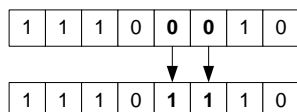


## 6. PRIMENA UNARNIH I BINARNIH LOGIČKIH INSTRUKCIJA

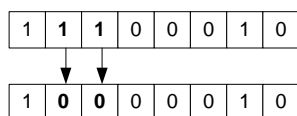
### TEORIJA:

UNARNE INSTRUKCIJE su instrukcije sa **jednim** operandom:

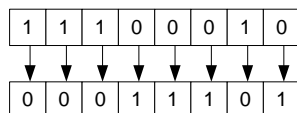
- **Postavljanje (SET)** - logička vrednost **0** nekog **bita** pretvara se u logičku vrednost **1**.  
Primenjuje se na bite u registru stanja, registrima CPU i memorijskim lokacijama.



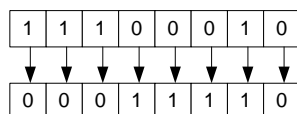
- **Brisanje (CLEAR, RESET)** - logička vrednost **1** željenog **bita** se pretvara u logičku vrednost **0**.  
Primenjuje se na bite u registru stanja, registrima CPU i memorijskim lokacijama.



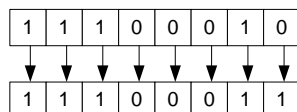
- **Komplementiranje (COMPLEMENT)** - sve logičke **0** u jednom broju se pretvaraju u **1** i obrnuto, tako da se dobija **prvi komplement**.  
Operacija se odnosi na sadržaj celog registra.



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



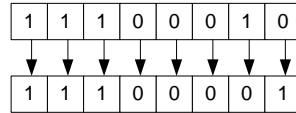
- **Inkrementiranje (INCREMENT)** - povećava vrednosti operanda za **1**.  
Odnosi se na sadržaj celog registra.  
Kod nekih procesora se primenjuje samo na registre u okviru CPU, a kod drugih se primenjuje i na sadržaj memorijskih lokacija.



- **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 pomera sadržaj za više pozicija.

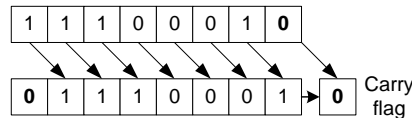
Upražnjena mesta se obično popunjavaju logičkim nulama (0).

#### ➤ Pomeranje udesno

Primena: Ispitivanje sadržaja bita u registru

Deljenje sa  $2^N$ ,  $N$  = broj pomeranja

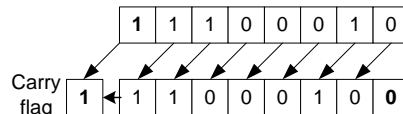
Aritmetičko pomeranje udesno - bit znaka ostaje na MSB poziciji, a pomeraju se svi ostali biti.



#### ➤ Pomeranje ulevo

Primena: Ispitivanje sadržaja bita u registru

Množenje sa  $2^N$ ,  $N$  = broj pomeranja

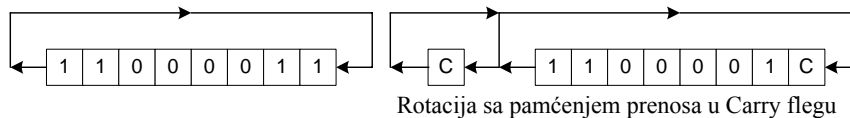
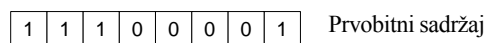


- **Rotacija (ROTATE)** - rotiraju se biti u okviru jednog registra ulevo ili udesno.

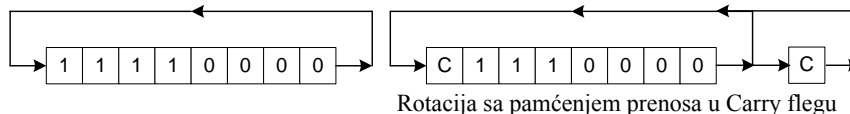
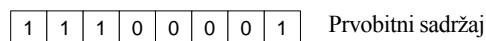
Bit koji se rotira upisuje se na upražnjeno mesto na drugom kraju registra, a pri tome može da se čuva i u Carry flag-u.

Kod nekih procesora i sam bit prenosa (C, carry) učestvuje u rotaciji.

#### ➤ Rotacija ulevo



#### ➤ Rotacija udesno



**BINARNE LOGIČKE INSTRUKCIJE** su instrukcije sa **dva** operanda, od kojih je najčešće bar jedan u nekom od akumulatora (ili drugim registrima CPU), dok je drugi u nekom od registara memorije.

Logičke instrukcije obrađuju sve bite u registru, ali se primenjuju na svaki par bita ponaosob i za svaki par generišu poseban rezultat, koji ne utiče na rezultat u drugim bitima (nema prenosa između dva susedna razreda u registru).

- **Logička I (AND) operacija**

Primena: Izdvajanje sadržaja dela registra,

Resetovanje željenih bita,

pomoću **maske** koja sadrži **1** na mestu bita čiji sadržaj treba da ostane **nepromenjen** (koji treba **izdvojiti**), a **0** na mestu bita koje treba **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 **setovati**.

A        1 0 1 1 0 0 0 0 30 (HEX) originalan sadržaj registra

B        1 0 0 0 0 1 0 1 05 (HEX) maska

A OR B   1 0 1 1 0 1 0 1 35 (HEX) 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

## **PRIMERI:**

1. 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
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

2. 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: 00110101010101. 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
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

$A \text{ XOR } M \rightarrow A$ : 

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

3. 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
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

$A \text{ XOR } M \rightarrow A$ : 

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

4. 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:**

A: 

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

M: 

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

$A \text{ XOR } M \rightarrow A$ : 

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

5. U akumulatoru A nalazi se sledeći podatak A : 100110111011110, 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. Nakon izvršenja logičke operacije rezultat se nalazi u akumulatoru A.

**Rešenje:**

A: 

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

M: 

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

$A \text{ AND } M \rightarrow A$ : 

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

6. 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
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

7. Primenom odgovarajuće maske i logičke operacije **resetovati** 0.,3.,4.,8.,11.,14. i 15. bit u 16-bitnom broju koji se nalazi u akumulatoru A: 1010101010101010. 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
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

8. 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
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

9. 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
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

10. 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	0	1	0	1	0	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

A AND M → A: 

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

11. 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
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

12. 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**

A: 

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

M: 

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

A AND M → A: 

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

13. 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**<sub>(2)</sub>
- Posle rotacije za 3 mesta ulevo, sadržaj u registru A je: **11110011**<sub>(2)</sub>
- Posle pomeranja za 3 mesta udesno, sadržaj u registru A je: **00011111**<sub>(2)</sub>
- Posle rotacije za 3 mesta udesno, sadržaj u registru A je: **11001111**<sub>(2)</sub>

14. 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:

- a) viši nibl u registru A bude 0
- b) niži nibl u registru A bude 0
- c) viši nibl dođe na mesto nižeg nibla
- d) niži nibl dođe na mesto višeg nibla

**Rešenje:**

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

- a) Posle pomeranja za 4 mesta udesno, sadržaj u registru A je **00000011**<sub>(2)</sub> = **03**<sub>(16)</sub>
- b) Posle pomeranja za 4 mesta ulevo, sadržaj u registru A je **10010000**<sub>(2)</sub> = **90**<sub>(16)</sub>
- c) Posle rotacije za 4 mesta udesno, sadržaj u registru A je **10010011**<sub>(2)</sub> = **93**<sub>(16)</sub>
- d) Posle rotacije za 4 mesta ulevo, sadržaj u registru A je **10010011**<sub>(2)</sub> = **93**<sub>(16)</sub>

15. 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**<sub>(2)</sub> = **3D**<sub>(16)</sub>
- b) Posle pomeranja za 4 mesta ulevo, sadržaj u registru A je **01100000**<sub>(2)</sub> = **60**<sub>(16)</sub>
- c) Posle rotacije za 4 mesta udesno, sadržaj u registru A je **01101111**<sub>(2)</sub> = **6F**<sub>(16)</sub>
- d) Posle rotacije za 4 mesta ulevo, sadržaj u registru A je **01101111**<sub>(2)</sub> = **6F**<sub>(16)</sub>

16. 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**<sub>(2)</sub>
- b) Posle rotacije za 2 mesta ulevo, ako je C = **1** sadržaj u registru A je **11001111**<sub>(2)</sub>
- c) Posle rotacije za 4 mesta udesno, ako je C = **0** sadržaj u registru A je **01101011**<sub>(2)</sub>
- d) Posle rotacije za 2 mesta ulevo, sadržaj u registru A je **11001101**<sub>(2)</sub>

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

**Rešenje:**      **0001100100100001**

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

**Rešenje:**        **110000000000111**

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

**Rešenje:**        **0101001000000000**

20. 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**

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

- a) Primenom odgovarajuće logičke operacije i maske pretvoriti ASCII kod brojeva u registru A i B u raspakovani format BCD koda "8421".
- b) Kako primenom rotacije i odgovarajuće logičke operacije nad sadržajem registara A i B dobijenim u tački a) može da se postigne da se u registru A nalazi sadržaj  $79_{(16)}$  (pakovani BCD format).
- c) Kako primenom pomeranja i odgovarajuće logičke operacije nad sadržajem registara A i B dobijenim u tački a) može da se postigne da se u registru B nalazi sadržaj  $79_{(16)}$  (pakovani BCD format).

**Rešenje:**

Prvobitni sadržaj u registru A je  $00111001_{(2)}$ , a u registru B je  $00110111_{(2)}$

a) Primenom logičke I (AND) operacije nad sadržajem registara A, odnosno B i maske  $00001111_{(2)}$  dobija se zapis brojeva u raspakovanom formatu BCD koda "8421":

$$A = 00111001_{(2)} \text{ AND } 00001111_{(2)} = \mathbf{00001001}_{(2)}$$

$$B = 00110111_{(2)} \text{ AND } 00001111_{(2)} = \mathbf{00000111}_{(2)}$$

b) Primenom rotacije sadržaja registra B za 4 mesta udesno dobija se  $B = 01110000_{(2)}$

Primenom operacije **ILI (OR)** između registara A i B, pri čemu se rezultat čuva u registru A dobija se  $A = A \text{ OR } B = \mathbf{01111001}_{(2)} = \mathbf{79_{(16)}}$

c) Pomeranjem sadržaja registra B za četiri mesta ulevo dobija se  $B = 01110000_{(2)}$

Primenom operacije **ILI (OR)** između registara A i B, pri čemu se rezultat čuva u registru B dobija se  $B = A \text{ OR } B = \mathbf{01111001}_{(2)} = \mathbf{79_{(16)}}$

22. U registru A se nalazi binarni zapis broja  $24_{(10)}$ .

- a) Koji dekadni broj se nalazi u registru posle 3 pomeranja binarnog sadržaja udesno?
- b) Koja aritmetička operacija je realizovana na ovaj način?

**Rešenje:**

Prvobitni sadržaj u registru A je  $00011000_{(2)}$

a) Posle pomeranja za 3 mesta udesno sadržaj registra A je  $\mathbf{00000011}_{(2)} = \mathbf{3_{(10)}}$

b) Na ovaj način je izvršeno deljenje broja 24 sa  $2^3 = 8$ .



23. U registru A se nalazi binarni zapis broja 6.  
a) Koji dekadni broj se nalazi u registru posle 5 pomeranja binarnog sadržaja ulevo?  
b) Koja aritmetička operacija je realizovana na ovaj način?

**Rešenje:**

Prvobitni sadržaj u registru A je 00000110<sub>(2)</sub>

- a) Posle pomeranja za 5 mesta ulevo sadržaj registra A je **11000000**<sub>(2)</sub> = **192**<sub>(10)</sub>  
b) Na ovaj način je izvršeno množenje broja 6 sa  $2^5 = 32$

24. U registru A se nalazi binarni zapis broja 72.  
a) Koji dekadni broj se nalazi u registru posle 2 pomeranja binarnog sadržaja ulevo?  
b) Koja aritmetička operacija je realizovana na ovaj način?
25. U registru A se nalazi binarni zapis broja 9.  
a) Koji dekadni broj se nalazi u registru posle 4 pomeranja binarnog sadržaja ulevo?  
b) Koja aritmetička operacija je realizovana na ovaj način?
26. U registru A se nalazi binarni zapis broja 40.  
a) Kako primenom operacije pomeranja u odgovarajućem smeru broj može da se podeli sa 8?  
b) Šta se nalazi u registru nakon poslednje operacije pomeranja?
27. U registru A se nalazi binarni zapis broja 16.  
a) Kako primenom operacije pomeranja u odgovarajućem smeru ovaj broj može da se pomnoži sa 16?  
b) Šta se nalazi u registru nakon poslednje operacije pomeranja?
28. U 16-bitnom registru A nalazi se binarni broj 0011000011101011. Primenom odgovarajuće maske i logičke operacije **invertovati**:  
a) 1., 2., 5., 8., 10., 12. i 14. bit;  
b) niži bajt i najviši bit;  
c) po četiri krajnja bita;  
d) svaki neparni bit;  
e) 4., 5., 8., 9., 11., 13. i 15. bit.
29. U 16-bitnom registru A nalazi se binarni broj 0011010100001111. Primenom odgovarajuće maske i logičke operacije **isključiti**:  
a) 0., 2., 7., 10., 11., 12. i 15. bit;  
b) sve parne bitove;  
c) sve neparne bitove;  
d) viši bajt;  
e) 3., 4., 8., 9., 12. i 13. bit.
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.

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.
32. U 16-bitnom registru A nalazi se binarni broj 1111100111101111. Primenom odgovarajuće maske i logičke operacije:
- a) izdvojiti viši bajt
  - b) izdvojiti niži bajt
  - c) izdvojiti 5., 6., 7., 8., 12., 13., 14. i 15. bit.
  - d) resetovati najniža 4 bita i najviša 4 bita
  - e) resetovati 1., 3., 5., 12. i 14. bit
33. 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
34. 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
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.
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.
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.

38. U 8 – bitnom registru A se nalazi HEX broj 4, a u registru B je HEX broj 5.

- a) Primenom odogovarajuće logičke operacije i maske konvertovati sadržaj registara A i B u odgovarajuće ASCII kodove.
- b) Kako primenom pomeranja i odgovarajuće logičke operacije nad zadatim prvobitnim sadržajem registara A i B može da se postigne da se u registru A nalazi HEX broj  $45_{(16)}$  (pakovani BCD format).
- c) Kako primenom pomeranja i odgovarajuće logičke operacije nad zadatim prvobitnim sadržajem registara A i B može da se postigne da se u registru B nalazi HEX broj  $54_{(16)}$  (pakovani BCD format).