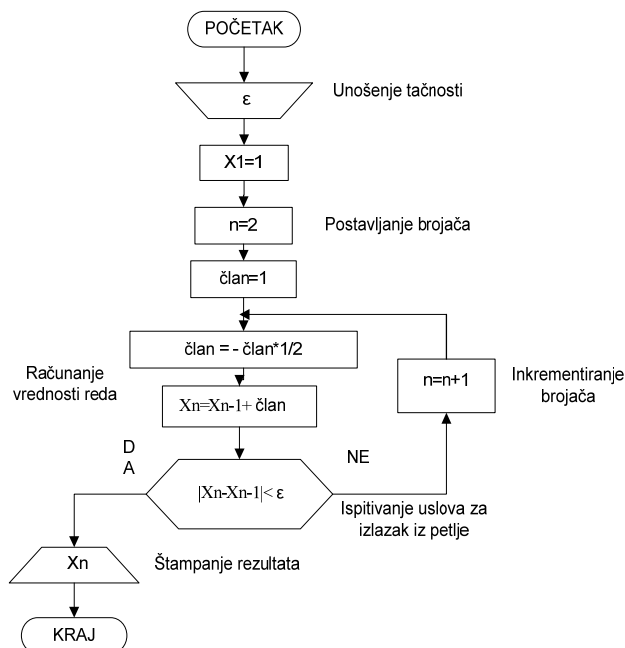
Rešenje:



Primer 15. Napraviti algoritam za izračunavanje i štampanje reda $1 - \frac{1}{2} + \frac{1}{4} - \frac{1}{8} \dots$ sa tačnošću ϵ .

Rešenje:

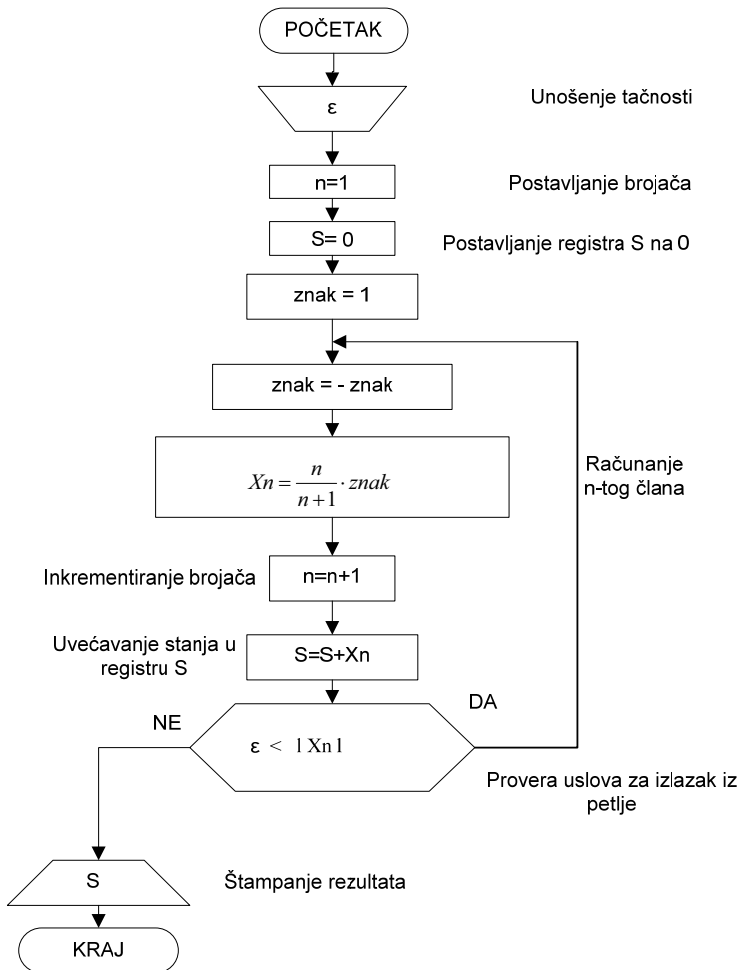
$$\begin{aligned}
 X_1 &= 1 \\
 X_2 &= X_1 - 1/2^1 \\
 X_3 &= X_2 + 1/2^2 \\
 X_4 &= X_3 - 1/2^3 \\
 X_n &= X_{n-1} + ((-1)^{n+1} * 1/2^{n1})
 \end{aligned}$$



Primer 16. Napraviti algoritam za izračunavanje i štampanje reda $-\frac{1}{2} + \frac{2}{3} - \frac{3}{4} + \frac{4}{5} - \frac{5}{6} + \frac{7}{6} \dots$ sa tačnošću ε .

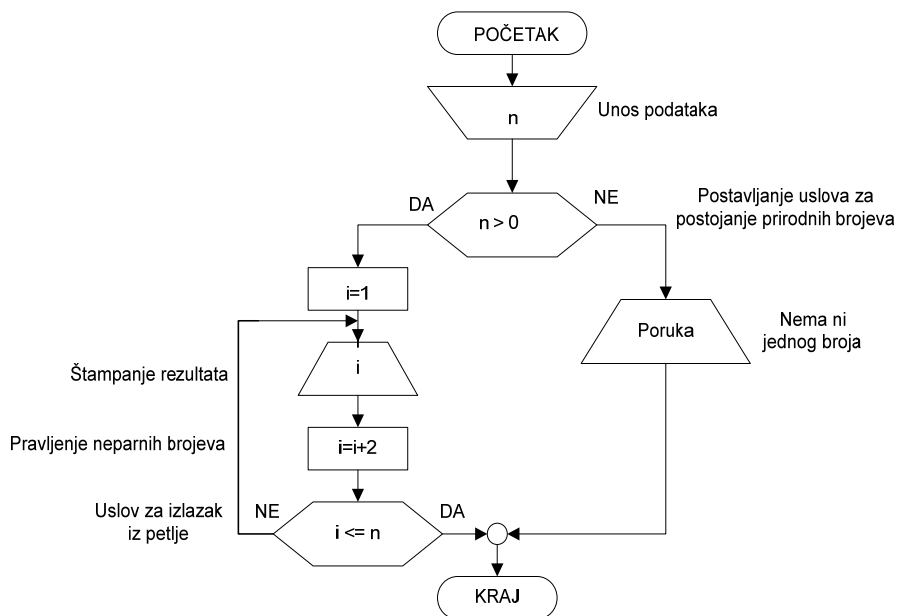
Rešenje: Opšti član je:

$$X_n = \frac{n}{n+1} \cdot (-1)^n$$



Primer 17. Napraviti algoritam koji štampa sve neparne prirodne brojeve koji su manji i jednaki n.

Rešenje:

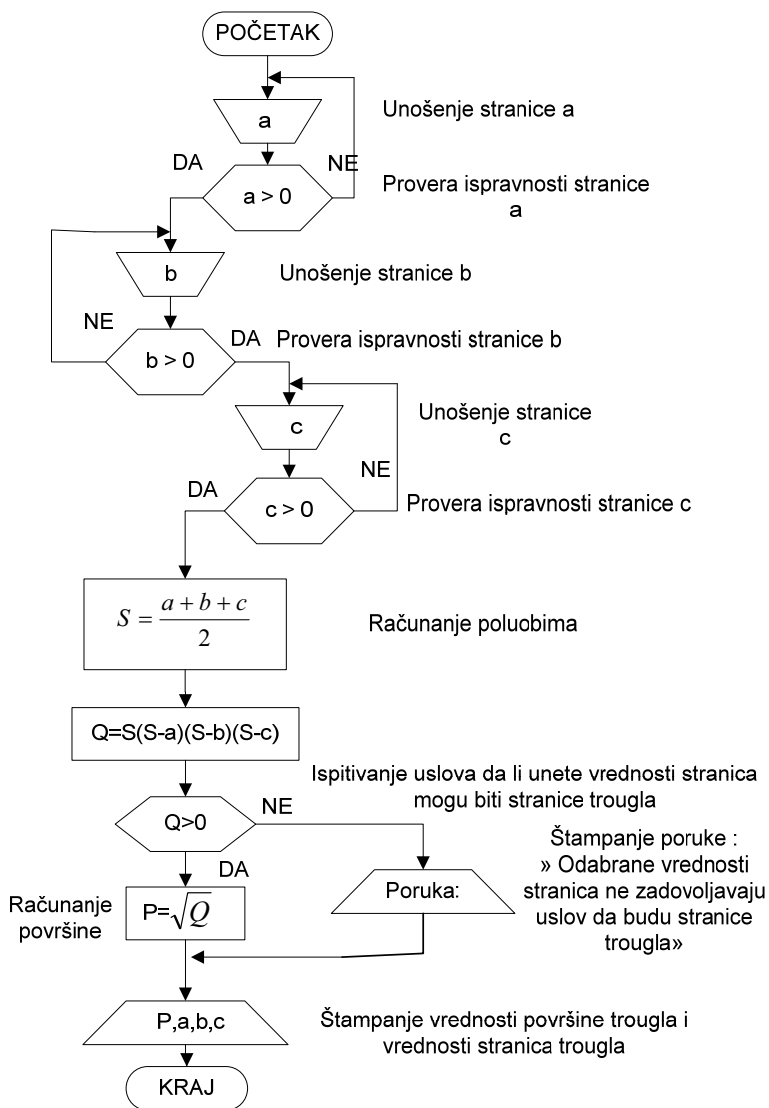


Primer 18. Napraviti algoritam koji učitava stranice trougla a , b , c i izračunava površinu trougla po Heronovom obrascu i štampa vrednost površine trougla i stranice tog trougla.

Rešenje:

$$P = \sqrt{S(S-a)(S-b)(S-c)} \quad \text{Heronov obrazac.}$$

$$S = \frac{a+b+c}{2} \quad \text{Poluobim.}$$

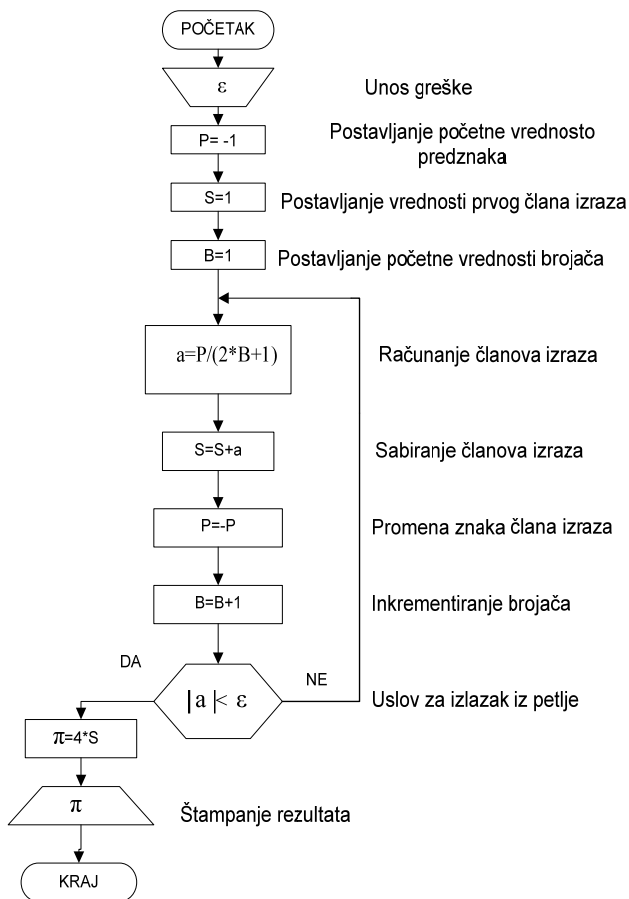


Primer 19. Napraviti algoritam koji izračunava i štampa vrednost broja π primenom izraza:

$$\frac{\pi}{4} = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \dots$$

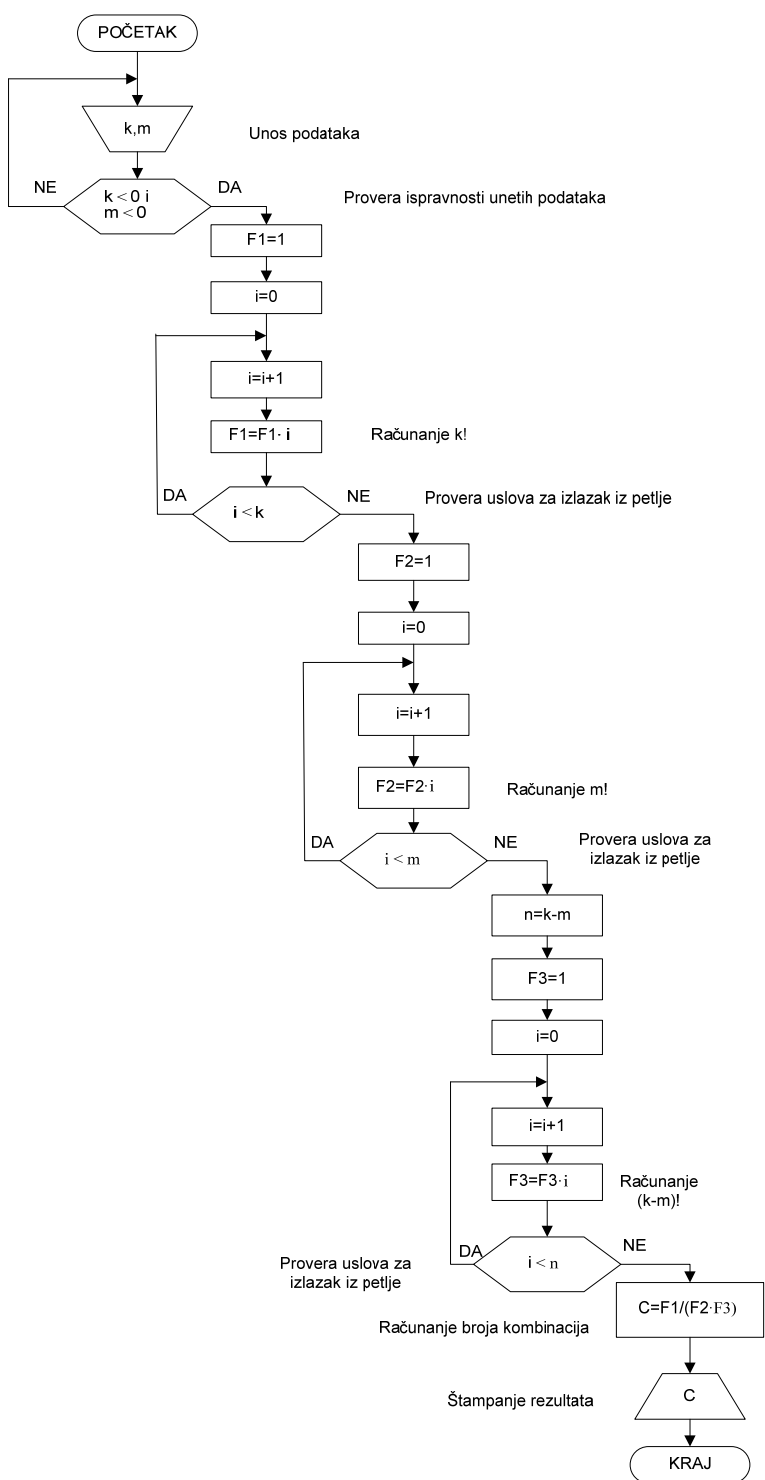
Uslov izračunavanja je da je n -ti član reda $|a_n| < \varepsilon$.

Rešenje:



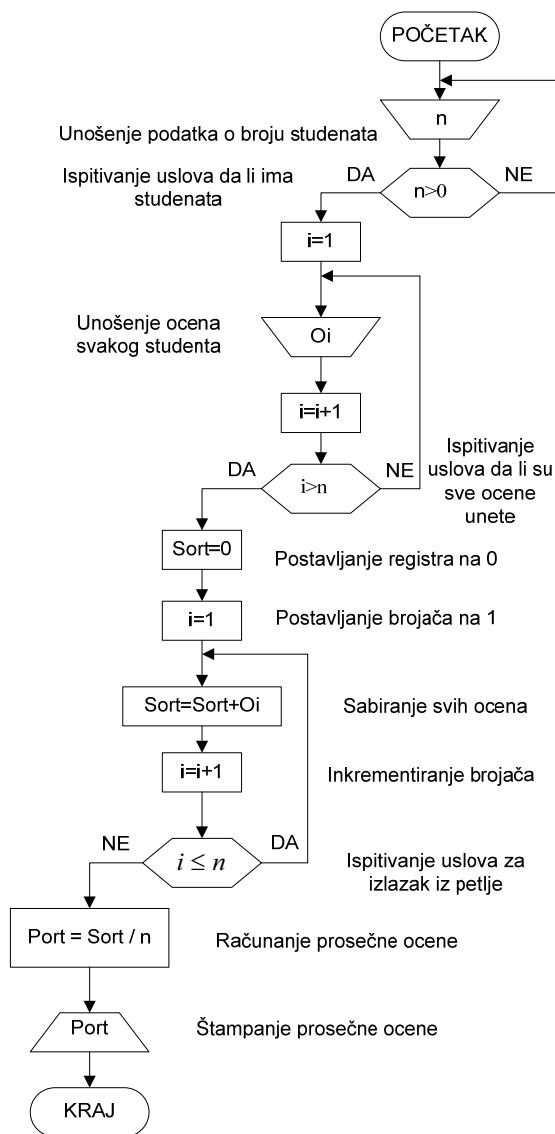
Primer 20. Zadati su dva prirodna broja k i m ($k < m$). Sastaviti algoritam koji određuje broj kombinacija k -te klase od m elemenata.

Rešenje: Broj kombinacija k -te klase od m elemenata:
$$C = \frac{k!}{m!(k-m)!}$$



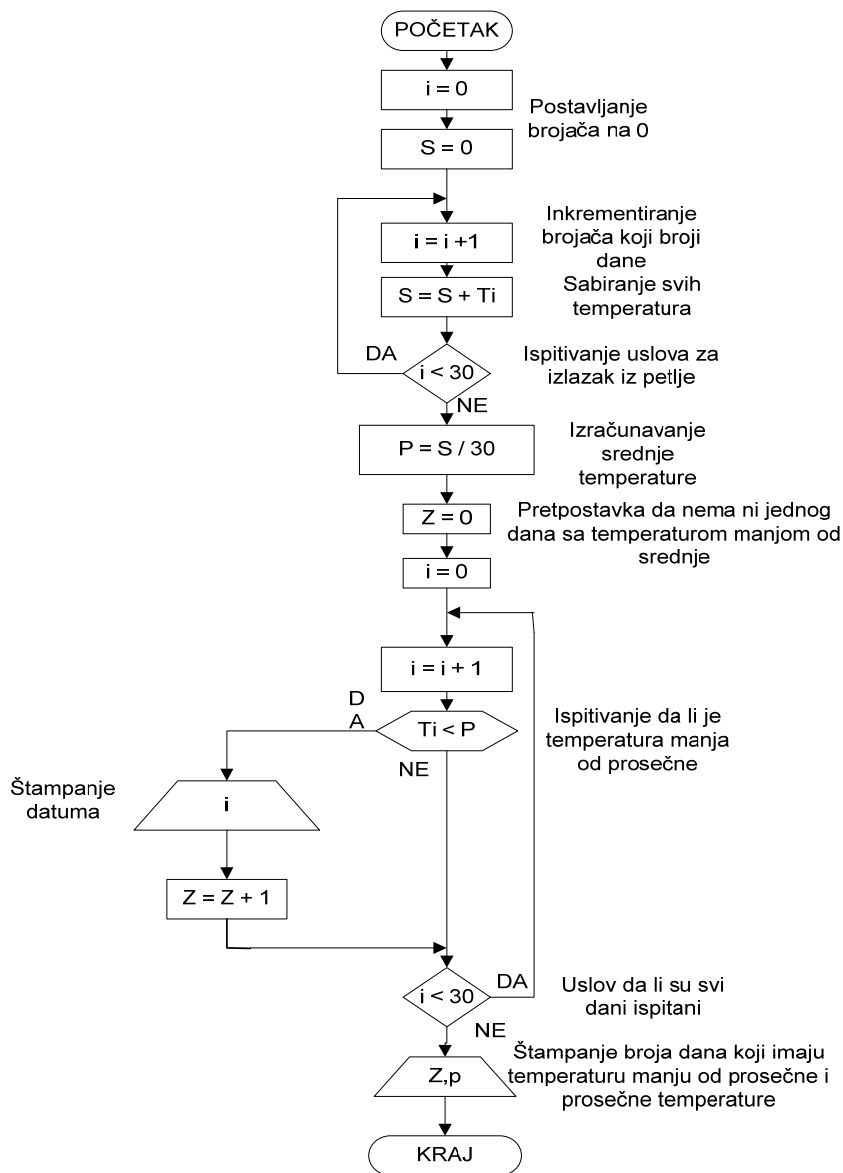
Primer 21. Napraviti algoritam koji izračunava prosečnu srednju ocenu na ispitu iz ORT-a.

Rešenje:



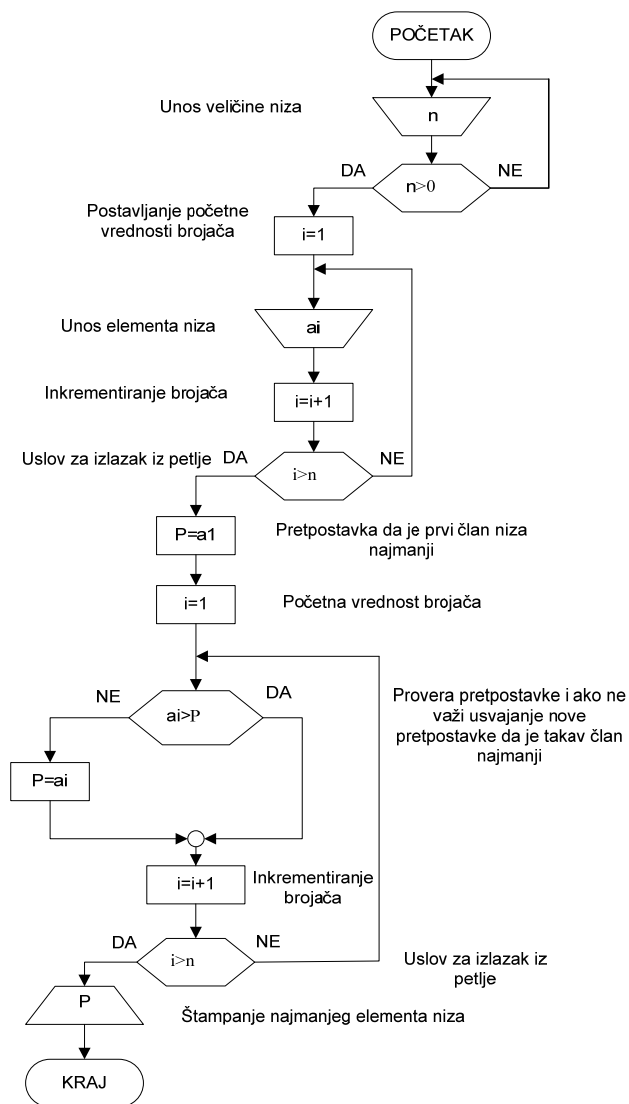
Primer 22. U datoteci se nalazi 30 brojeva koji predstavljaju temperaturu od 1–30 juna 2004. godine (T_1 - T_{30}). Sastaviti algoritam koji izračunava prosečnu temperaturu u mesecu junu, štampa datum dana u kojima je temperatura bila manja od prosečne i broj dana kada je temperatura bila manja od prosečne.

Rešenje:



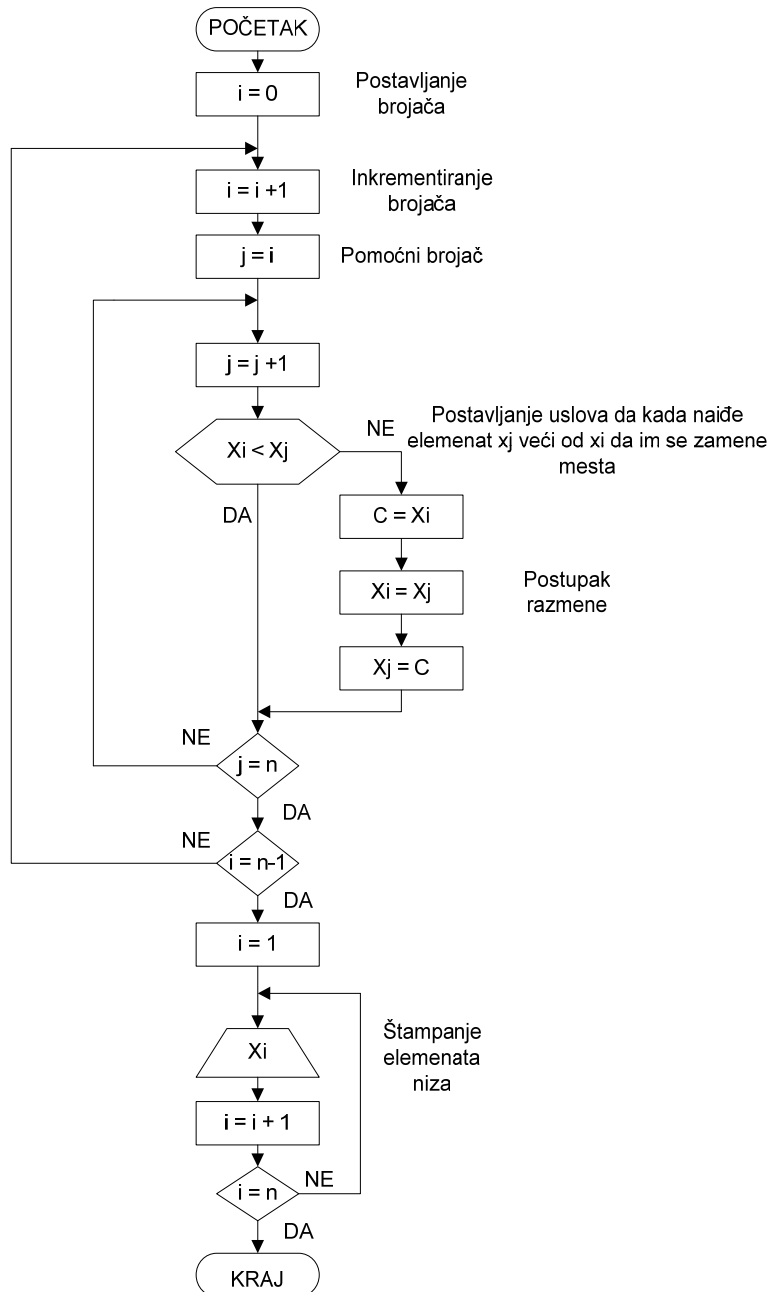
Primer 23. Napraviti algoritam koji učitava niz a_i , pronalazi i štampa najmanji elemenat niza.

Rešenje:



Primer 24. Napraviti algoritam koji dati niz dimenzije n , uređuje u rastućem redosledu.

Rešenje:



Primer 25. Napraviti algoritam koji određuje rešenja kvadratne jednačine $ax^2+bx+c=0$.

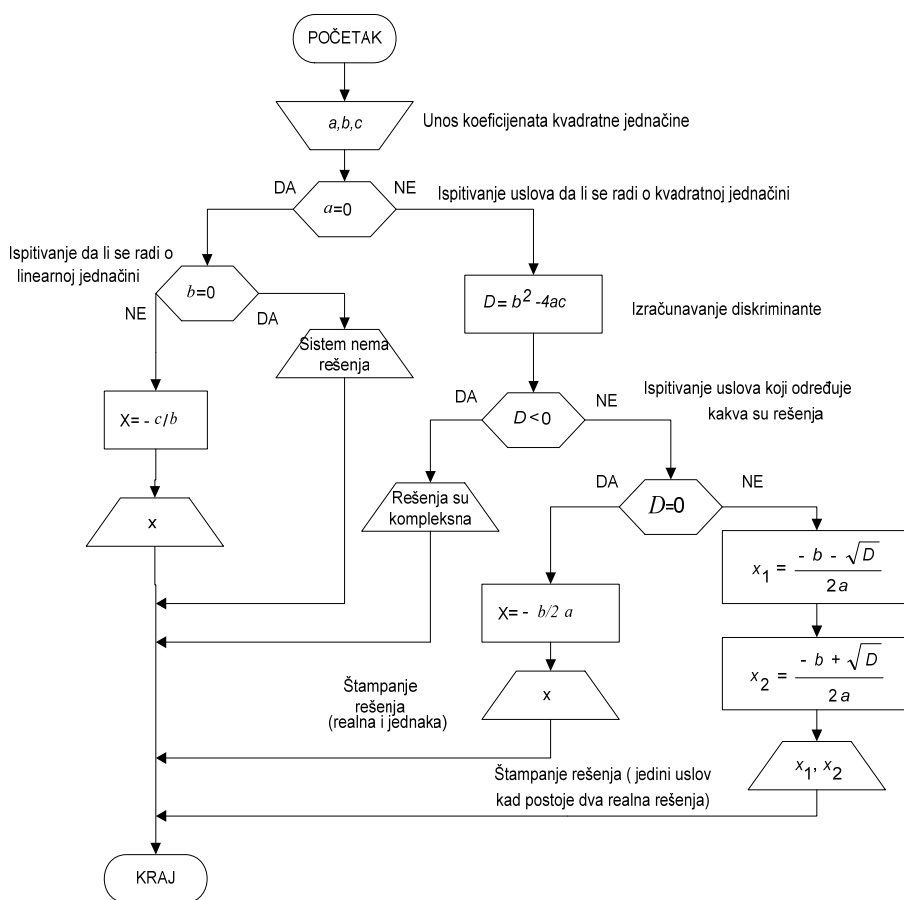
Rešenje:

$a \neq 0$, uslov da je jednačina kvadratna; Diskriminanta: $D=b^2-4ac$

Realna i različita rešenja: $x_1 = \frac{-b - \sqrt{D}}{2a}$ i $x_2 = \frac{-b + \sqrt{D}}{2a}$ ($a, b, c \neq 0$)

Realna i jednaka rešenja: $x = \frac{-b}{2a}$ ($a, b, c \neq 0$ i $b^2=4ac$).

Rešenja su konjugovano kompleksna ($b^2-4ac < 0$). Jednačina nije kvadratna, $x=-c/b$ ($a=0$ i $b \neq 0$).



Primer 26. Sastaviti algoritamsku šemu za rešavanje i štampanje rešenja sistema linearnih jednačina $ax + by = p$ i $mx + ny = q$

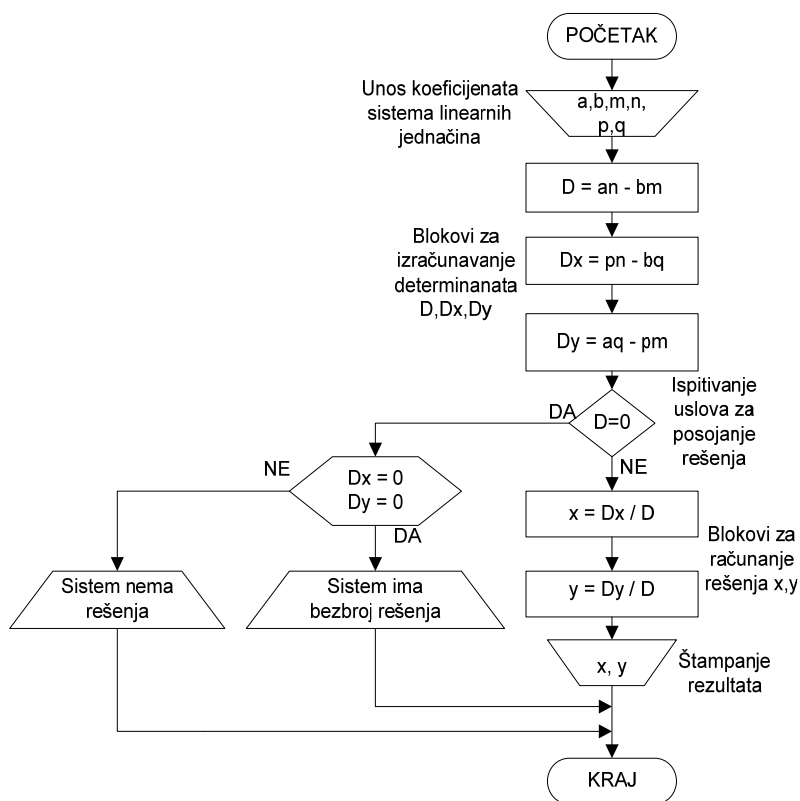
Rešenje:

$$ax + by = p, \quad mx + ny = q,$$

$$x = \frac{D_x}{D}, \quad y = \frac{D_y}{D}$$

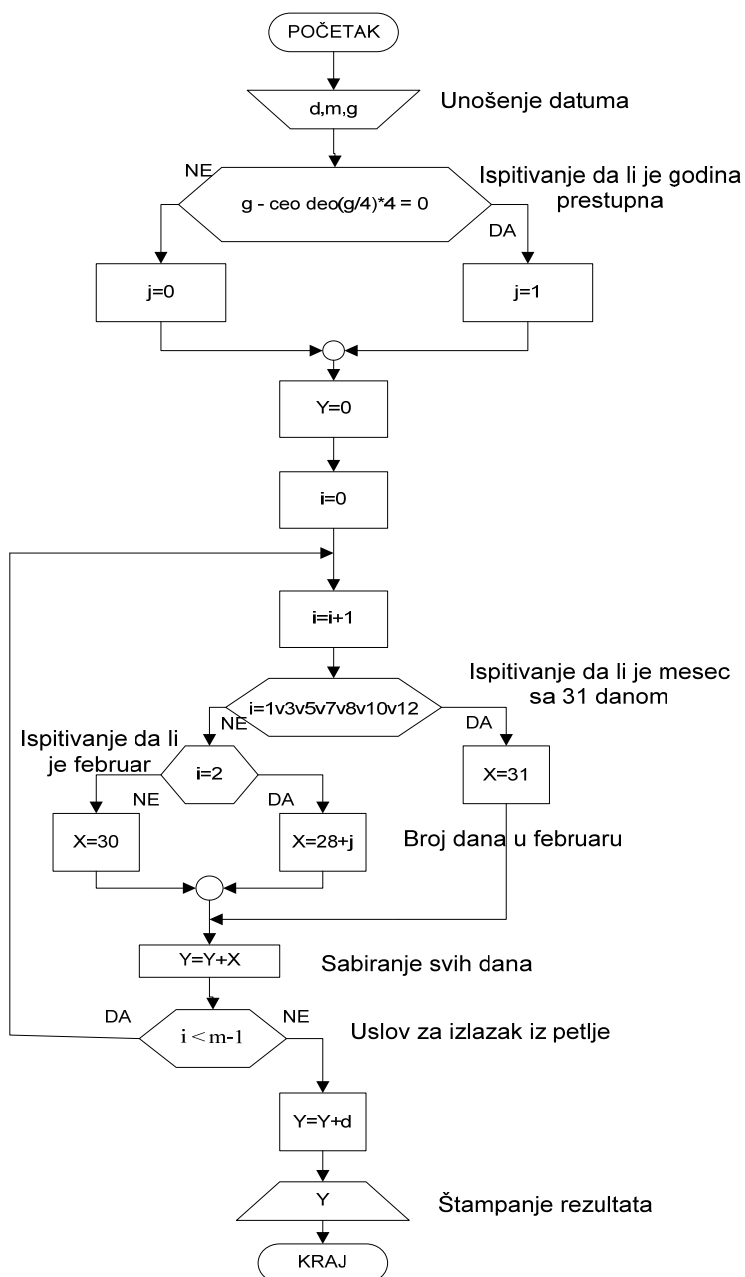
$$D = \begin{pmatrix} a & b \\ m & n \end{pmatrix} = an - bm \quad D_x = \begin{pmatrix} p & b \\ q & n \end{pmatrix} = pn - bq \quad D_y = \begin{pmatrix} a & p \\ m & q \end{pmatrix} = aq - pm$$

$D=0$; $D_x \neq 0$; $D_y \neq 0$ sistem nema rešenje. $D=0$; $D_x = 0$ i $D_y = 0$ sistem ima bezbroj rešenja



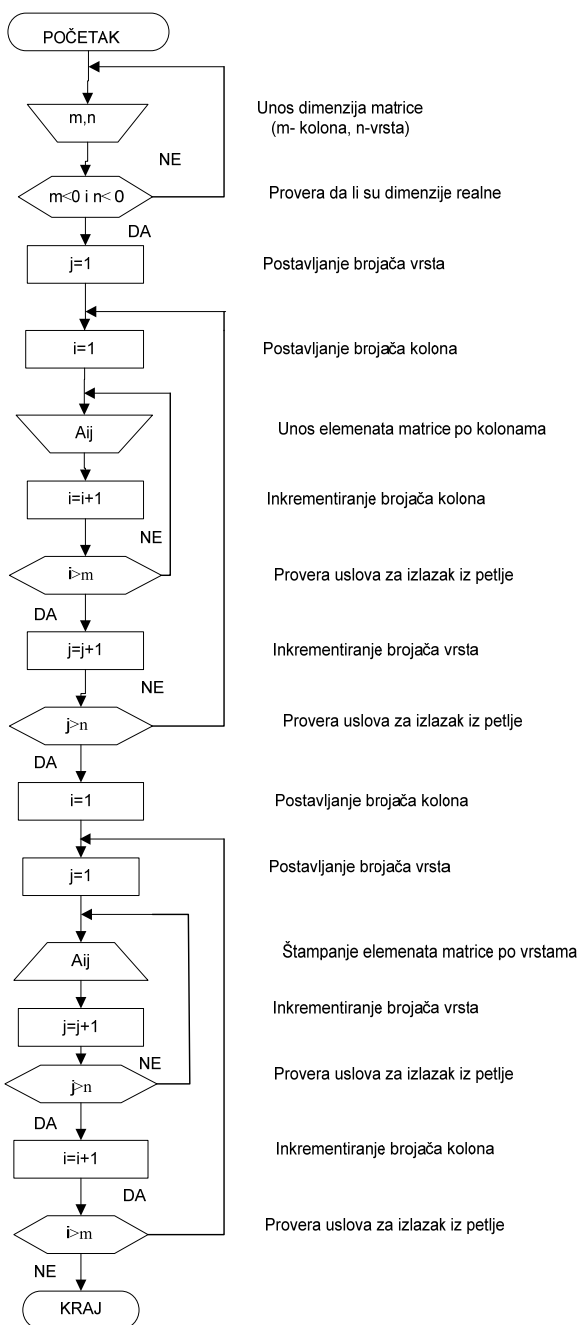
Primer 27. Napraviti algoritam koji učitava datum u obliku d,m,g ; što znači dan, mesec, godina, a zatim izračunava i štampa koji je to dan u godini.

Rešenje:



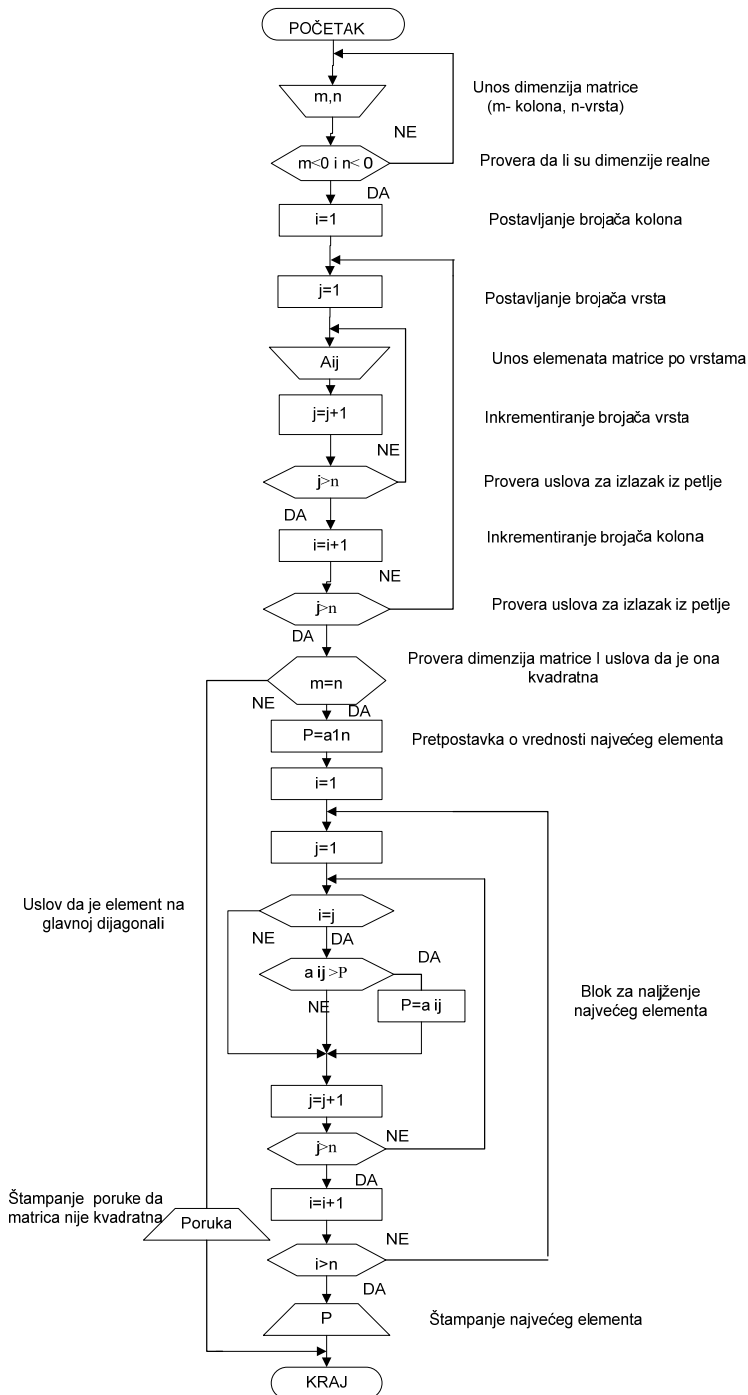
Primer 28. Napraviti algoritam koji učitava elemente matrice kolonu po kolonu, a štampa matricu vrstu po vrstu.

Rešenje:



Primer 29. Napraviti algoritam koji učitava elemente kvadratne matrice ($m=n$) vrstu po vrstu, nalazi i štampa najveći element na glavnoj dijagonali.

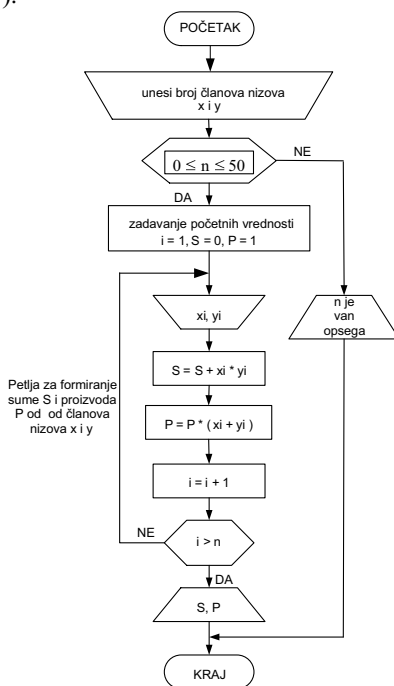
Rešenje:



Primer 30. Nacrtati algoritam za program koji učitava n parova brojeva ($1 \leq n \leq 50$) x_i i y_i ($i = 1, 2, \dots, n$) 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 zadanom opsegu, program generiše izveštaj i skače na kraj. Kad se broj n nalazi u zadanom opsegu, novi parovi brojeva ($x_i y_i$) se učitavaju u petlji pre formiranja sume proizvoda (S), odnosno proizvoda suma (P).



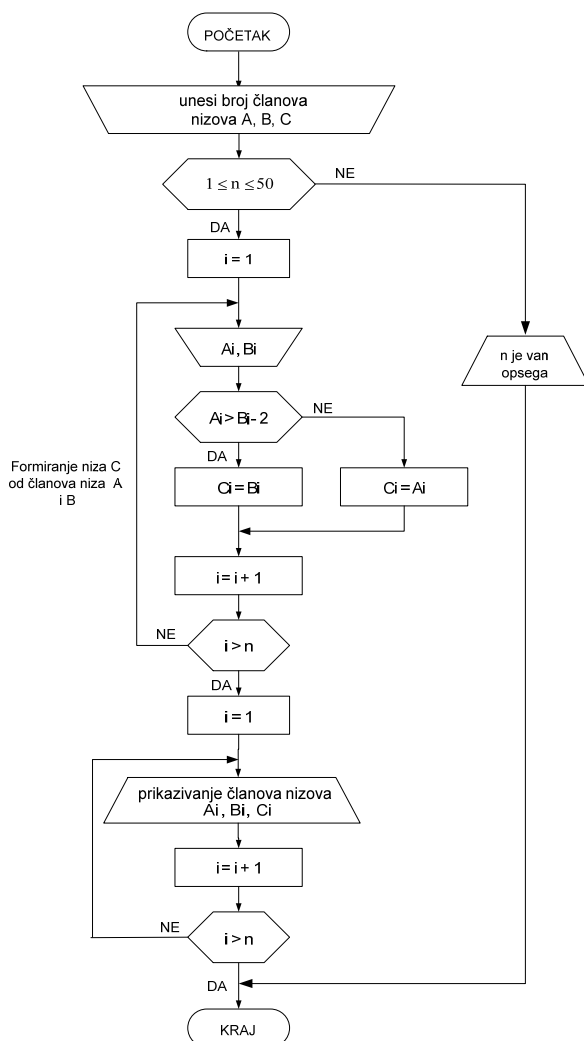
Primer 31. Nacrtati algoritam za program koji od dva niza A i B koji imaju po $1 \leq n \leq 15$ brojeva formira niz C sa elementima:

$$C_i = A_i, A_i \leq B_i - 2, i = 1, 2, \dots, n \text{ i } C_i = B_i, A_i > B_i - 2, i = 1, 2, \dots, n$$

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 zadanom 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.

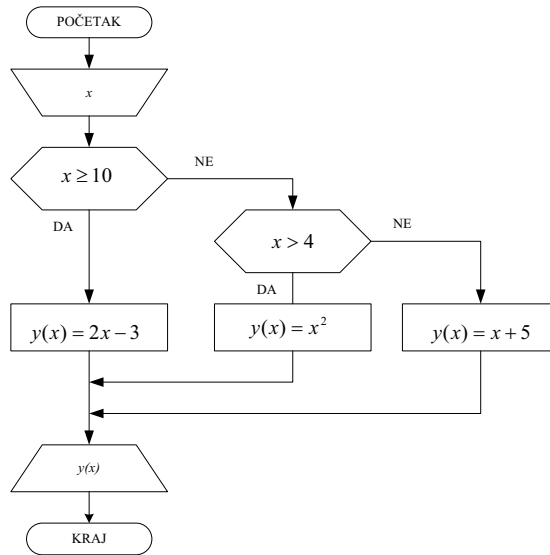


Primer 32. Nacrtati algoritam programa za izračunavanje i prikazivanje funkcije $y(x)$, gde je x zadata celobrojna ulazna promenljiva:

$$y(x) = \begin{cases} 2x - 3 & x \geq 10 \\ x^2 & 4 < x < 10 \\ x + 5 & x \leq 4 \end{cases}$$

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 $f(x)$:

$$f(x) = a_0 + a_1x + a_2x^2 + \dots + a_nx^n \quad 0 \leq n \leq 20$$

$$x = x_0 + k\Delta x \quad k = 0, 1, \dots, m$$

Poznate su ulazne vrednosti: $n, a_0, a_1, \dots, a_n, x_0$ i Δx i m .

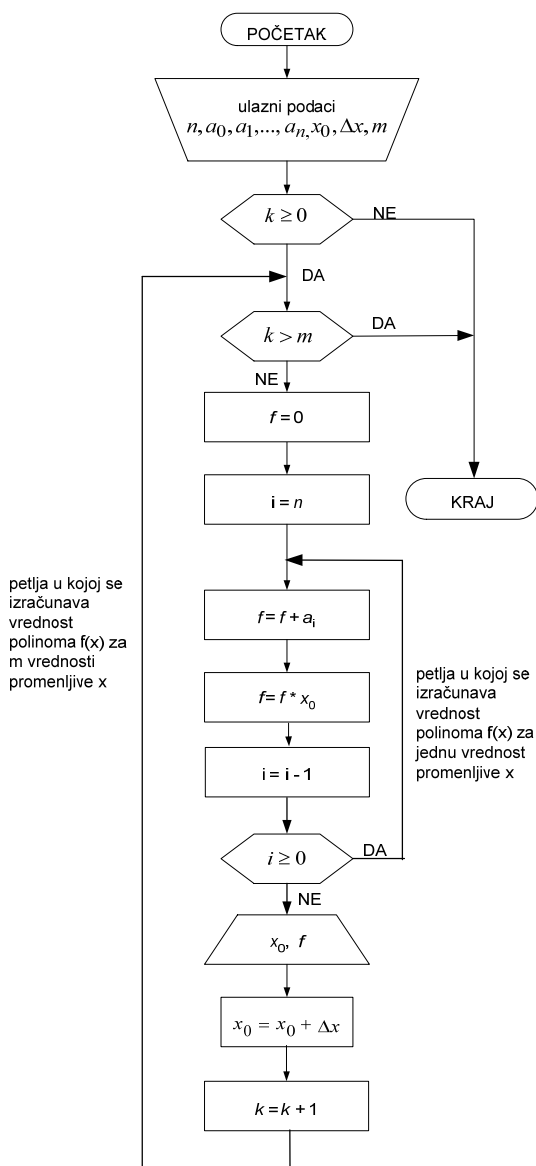
Rešenje:

Za organizovanje programskog ciklusa za izračunavanje polinoma $f(x)$, pogodno je da se polinom zapiše u obliku:

$$f(x) = a_0 + x(a_1 + x(a_2 + \dots x(a_{n-1} + xa_n)))$$

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 $i = n, n-1, \dots, 0$ 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 $k = 0, 1, \dots, m$ sa korakom +1, odnosno onoliko puta koliko puta treba izračunati vrednost polinoma.



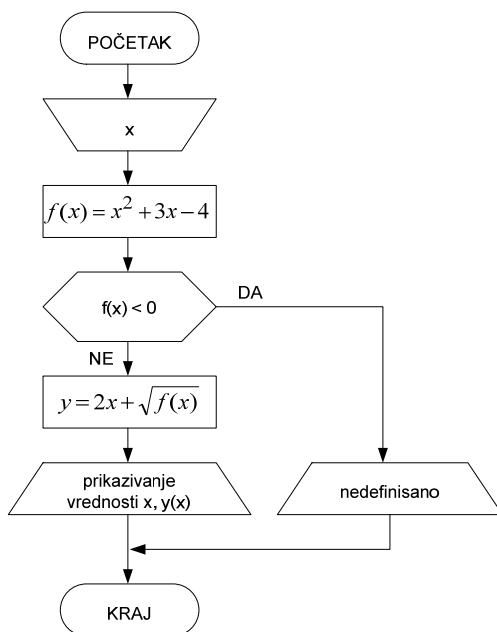
Primer 34. Definisati dijagram toka programa koji za zadatu vrednost promenljive x izračunava vrednost funkcije:

$$y(x) = 2x + \sqrt{x^2 + 3x - 4}$$

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 $y(x)$.



Primer 35. Definisati dijagram toka koji za zadate celobrojne vrednosti x_i i y_i , $i = 1, 2$ izračunava i prikazuje vrednost funkcije Z :

$$Z = \frac{F(x_1, y_1)}{F(x_2, y_2)}$$

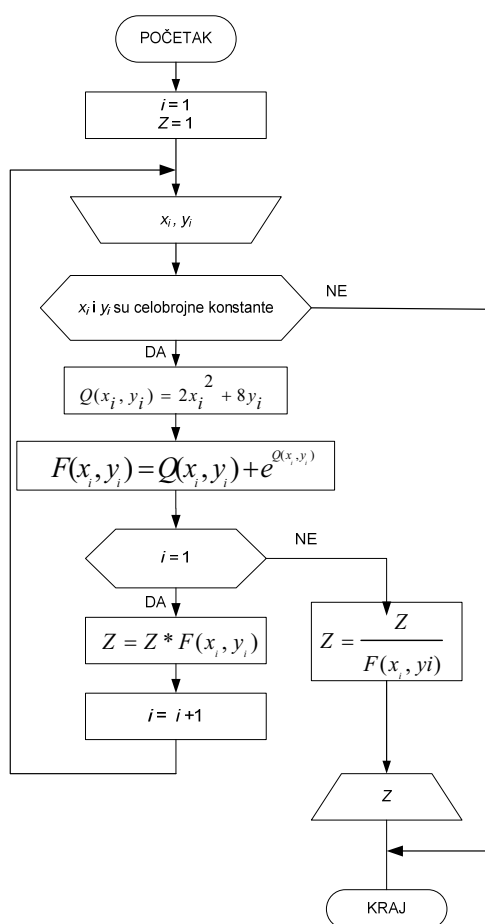
$$\text{ako je } F(x, y) = 2x^2 + 8y + e^{2x^2 + 8y}$$

Rešenje:

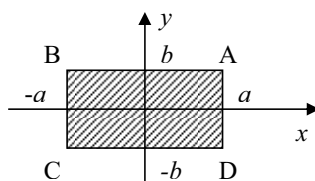
Izraz koji se ponavlja u funkciji F može da se definiše kao posebna funkcija $Q(x, y)$:

$$Q(x, y) = 2x^2 + 8y$$

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 $F(x_2, y_2)$.

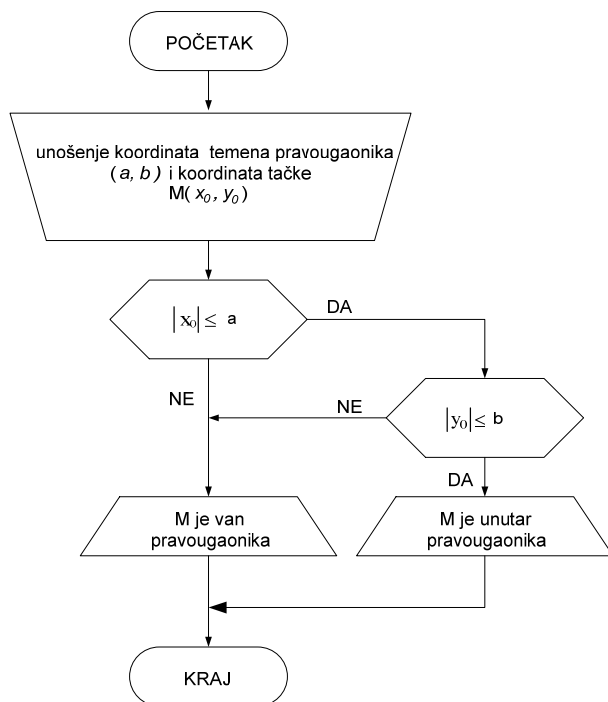


Primer 36. U pravougaonom koordinatnom sistemu zadat je pravougaonik sa temenima A, B, C i D. Ako su poznate koordinate temena, nacrtati dijagram toka kojim se određuje da li je tačka $M(x_0, y_0)$ unutar ili izvan pravougaonika.



Rešenje:

Ako su zadate veličine a i b , tada su poznate koordinate svih temena pravougaonika 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:

$$y(x) = \begin{cases} \sin 3x & k = 1 \\ \sin\left(\frac{\pi}{4} - x\right) \cos x & k = 2 \\ \cos(\pi + x) & k = 3 \\ 0 & k \notin \{1, 2, 3\} \end{cases}$$

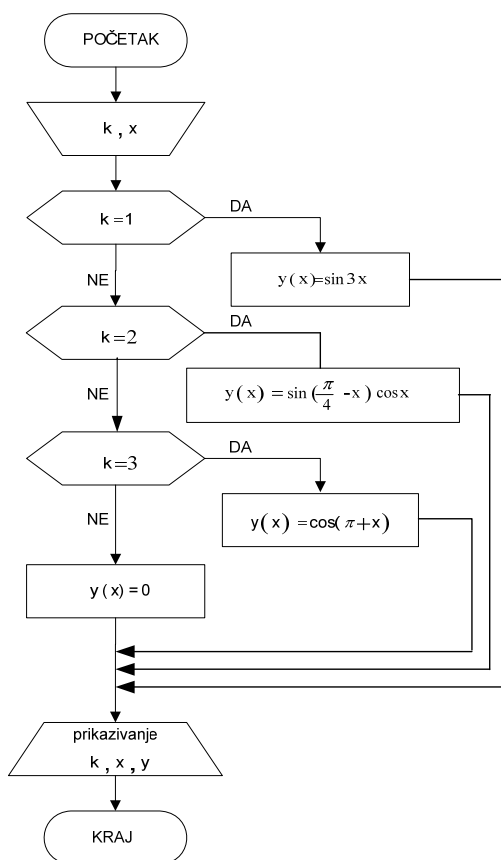
Rešenje:

Algoritam može da se realizuje kao:

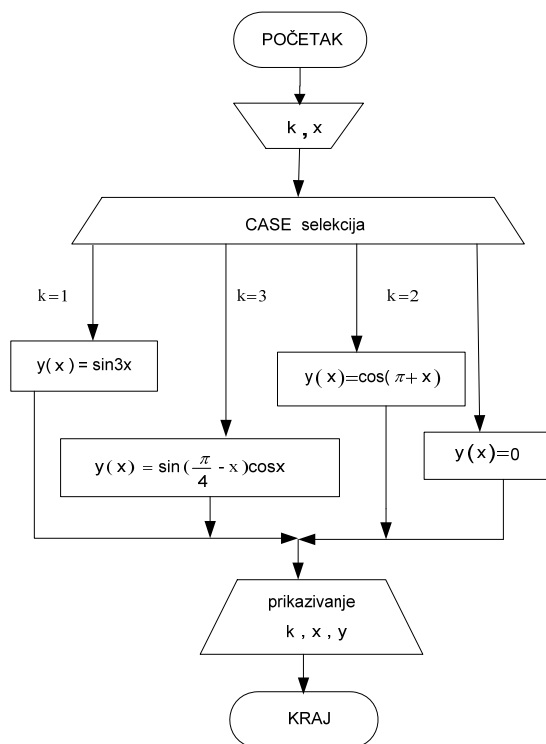
- klasična struktura sa višestrukim odlučivanjem i
- kao struktura sa CASE odlučivanjem.

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 $a_1, a_2, \dots, a_n, 1 \leq n \leq 10$ uređuje u opadajući niz.

Rešenje: Niz brojeva može da se uredi upoređivanjem svaka dva susedna broja u nizu:

$$a_i \geq a_{i+1}, \quad i = 1, 2, \dots, n$$

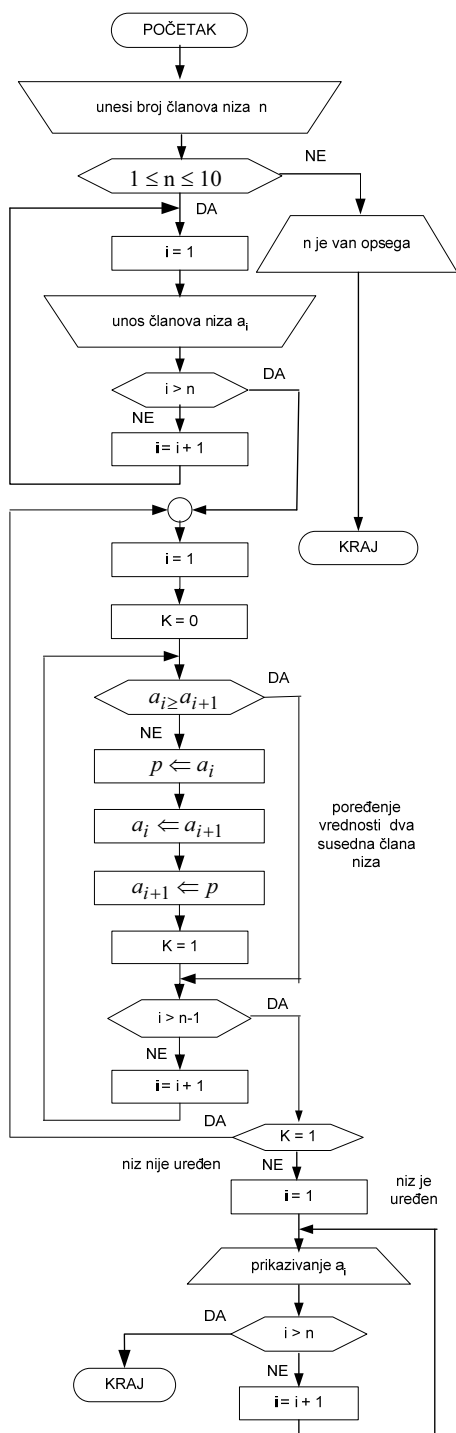
Mogu da nastanu dva slučaja: Brojevi a_i i a_{i+1} zadovoljavaju zahtevanu relaciju, pa takve članove niza ne treba premeštati. Brojevi a_i i a_{i+1} ne zadovoljavaju zahtevanu relaciju, pa takvim brojevima treba promeniti mesta u nizu.

$$p \leftarrow a_i$$

Razmena mesta članova u nizu može da se uradi na sledeći način: $a_i \leftarrow a_{i+1}$

$$a_{i+1} \leftarrow p$$

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.



Primer 39. Izračunati vrednost funkcije $f(x) = e^{-4x} \cos 4x$ u zadanom 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:

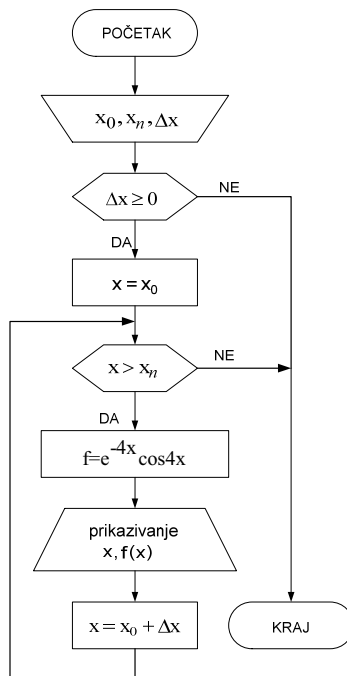
Vrednost funkcije $f(x)$ određuje se u petlji, počev od vrednosti x_0 . Petlja se izvršava sve dok promenljiva x ne dostigne vrednost x_n , s tim što se pri svakom prolasku kroz petlju vrednost promenljive x uvećava za Δx . Na početku programa treba uneti vrednosti za x_0 , x_n i Δx , pri čemu je uslov za izvršavanje petlje $x_0 \leq x_n$. Pošto je vrednost priraštaja $\Delta x \geq 0$, broj ciklusa izvršavanja 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.

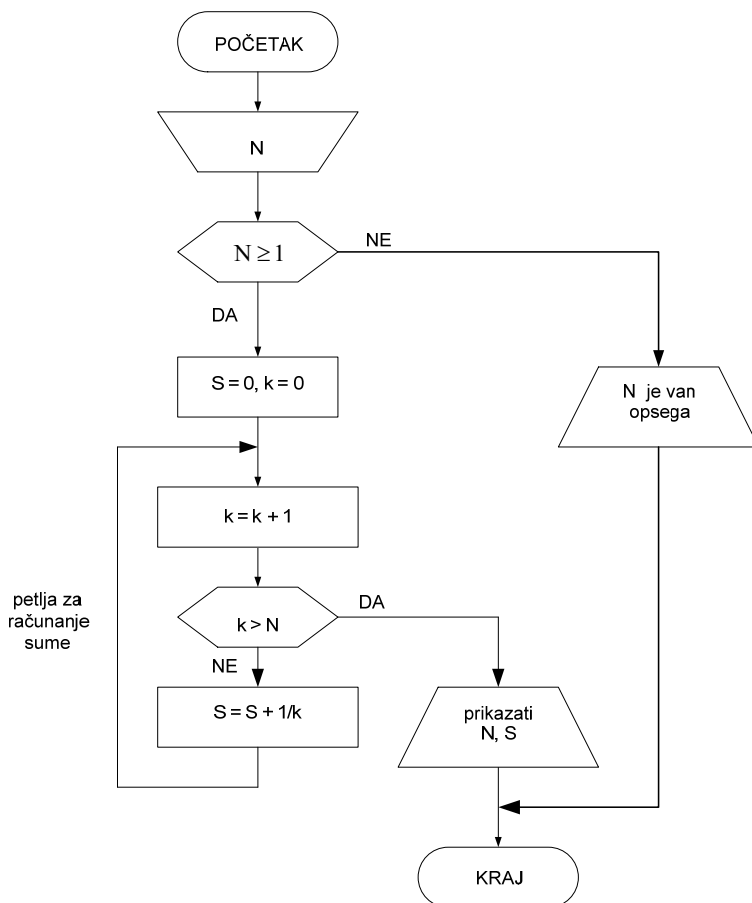


Primer 40. Nacrtati dijagram toka programa koji za zadatu vrednost promenljive $N \geq 1$ računa sumu:

$$S = \sum_{k=1}^N \frac{1}{k}$$

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 .



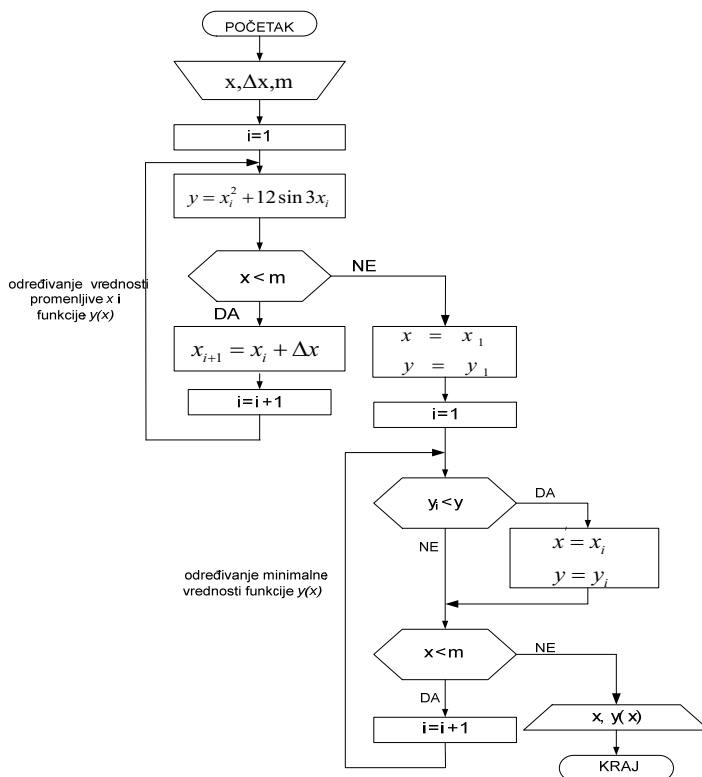
Primer 41. Nacrtati dijagram toka programa koji računa m vrednosti funkcije $y(x)$:

$$y(x) = x^2 + 12 \sin 3x$$

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:

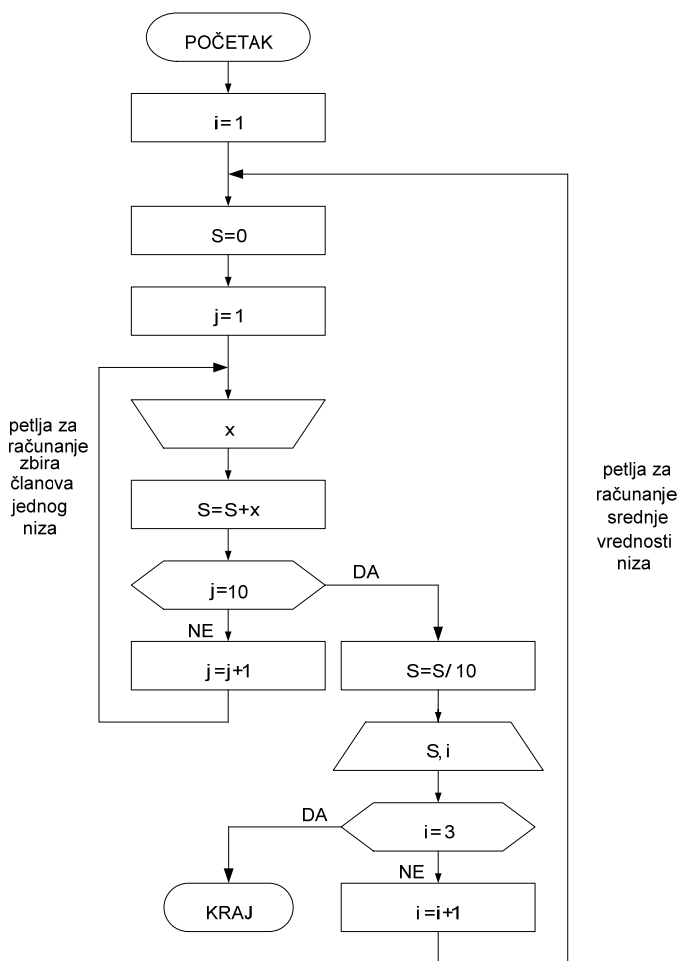
Dijagram toka sadrži dve nezavisne petlje. U prvoj se obavlja izračunavanje vrednosti funkcije y , i tada se dobija m parova $(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)$. U drugoj petlji se određuje par (x, y) koji se sastoji od minimalne vrednosti funkcije y i odgovarajuću vrednosti promenljive x , pri 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:

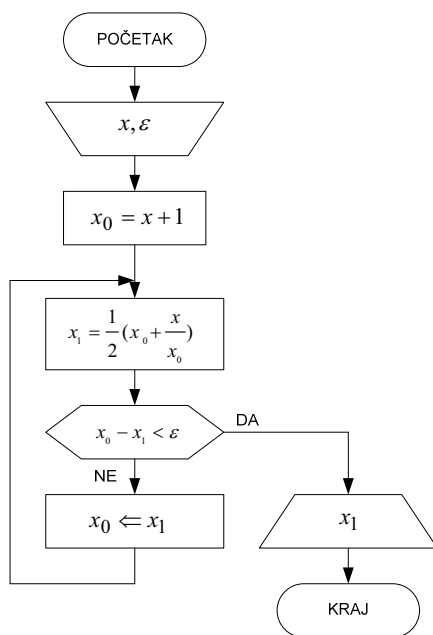
$$x_{i+1} = \frac{1}{2} \left(x_i + \frac{x}{x_i} \right) \quad i = 0, 1, 2, \dots$$

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 :

$$a_1x_i + b_1y_i = c_1$$

$$a_2x_i + b_2y_i = c_2$$

5. Sastaviti algoritam koji za dato x izračunava y prema navedenoj formuli:

$$y = \begin{cases} -4, & x < 0 \\ x + 2, & 0 \leq x < 1 \\ 3x - 1, & 1 \leq x < 5 \\ 2x, & x \geq 5 \end{cases}$$

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 ($i = 1, \dots, 8$):

$$Z_i(x) = \sqrt{\ln(x_i) \sin^2 x_i}$$

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 $\forall a \in \mathbb{R}$ i $\forall n \in \mathbb{Z}$.
14. Napraviti 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 $\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \dots$ 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 $\cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \dots$ 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 učitavaju 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", *Akadska štampa*, Beograd, 2003.
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.

SADRŽAJ

1. Brojni sistemi i konverzija brojeva iz jednog brojnog sistema u drugi.....	1
2. Pojam komplementa, binarni brojni sistem i binarni brojevi sa znakom.....	10
3. Format zapisa brojeva u računarskom sistemu.....	26
4. Sabiranje primenom kodova "8421" i "Više 3".....	40
5. Analiza i sinteza logičkih funkcija.....	46
6. Minimizacija logičkih funkcija primenom Karnoovih mapa.....	60
7. Primena unarnih i binarnih logičkih operacija.....	94
8. Algoritmi.....	107
Literatura.....	148