BROJNI SISTEMI I KODOVI

Zadatak 1

Sledeće brojeve date u brojnim sistema sa različitom osnovom, prebaciti u traženi brojni sistem.

a) Brojeve

- 1100101.110_2
- $A38C.B_{16}$
- 753.1_8
- 254.61_7

prebaciti u broj u decimalnom brojnom sistemu.

b) Brojeve

- 191.18_{10}
- 01101010.110_2
- 156.5_8

prebaciti u broj u heksadecimalnom brojnom sistemu.

c) Brojeve

- 1105.33_{10}
- $18DA.C_{16}$
- 6655.44_8
- 254.61_7

prebaciti u broj u binarnom brojnom sistemu.

d) Brojeve

- 2245.9_{10}
- 123.45_{16}
- 11010101.1101_2

prebaciti u broj u oktalnom brojnom sistemu.

REŠENJE:

a) Konverzija brojeva datih u proizvoljnom brojnom sistemu, u broj u decimalnom brojnom sistemu, vrši se na osnovu težinskih koeficijenata cifara, koji odgovaraju poziciji cifre u posmatranom broju i decimalnih vrednosti cifara.
Za broj dat u binarnom brojnom sistemu decimalna vrednost broja biće određena:
$1100101.110_2 = 1\cdot2^6+1\cdot2^5+0\cdot2^4+0\cdot2^3+1\cdot2^2+0\cdot2^1+1\cdot2^0+1\cdot2^{-1}+1\cdot2^{-2}+0\cdot2^{-3} = 101.75_{10}$

Konverzija broja datog u heksadecimalnom brojnom sistemu u broj dat u decimalnom brojnom sistemu izgledaće:
$A38C.B_{16} = 10\cdot16^3+3\cdot16^2+8\cdot16^1+12\cdot16^0+11\cdot16^{-1} = 41868.6875_{10}$

Konverzija broja datog u oktalnom brojnom sistemu u broj u decimalnom brojnom sistemu vrši se na sledeći način:

$753.1_8 = 7 \cdot 8^2 + 5 \cdot 8^1 + 3 \cdot 8^0 + 1 \cdot 8^{-1} = 491.125_{10}$

Konverzija broja datog u brojnom sistemu sa osnovom 7 u broj u decimalnom brojnom sistemu izgledaće:
$254.61_7 = 2 \cdot 7^2 + 5 \cdot 7^1 + 4 \cdot 7^0 + 6 \cdot 7^{-1} + 1 \cdot 7^{-2} = 132.8776_{10}$

b) Konverziju broja, iz decimalnog brojnog sistema, u broj u nekom drugom brojnom sistemu, vršimo na osnovu ostatka deljenja broja u decimalnom brojnom sistemu sa osnovom traženog brojnog sistema (decimalnom vrednosti osnove traženog sistema). Dakle, ukoliko broj u decimalnom brojnom sistemu prebacujemo u heksadecimalni brojni sistem, deljenje vršimo brojem 16. Ostatak inicijalnog deljenja predstavlja LSD[1] cifru traženog broja. Postupak se nastavlja tako što rezultat deljenja, dalje delimo sa decimalnom vrednosti osnove traženog sistema (decimalnim brojem 16), itd. Cifre iza decimalne tačke na sličan način konvertujemo u heksadecimalni broj, sa tom razlikom, što umesto operacije deljenja obavljamo operaciju množenja sa decimalnim brojem koji predstavlja osnovu traženog sistema (u slučaju heksadecimalnog brojnog sistema vrši se množenje brojem 16).
Za date decimalne brojeve iz postavke zadatka, imamo:
$191_{10} = BF_{16}$
$191/16 = 11$ ostatak $15_{10} = F_{16}$ (LSD-Cifra najmanje težine)
$11/16 = 0$ ostatak $11_{10} = B_{16}$ (MSD[2]- Cifra najmanje težine)
$0.18_{10} = 0.2E14\ldots_{16}$
$0.18 \cdot 16 = 2_{10} = 2_{16}$ (MSD) ostatak 0.88
$0.88 \cdot 16 = 14_{10} = E_{16}$ ostatak 0.08
$0.08 \cdot 16 = 1_{10} = 1_{16}$ ostatak 0.28
$0.28 \cdot 16 = 4_{10} = 4_{16}$ (LSD) ostatak 0.48, itd.

Dakle, dobijamo da je:
$191.18_{10} = BF.2E14\ldots_{16}$

Konverziju broja iz binarnog u heksadecimalni brojni sistem, vršimo direktno, obzirom da je svaka heksadecimalna cifra predstavljena sa 4 bita u binarnom obliku.
$01101010.110_2 = 6A.C_{16}$

Slično kao u slučaju broja datog u binarnom brojnom sistemu, svaka cifra u oktalnom brojnom sistemu može se predstaviti sa 3 bita u binarnom brojnom sistemu i zatim se dobijeni broj u binarnom obliku lako predstavlja u heksadecimalnom brojnom sistemu.
$156.5_8 = 001101110.101_2 = 6E.A_{16}$

c) Za konverziju broja datog u decimalnom, oktalnom ili heksadecimalnom brojnom sistemu u broj u binarnom brojnom sistemu, pogledati rešenje iz tačke **b**.
$1105_{10} = 10001010001_2$
$1105/2 = 552$ ostatak 1 (LSB[3])
$552/2 = 276$ ostatak 0
$276/2 = 138$ ostatak 0

[1] LSD - *Least Significant Digit*
[2] MSD - *Most Significant Digit*
[3] LSB - *Least Significant Bit*

$138/2 = 69$ ostatak 0
$69/2 = 34$ ostatak 1
$34/2 = 17$ ostatak 0
$17/2 = 8$ ostatak 1
$8/2 = 4$ ostatak 0
$4/2 = 2$ ostatak 0
$2/2 = 1$ ostatak 0
$1/2 = 0$ ostatak 1 (MSB^4)

$0.33_{10} = 010101\ldots_2$
$0.33 \cdot 2 = 0$ (MSB) ostatak 0.66
$0.66 \cdot 2 = 1$ ostatak 0.32
$0.32 \cdot 2 = 0$ ostatak 0.64
$0.64 \cdot 2 = 1$ ostatak 0.28
$0.28 \cdot 2 = 0$ ostatak 0.56
$0.56 \cdot 2 = 1$ (LSB) ostatak 0.12, itd.

Dakle, dobijamo da je:
$1105.33_{10} = 10001010001.010101\ldots_2$
Na sličan način dobijamo:
$18DA.C_{16} = 0001100011011010.1100_2$
$6655.44_8 = 110110101101.100100_2$

Prilikom prebacivanja broja iz osnove 7 u osnovu 2 (i osnovu koja je stepen broja 2) najjednostavnije je prvo prebaciti broj u osnovu 10, a zatim u broj sa osnovom 2:
$254.61_7 = 137.8776_{10} = 10001001.111000\ldots_2$

d) Za konverzije brojeva iz decimalnog, heksadecimalnog i binarnog brojnog sistema u broj u oktalnom brojnom sistemu, pogledati rešenje iz tačke **b**.
$2245_{10} = 4305_8$
$2245/8 = 280$ ostatak 5_8 (LSD)
$280/8 = 35$ ostatak 0
$35/8 = 4$ ostatak 3_8
$4/8 = 0$ ostatak 4_8 (MSD)

$0.9_{10} = 0.714631463\ldots_8$ $\quad 0.9 \cdot 8 = 7$ (MSD) ostatak 0.2
$0.2 \cdot 8 = 1$ ostatak 0.6
$0.6 \cdot 8 = 4$ ostatak 0.8
$0.8 \cdot 8 = 6$ ostatak 0.4
$0.4 \cdot 8 = 3$ (LSD) ostatak 0.2, itd.

Dakle, dobijamo da je: $2245.9_{10} = 4305.714631463\ldots_8$

Na sličan način dobijamo:
$123.45_{16} = 000100100011.01000101_2 = 443.212_8$
$11010101.1101_2 = 325.64_8$

[4] MSB - *Most Significant Bit*

Zadatak 2

a) Data je jednačina $x^2 - 11x + 22 = 0$ i njeno jedno rešenje $x = 3$. U kom brojnom sistemu je data jednačina i njeno rešenje?
b) Data je jednačina $5x^2 - 74x + 207 = 0$ i njeno jedno rešenje $x = 11$. U kom brojnom sistemu je data jednačina i njeno rešenje?
c) Data je jednačina $5x^2 - 50x + 125 = 0$ i oba rešenja jednačine $x_1 = 5$ i $x_2 = 8$. U kom brojnom sistemu je data jednačina i oba njena rešenja?
d) Odrediti rešenje jednačine $202_x = 20_{50}$.

REŠENJE:

a) Kako je data jednačina data u nepoznatom brojnom sistemu, ista jednačina će u decimalnom brojnom sistemu imati oblik:

$$x^2 - (p+1)x + (2p+2) = 0 \tag{2.1}$$

U izrazu 2.1, p predstavlja osnovu nepoznatog brojnog sistema u kome je data polazna jednačina. U jednačini 2.1, broj 11 u sistemu sa osnovom p, predstavljen je u decimalnom brojnom sistemu kao:

$$1 \cdot p^1 + 1 \cdot p^0 = p + 1$$

Na sličan način je u jednačini 2.1, predstavljen broj 22 dat u sistemu sa osnovom p.

Kako rešenje jednačine $x = 3$, ima isti oblik u decimalnom brojnom sistemu, nakon zamene u jednačinu 2.1, dobijamo polaznu jednačinu iz postavke zadatka u decimalnom brojnom sistemu:

$$9 - (p+1)3 + (2p+2) = 0$$

Nakon sređivanja, dobijamo rešenje jednačine:

$$p = 8$$

Dakle jednačina iz tačke **a** i njeno rešenje $x = 3$ su dati u oktalnom brojnom sistemu.

b) Kako je data jednačina data u nepoznatom brojnom sistemu, ista jednačina će u decimalnom brojnom sistemu izgledati:

$$2 - (7p+4)x + (2p^2+7) = 0 \tag{2.2}$$

U izrazu 2.2, p predstavlja osnovu traženog brojnog sistema u kome je data polazna jednačina.
Kako je, prema tekstu zadatka, rešenje jednačine 2.2, $x = 11$ dato u nepoznatom brojnom sistemu, treba voditi računa da će isto rešenje jednačine u decimalnom brojnom sistemu imati drugačiji oblik, odnosno zavisiće od decimalne vrednosi osnove brojnog sistema u kome je dato.
Dakle, rešenje jednačine u decimalnom brojnom sistemu imaće oblik:

$$x = 1 \cdot p^1 + 1 \cdot p^0 = p + 1$$

Nakon zamene rešenja jednačine $x = 11$ u izraz 2.2, dobijamo:

$$(p+1)2 - (7p+4)(p+1) + (2p^2+7) = 0 \tag{2.3}$$

Nakon sređivanja izraza 2.3, dobijamo rešenje koje predstavlja osnovu traženog brojnog sistema:

$$p = 8$$

Dakle jednačina iz tačke **b** i njeno rešenje su, isto kao i u tački **a**, dati u oktalnom brojnom sistemu.

c) Kako je data jednačina data u nepoznatom brojnom sistemu, ista jednačina će u decimalnom brojnom sistemu imati oblik:

$$2 - (5p)x + (1p^2 + 2p + 5) = 0 \tag{2.4}$$

Slično kao i ranije, u izrazu 2.4, p predstavlja osnovu brojnog sistema u kome je data polazna jednačina.
Kako je jedno od rešenja jednačine $x_1 = 5$, zamenom istog u jednačinu 2.4 dobijamo kvadratnu jednačinu:

$$p^2 - 23p + 130 = 0 \tag{2.5}$$

Rešenja jednačine 2.5, su:

$$p = 10 \text{ i } p = 13$$

Zaključujemo da je brojni sistem u kome je data jednačina sistem sa osnovom 13, obzirom da $x = 8$ nije rešenje jednačine $5x^2 - 50x + 125 = 0$ u decimalnom brojnom sistemu.

d) Potrebno je jednačinu predstaviti u decimalnom brojnom sistemu:

$$2x^2 + 2 = 2 \cdot 50 \tag{2.6}$$

Rešavanjem kvadratne jednačine dobija se da je $x = 7$ pošto osnova brojnog sistema mora da bude pozitivan broj.

Zadatak 3

a) Izvršiti operacije sabiranja sledećih neoznačenih binarnih brojeva:

$$10101011 + 00011100 \text{ i } 11101001 + 11011110,$$

i odrediti sve bite rezultata sabiranja s_i i sve bite prenosa c_i ($c_{i+1|i+1\neq 0}$ je prenos koji nastaje prilikom operacije sabiranja bita na poziciji i, tj. bita a_i, b_i i c_i, dok je c_0 ulazni bit prenosa koji se koristi prilikom operacije sabiranja bita na poziciji 0).

$$\begin{array}{rrcl} & c_8c_7c_6c_5c_4c_3c_2c_1c_0 & = & C \\ & a_7a_6a_5a_4a_3a_2a_1a_0 & = & A \\ + & b_7b_6b_5b_4b_3b_2b_1b_0 & = & B \\ \hline & s_8s_7s_6s_5s_4s_3s_2s_1s_0 & = & S \end{array}$$

b) Dati su brojevi A i B u hesadecimalnom brojnom sistemu. Izvršiti operacije sabiranja datih brojeva i odrediti sve cifre prenosa c_i i cifre rezultata sabiranja s_i:

$$1234ABCD + BA1B2C3F4 \text{ i } A00C00DE + 7FFFE000.$$

c) Dati su brojevi A i B u oktalnom brojnom sistemu. Izvršiti operacije sabiranja datih brojeva i odrediti sve cifre prenosa c_i i sve cifre rezultata sabiranja s_i:

$$12347650 + 33445566 \text{ i } 77665544 + 11223344.$$

REŠENJE:

a) Operaciju sabiranja brojeva datih u proizvoljnom brojnom sistemu započinjemo, isto kao i u decimalnom brojnom sistemu, sabiranjem cifara najmanje težine, pri čemu podrazumevamo da je vrednost ulaznog bita prenosa $c_0 = 0$. Ukoliko je rezultat operacije sabiranja cifara na poziciji i veći ili jednak osnovi brojnog sistema u kome se vrši sabiranje, generiše se izlazni prenos na poziciji sabiranja cifre (pozicija i), tj. $c_{i+1} = 1$. U opštem slučaju, rezultat sabiranja na poziciji i je jednak rezultatu sabiranja $d_i = a_i + b_i + c_i$ po modulu r, gde je r osnova brojnog sistema u kome se vrši sabiranje. Prema postavci zadatka, vidimo da nema ograničenja u broju cifara rezultata sabiranja obzirom da je potrebno odrediti i cifru rezultata sabiranja s_8. Prilikom operacije sabiranja dva broja sa n cifara rezultat operacije sabiranja se uvek može predstaviti sa $n+1$ cifrom. *MSD* cifra rezultata sabiranja biće jednaka izlaznom prenosu c_8 i može imati vrednost 0 ili 1.
Prilikom sabiranja brojeva datih u binarnom brojnom sistemu, imaćemo:

`C =   001110000` (*LSB* bit je ulazni bit prenosa c_0)
`A      10101011`
`+ B    00011100`
`S =   011000111`

`C =   111110000` (*LSB* bit je ulazni bit prenosa c_0)
`A      11101001`
`+ B    11011110`
`S =   111000111`

b) Za pojašnjenje operacije sabiranja brojeva u heksadecimalnom brojnom sistemu pogledati rešenje iz tačke **a**.

`C =   000010110` (*LSB* bit je ulazni bit prenosa c_0)
`A      1234ABCD`
`+ B    A1B2C3F4`
`S =   0B3E76FC1`

`C =   111100000` (*LSB* bit je ulazni bit prenosa c_0)
`A      A00C00DE`
`+ B    7FFFE000`
`S =   1200BE0DE`

c) Slično kao u tačkama **a** i **b**, operacije sabiranja u oktalnom brojnom sistemu će izgledati:

```
    C =     001111100 (LSB bit je ulazni bit prenosa c0)
    A        12347650
+   B        33445566
    S =     046015436

    C =     111111110 (LSB bit je ulazni bit prenosa c0)
    A        77665544
+   B        11223344
    S =     111111110
```

Treba napomenuti da svi biti prenosa mogu imati vrednosti 0 ili 1, bez obzira na osnovu brojnog sistema u kome se definisana operacija sabiranja izvršava.

Zadatak 4

a) Izvršiti operacije oduzimanja sledećih neoznačenih binarnih brojeva $X = 10101011$ i $Y = 00011100$, i odrediti sve bite pozajmice b_i i bite rezultata oduzimanja d_i. Bit $b_{i+1|i+1\neq 0}$ je bit pozajmice koji nastaje prilikom operacije oduzimanja bita na poziciji i, tj. prilikom oduzimanja bita $x_i - y_i - b_i$. Bit b_0 je bit ulazne pozajmice koji se koristi prilikom operacije oduzimanja bita na poziciji 0.

$$
\begin{array}{lr}
 & b_8b_7b_6b_5b_4b_3b_2b_1b_0 = B \\
 & x_7x_6x_5x_4x_3x_2x_1x_0 = X \\
- & y_7y_6y_5y_4y_3y_2y_1y_0 = Y \\
\hline
 & d_7d_6d_5d_4d_3d_2d_1d_0 = D
\end{array}
$$

b) Izvršiti oduzimanje sledećih brojeva datih u heksadecimalnom brojnom sistemu i odrediti sve cifre pozajmice b_i i bite rezultata oduzimanja d_i.

$$A234ABCD - 11B2C3F4$$

c) Izvršiti oduzimanje sledećih brojeva datih u oktalnom brojnom sistemu i odrediti sve cifre pozajmice b_i i bite rezultata oduzimanja d_i.

$$62347650 - 33445566$$

REŠENJE:

a) Prilikom operacije oduzimanja $X - Y$, brojeva X i Y datih u binarnom obliku ($X = x_n \dots x_i \dots x_0$ i $Y = y_n \dots y_i \dots y_0$), bit pozajmice na poziciji i se generiše ($b_{i+1} = 1$), ukoliko je vrednost cifre $x_i < y_i + b_i$. Generisani bit pozajmice b_{i+1}, slično kao u slučaju određivanja bita prenosa kod operacije sabiranja, može imati vrednost 0 ili 1. Ukoliko se generiše bit pozajmice ($b_{i+1} = 1$), operacija oduzimanja na poziciji i se svodi na operaciju oduzimanja u obliku:

$$d_i = r + x_i - y_i - b_i \tag{4.1}$$

U izrazu 4.1, r predstavlja osnovu brojnog sistema u kome se vrši operacija oduzimanja. Rezultat operacije oduzimanja dva neoznačena broja sa n-cifara u proizvoljnom brojnom sistemu može maksimalno imati n cifara, tako da je predviđen dovoljan broj cifara za predstavljanje rezultata operacije oduzimanja. Prema postavci zadatka ulazni bit pozajmice uzimamo da je jednak nuli, tj. $b_0 = 0$.
Operacija oduzimanja datih brojeva iz postavke zadatka, u binarnom brojnom sistemu, će izgledati:

```
    B =    000111000 (LSB ulazni bit pozajmice b0)
    X       10101011
 -  Y       00011100
    D =     10001111
```

b) Za pojašnjenje operacije oduzimanja brojeva u heksadecimalnom brojnom sistemu pogledati rešenje iz tačke **a**.

```
    B =    001010100 (LSB ulazni bit pozajmice b0)
    X       A234ABCD
 -  Y       11B2C3F4
    D =     9081E7D9
```

c) Slično kao u tačkama **a** i **b**, operacija oduzimanja brojeva datih u oktalnom brojnom sistemu će izgledati:

```
    B =    011000110 (LSB ulazni bit pozajmice b0)
    X       62347650
 -  Y       33445566
    D =     26702062
```

Zadatak 5

a) Ako je na raspolaganju ukupno 6 bita za predstavljanje binarnih brojeva u kodu znak i apsolutna vrednost broja, odrediti opseg decimalnih vrednosti brojeva koje je moguće predstaviti na ovaj način.
b) Odrediti decimalnu vrednost sledećih binarnih brojeva datih u kodu znak i apsolutna vrednost:

$$A = 10101,\ B = 00011,\ C = 10001,\ D = 10000$$

c) Izvršiti sledeće aritmetičke operacije nad brojevima $A = 101011$ i $B = 010101$ koji su dati u kodu znak i apsolutna vrednost broja a rezultat operacije predstaviti sa minimalnim brojem bita:

$$A - B,\ B - A,\ A + 2B,\ 2B - A$$

Napomena: Za predstavljanje rezultata zadatih aritmetičkih operacija na raspolaganju je proizvoljan broj bita.

REŠENJE:

a) U kodu znak i apsolutna vrednost broja, MSB bit predstavlja znak binarnog broja, tako da je za predstavljanje apsolutne vrednosti broja na raspolaganju ukupno $n-1$ bita. Opseg decimalne vrednosti n-to bitnog binarnog broja, datog u predstavi znak i apsolutna vrednost, dat je izrazom:

$$-(2^{n-1}-1) \leq DZN + APS \leq 2^{n-1}-1$$

U slučaju šestobitne predstave broja u kodu znak i apsolutna vrednost, opseg decimalnih vrednosti brojeva, koje je moguće u ovom kodu predstaviti, kreće se u opsegu od -31 do 31.

b) $A = 10101 = -5$
$B = 00011 = +3$
$C = 10001 = -1$
$D = 10000 = 0$ (u kodu znak i apsolutna, brojevi 00000 i 10000 imaju decimalnu vrednost 0)

c) Prilikom operacija sabiranja i oduzimanja brojeva u kodu znak i apsolutna vrednost, znak rezultata se određuje na osnovu apsolutne vrednosti brojeva i operacije koja se nad brojevima izvršava. Dakle, za različite apsolutne vrednosti datih brojeva, na različit način se određuje rezultat operacije.
Ako je broj $A < 0$ i $B > 0$, rezultat operacije $A - B$ će biti negativan broj, čija je apsolutna vrednost jednaka zbiru apsolutnih vrednosti brojeva A i B.

$$A - B = A + (-B) = -(|A| + |B|) = 1100000$$

Dakle, potrebno je 7-bita za predstavu rezultata operacije.
Sa druge strane, rezultat operacije $B - A$ biće pozitivan broj, odnosno:

$$B - A = +(|A| + |B|) = 0100000$$

Dakle, potrebno je 7-bita za predstavu rezultata operacije.
U slučaju, $A + 2B$, $2B > A$, rezultat operacije sabiranja biće pozitivan, odnosno:

$$A + 2B = +(2|B| - |A|) = 011111$$

Dakle, potrebno je 6-bita za predstavu rezultata operacije.
U slučaju, $2B - A$, imamo da je $2B > A$, pa je rezultat operacije pozitivan, odnosno:

$$2B - A = +(2|B| + |A|) = 0110101$$

Dakle, potrebno je 7-bita za predstavu rezultata operacije.

Zadatak 6

a) Dati su sledeći brojevi u brojnom sistemu sa osnovom 10: 1845, 2067, 0100, 0000, 0007. Predstaviti date brojeve u komplementu osnove sa ukupno četiri cifre.
b) Dati su sledeći brojevi u brojnom sistemu sa osnovom 8: 1445, 2067, 0100, 0000, 0007. Predstaviti date brojeve u komplementu osnove sa ukupno četiri cifre.
c) Odrediti opseg decimalnih vrednosti označenih binarnih brojeva koje je moguće predstaviti sa 6-bita, ako je poznato da su negativni brojevi izraženi u komplementu osnove.

d) Dat je broj D u sistemu sa osnovom r. Polazeći od definicije predstave broja D u komplementu osnove (komplement do r), pokazati da se u broj u komplementu osnove dobija tako što se na broj D dat u komplementu do maksimalne vrednosti (komplement do $r-1$) doda jedinica.

REŠENJE:

a) U brojnom sistemu sa osnovom r, svaka cifra broja D može imati vrednost u opsegu od 0 do $r-1$, tako da je najveći broj (broj sa maksimalnom vrednosti, označen sa D_{max}) koji je moguće predstaviti u sistemu sa osnovom r, sa ukupno n cifara, dat izrazom:

$$D_{max} = r^{n-1} \tag{6.1}$$

Komplement do maksimalne vrednosti, datog broja D, ili komplement do $r-1$ (označen sa D_{r-1}), dobija se tako što se od broja sa maksimalnom vrednosti koji je dat izrazom 6.1, oduzme broj D. Dakle:

$$D_{r-1} = r^n - 1 - D \tag{6.2}$$

Komplement do $r-1$ broja $D = d_{n-1}d_{n-2}\dots d_1d_0$, ne određuje se aritmetičkom operacijom oduzimanja, koja je definisana izrazom 6.2, već se određuje tako što svaka cifra broja u komplementu do $r-1$ ima vrednost $r-1-d_i$. U sistemu sa osnovom r, najveća vrednost svake cifre je upravo $r-1$.
U sistemu sa osnovom r, n-to bitni broj $D = d_{n-1}d_{n-2}\dots d_1d_0$ u komplementu osnove (komplementu do r ili komplementu r) koji ćemo označiti sa D_r, dat je prema definiciji komplementa osnove, u obliku:

$$D_r = r^n - D \tag{6.3}$$

Polazeći od izraza 6.3, proširenjem se dobija:

$$D_r = (r^n - 1 - D) + 1 = D_{r-1} + 1 \tag{6.4}$$

Na osnovu izraza 6.2 i 6.4, zaključujemo, do se do broja D u komplementu osnove, dolazi tako što se se na komplement do maksimalne vrednosti, odnosno D_{r-1}, doda jedinica.
U komplementu osnove, dati decimalni brojevi imaće vrednosti:

$$\begin{gathered}(9999-1845)+1 = 8154+1 = 8155_{10},\\ (9999-2067)+1 = 7932+1 = 7933_{10},\\ (9999-0100)+1 = 9899+1 = 9900_{10},\\ (9999-0000)+1 = 9999+1 = 10000 = 0000_{10}\end{gathered}$$

U poslednjem slučaju ignoriše se vodeća jedinica rezultata (10000) pa je broj 0 u komplementu osnove jednak 0. Dakle, u slučaju da prilikom dodavanja jedinice na broj u komplementu do $r-1$, dođe do prekoračenja broja cifara rezultata, broj u komplementu do r se dobija tako što se odbacuje vodeća jedinica rezultata. Ova situacija se javlja u slučaju predstavljanja broja 0 u komplementu osnove.
Za broj 7 u brojnom sistemu sa osnovom 10, u komplementu osnove biće dat u obliku:

$$(9999-0007)+1 = 9992+1 = 9993_{10}$$

b) Slično kao u tački **a**, dati oktalni brojevi u komplementu osnove imaće vrednosti:

$$6333,\ 5711,\ 7700,\ 0000,\ 7771.$$

c) Sa ukupno 6 bita moguće je predstaviti binarne brojeve u opsegu od 000000 do 111111. Da bismo odredili opseg decimalnih vrednosti binarnih brojeva koji se predstavljaju, najpre zaključujemo da su svi binarni brojevi, kod kojih je $MSB = 1$, negativni i dati u komplementu do dva. Najveći pozitivni broj koji je moguće predstaviti je broj 011111 u binarnom obliku, odnosno decimalni broj 31. U slučaju negativnih brojeva, situacija je drugačija, odnosno, najmanji negativni broj koji je moguće predstaviti u komplementu do dva sa ukupno 6 bita je binarni broj 100000, odnosno -32 decimalno. Dakle sa ukupno n bita, na ovaj način, je moguće predstaviti brojeve u opsegu od $2^{n-1} - 1$ do -2^{n-1}.

d) Pogledati rešenje iz tačke **a**.

Zadatak 7

a) Sledeće negativne decimalne brojeve predstaviti u binarnom obliku u drugom komplementu sa ukupno osam bita:

$$-17,\ -119,\ -0,\ -1,\ -128$$

b) Dati su trobitni označeni binarni brojevi D u drugom komplementu. Odrediti brojeve $-D$ u drugom komplementu sa osam bita.

$$011,\ 101,\ 100,\ 111,\ 000$$

REŠENJE:

a) Decimalni broj 17 u binarnom obliku sa 8 bita dat je u obliku 00010001. Broj -17, prema postavci zadatka, predstavljen u drugom komplementu izgledaće:

$$-17 = -00010001 = 11101110 + 1 = 11101111_2$$

Na sličan način, dobijamo:

$$-119 = -01110111 = 10001000 + 1 = 10001001_2$$
$$-0 = -00000000 = 11111111 + 1 = 1000000002 = 00000000_2$$

Ignoriše se vodeća jedinica, pogledati zadatak 6.

$$-1 = -00000001 = 11111110 + 1 = 11111111_2$$
$$-128 = -10000000 = 01111111 + 1 = 10000000_2$$

b) Neka je dat m-to bitni binarni broj u drugom komplementu u obliku $D = d_{m-1}d_{m-2}\dots d_1d_0$, isti broj u n-to bitnom zapisu ($n > m$) dobija se tako što vodećih $n - m$ bita ima vrednost bita znaka (bita d_{m-1}) polaznog broja D. Dakle, broj D u n-to bitnom zapisu biće dat u obliku:

$$D = d_{m-1}d_{m-1}\dots d_{m-1}d_{m-2}\dots d_1d_0$$

Da bi smo odredili binarne brojeve u traženom obliku, najpre ćemo izvršiti proširenje broja bita i zatim prebacivanje dobijenih brojeva u drugi komplement.

Za brojeve date u postavci zadatka dobijamo:
$-011 = -00000011 = 11111100 + 1 = 11111101$ (dopisuju se nule u slučaju pozitivnog broja, $011 > 0$)
$-101 = -11111101 = 00000010 + 1 = 00000011$ (dopisuju se jedinice u slučaju negativnog broja)
Na sličan način dobijamo:
$-100 = -11111100 = 00000011 + 1 = 00000100$

Na prvi pogled, kao rezultat operacije komplementiranja broja 100 dobija se broj u istom obliku, međutim, polazni broj -100 predstavljen sa osam bita dat je u obliku -11111100, dok je dobijeni broj u komplementu do dva dat u obliku 00000100. Treba takođe imati u vidu da broj 100 u drugom komplementu ima decimalnu vrednost -4, dok broj 4 nije moguće predstaviti u drugom komplementu sa ukupno tri bita. Broj 4 je moguće predstaviti u drugom komplementu sa minimalno 4 bita i tada je broj 4 dat u obliku 0100.
$-111 = -11111111 = 00000000 + 1 = 00000001$
$-000 = -00000000 = 11111111 + 1 = 100000000 = 00000000$
Ignoriše se vodeća jedinica, pogledati zadatak 6, tačku pod **a**.

Zadatak 8

a) Sledeće neoznačene binarne brojeve predstaviti u prvom komplementu (komplementu do jedan), sa ukupno četiri bita: 101, 011, 1100, 0.
b) Sledeće neoznačene decimalne brojeve predstaviti u komplementu do devet sa četiri cifre: 100, 0, 987, 789.

REŠENJE:

a) Broj D dat u osnovi r, u komplementu $r-1$, dobija se na osnovu izraza:

$$D_{r-1} = r^n - 1 - D \tag{8.1}$$

Na osnovu izraza 8.1, predstava datih binarnih brojeva u komplementu do jedan dobija se kada se dati binarni broj oduzme od $2^n - 1$, gde je n broj bita koji je na raspolaganju za predstavljanje binarnog broja (pogledati zadatak 6, tačku pod **a**). Obzirom da se radi o neoznačenim binarnim brojevima proširenje broja bita svodi se na dopisivanje odgovarajućeg broja vodećih nula.
Za binarni broj 101 imamo:

$$1111 - 0101 = 1010_{KMV}$$

Slično, za binarni broj 011 imamo:

$$1111 - 0101 = 1100_{KMV}$$

Za binarni broj 1100 imamo:

$$1111 - 1100 = 0011_{KMV}$$

Za binarni broj 0 imamo:

$$1111 - 0000 = 1111_{KMV}$$

b) Predstava decimalnog broja D u komplementu do devet sa četiri cifre, dobija se kada se decimalni broj D oduzme od maksimalnog decimalnog broja kojeg je moguće predstaviti sa četiri cifre, tj. broja $10^4 - 1$. Slično kao u tački **a**, obzirom da se radi o neoznačenim decimalnim brojevima proširenje broja cifara svodi se na dopisivanje odgovarajućeg broja vodećih nula.

Za decimalni broj 100 imamo:

$$9999 - 0100 = 9899_{KMV}$$

Za decimalni broj 0 imamo:

$$9999 - 0000 = 9999_{KMV}$$

Za decimalni broj 987 imamo:

$$9999 - 0987 = 9012_{KMV}$$

Za decimalni broj 789 imamo:

$$9999 - 789 = 9210_{KMV}$$

Zadatak 9

Izvršiti sledeće operacije nad četvorobitnim binarnim brojevima datim u komplementu do dva. Za prikazivanje rezultata operacije na raspolaganju je ukupno 4 bita. Odrediti da li prilikom izvršavanja operacije dolazi do prekoračenja (*overflow*) i u slučaju prekoračenja naznačiti $OF = 1$.

a) $1101 + 1010$, $1000 + 1000$, $0101 + 0110$, $0111 + 0111$, $0111 + 1001$.
b) $0100 - 0011$, $0011 - 0100$, $0011 - 1100$, $1101 - 1100$.

REŠENJE:

a) Binarni broj dat u komplementu do dva je označen broj, pa se prilikom operacije sabiranja brojeva u komplementu do dva, prekoračenje broja bita rezultata sabiranja može javiti, u slučaju kada su biti na poziciji znaka oba sabirka međusobno isti, ali različiti od bita na poziciji znaka rezultata sabiranja. Prekoračenje je moguće detektovati i posmatranjem prenosa koji se javlja prilikom sabiranja na poziciji bita znaka, odnosno, do prekoračenja dolazi kada je ulazni bit prenosa, prilikom operacije sabiranja na poziciji znaka, različit od bita izlaznog prenosa.

Operacija oduzimanja $(A - B)$ se u slučaju brojeva datih u komplementu do dva standardno obavlja preko operacije sabiranja, pri čemu se umanjilac predstavlja u komplementu do dva ($-B$ se predstavlja u komplementu do dva). Prekoračenje se dalje određuje na isti način kao kod operacije sabiranja.

Za date vrednosti brojeva, 1101 i 1010, iz postavke zadatka, rezultat operacije sabiranja i vrednost bita prekoračenja (OF), određeni su sa:

$$\begin{array}{lr} & 10000 \\ & 1101_2 \\ + & 1010_2 \\ \hline & \cancel{1}0111_2 \end{array} \quad \texttt{OF=1}$$

Na sličan način određujemo ostale rezultate operacije sabiranja:

$$\begin{array}{lr} & 10000 \\ & 1000_2 \\ + & 1000_2 \\ \hline & \cancel{1}0000_2 \end{array} \quad \texttt{OF=1}$$

$$\begin{array}{lr} & 01000 \\ & 0101_2 \\ + & 0110_2 \\ \hline & \cancel{0}1011_2 \end{array} \quad \texttt{OF=1}$$

$$\begin{array}{lr} & 01110 \\ & 0111_2 \\ + & 0111_2 \\ \hline & \cancel{0}1110_2 \end{array} \quad \texttt{OF=1}$$

$$\begin{array}{lr} & 11110 \\ & 0111_2 \\ + & 1001_2 \\ \hline & \cancel{1}0000_2 \end{array} \quad \texttt{OF=0}$$

b) Rezultat operacije oduzimanja, $A - B$, izračunavamo preko operaciju sabiranja kao $A + (-B)$. Za date binarne brojeve iz postavke zadatka, imaćemo:

$$\begin{array}{lr} & \\ & 0100_2 \\ - & 0011_2 \\ \hline & \end{array} \quad \begin{array}{lr} & 11000 \\ & 0100_2 \\ + & 1101_2 \\ \hline & \cancel{1}0001_2 \end{array} \quad \texttt{OF=0}$$

$$\begin{array}{lr} & \\ & 0011_2 \\ - & 0100_2 \\ \hline & \end{array} \quad \begin{array}{lr} & 00110 \\ & 0011_2 \\ + & 1100_2 \\ \hline & \cancel{0}1111_2 \end{array} \quad \texttt{OF=0}$$

$$\begin{array}{lr} & \\ & 0011_2 \\ - & 1100_2 \\ \hline & \end{array} \quad \begin{array}{lr} & 00110 \\ & 0011_2 \\ + & 0100_2 \\ \hline & \cancel{0}0111_2 \end{array} \quad \texttt{OF=0}$$

$$\begin{array}{lr} & \\ & 1101_2 \\ - & 1100_2 \\ \hline & \end{array} \quad \begin{array}{lr} & 11000 \\ & 1101_2 \\ + & 0100_2 \\ \hline & \cancel{1}0001_2 \end{array} \quad \texttt{OF=0}$$

Zadatak 10

Izvršiti sledeće operacije nad četvorobitnim binarnim brojevima datim u komplementu do jedan. Odrediti da li dolazi do prekoračenja (*overflow*) prilikom izvršavanja operacije. U slučaju prekoračenja naznačiti $OF = 1$.

a) $1101 + 1010$, $1000 + 1000$, $0101 + 0110$, $0111 + 0111$.
b) $0100 - 0011$, $0011 - 0100$, $0011 - 1100$, $1101 - 1100$.

REŠENJE:

a) Operacija sabiranja u komplementu do jedan, izvršava se na sličan način kao i kod operacije sabiranja brojeva datih u drugom komplementu. U slučaju sabiranja dva n-to bitna broja, rezultat sabiranja na poziciji $n + 1$ (izlazni prenos sabiranja bita na poziciji n ili izlazni prenos sabiranja na poziciji znaka), dodaje se na rezultat sabiranja i tek nakon izvršene operacije vrši se provera da li je prilikom sabiranja došlo do prekoračenja opsega. Prekoračenje se generiše na isti način kao kod operacije sabiranja binarnih brojeva u drugom komplementu.
Operacija sabiranja datih binarnih brojeva iz postavke zadataka izgledaće:

$$\begin{array}{lr l}
 & \texttt{10000} & \\
 & \texttt{1101}_2 & \\
\texttt{+} & \texttt{1010}_2 & \\
\hline
 & \texttt{10111}_2 & \\
\texttt{+} & \hookrightarrow\ \texttt{1}_2 & \\
\hline
 & \texttt{1000}_2 & \texttt{OF=0}
\end{array}$$

Izlazni prenos sa pozicije znaka se dodaje na rezultat operacije, tako da je rezultat operacije 1000. Obzirom da biti na poziciji znaka oba sabirka isti kao i rezultata sabiranja, prilikom operacije nije došlo do prekoračenja opsega.
Na sličan način dobijamo:

$$\begin{array}{lr l}
 & \texttt{10000} & \\
 & \texttt{1000}_2 & \\
\texttt{+} & \texttt{1000}_2 & \\
\hline
 & \texttt{10000}_2 & \\
\texttt{+} & \hookrightarrow\ \texttt{1}_2 & \\
\hline
 & \texttt{0001}_2 & \texttt{OF=1}
\end{array}$$

Izlazni prenos sa pozicije bita znaka se dodaje na rezultat operacije, tako da je rezultat operacije binarni broj 0001. Kako je bit na poziciji bita znaka rezultata sabiranja različit od bita znaka sabiraka, prilikom izvšavanja aritmetičke operacije došlo je do prekoračenja.

$$\begin{array}{lr l}
 & \texttt{01000} & \\
 & \texttt{0101}_2 & \\
\texttt{+} & \texttt{0110}_2 & \\
\hline
 & \texttt{01011}_2 & \\
\texttt{+} & \hookrightarrow\ \texttt{0}_2 & \\
\hline
 & \texttt{1011}_2 & \texttt{OF=1}
\end{array}$$

$$
\begin{array}{lr}
 & \mathtt{01110} \\
 & \mathtt{0111}_2 \\
\mathtt{+} & \mathtt{0111}_2 \\
\hline
 & \mathtt{01110}_2 \\
\mathtt{+} & \hookrightarrow \ \mathtt{0}_2 \\
\hline
 & \mathtt{1110}_2 \quad \mathtt{OF=1}
\end{array}
$$

b) Operacija oduzimanja binarnih brojeva u prvom komplementu, izvršava se na isti način kao kod operacije oduzimanja brojeva u drugom komplementu. Dakle, operacija $A-B$, izvršava se kao $A+(-B)$, gde se broj $-B$ u prvom komplementu jednostavno određuje prostim komplementiranjem svih bita izvornog binarnog broja B.
Operacija oduzimanja datih binarnih brojeva iz postavke zadataka izgledaće:

$$
\begin{array}{lrclr}
 & & & & \mathtt{11000} \\
 & \mathtt{0100}_2 & & & \mathtt{0100}_2 \\
\mathtt{-} & \mathtt{0011}_2 & \rightarrow & \mathtt{+} & \mathtt{1100}_2 \\
\cline{1-2} \cline{4-5}
 & & & & \mathtt{10000}_2 \\
 & & & \mathtt{+} & \hookrightarrow \ \mathtt{1}_2 \\
\cline{4-5}
 & & & & \mathtt{0001}_2 \quad \mathtt{OF=0}
\end{array}
$$

Izlazni prenos sa pozicije bita znaka se dodaje na rezultat operacije, tako da je rezultat operacije binarni broj 0001.
Na sličan način dobijamo:

$$
\begin{array}{lrclr}
 & & & & \mathtt{00110} \\
 & \mathtt{0011}_2 & & & \mathtt{0011}_2 \\
\mathtt{-} & \mathtt{0100}_2 & \rightarrow & \mathtt{+} & \mathtt{1011}_2 \\
\cline{1-2} \cline{4-5}
 & & & & \mathtt{01110}_2 \\
 & & & \mathtt{+} & \hookrightarrow \ \mathtt{0}_2 \\
\cline{4-5}
 & & & & \mathtt{1110}_2 \quad \mathtt{OF=0}
\end{array}
$$

$$
\begin{array}{lrclr}
 & & & & \mathtt{00110} \\
 & \mathtt{0011}_2 & & & \mathtt{0011}_2 \\
\mathtt{-} & \mathtt{1100}_2 & \rightarrow & \mathtt{+} & \mathtt{0011}_2 \\
\cline{1-2} \cline{4-5}
 & & & & \mathtt{00110}_2 \\
 & & & \mathtt{+} & \hookrightarrow \ \mathtt{0}_2 \\
\cline{4-5}
 & & & & \mathtt{0110}_2 \quad \mathtt{OF=0}
\end{array}
$$

$$
\begin{array}{lrclr}
 & & & & \mathtt{00110} \\
 & \mathtt{1101}_2 & & & \mathtt{1101}_2 \\
\mathtt{-} & \mathtt{1100}_2 & \rightarrow & \mathtt{+} & \mathtt{0011}_2 \\
\cline{1-2} \cline{4-5}
 & & & & \mathtt{10000}_2 \\
 & & & \mathtt{+} & \hookrightarrow \ \mathtt{1}_2 \\
\cline{4-5}
 & & & & \mathtt{0001}_2 \quad \mathtt{OF=0}
\end{array}
$$

Izlazni prenos sa pozicije bita znaka se dodaje na rezultat operacije, tako da je rezultat operacije binarni broj 0001.

Zadatak 11

a) Izvršiti operaciju množenja dva neoznačena četvorobitna binarna broja na najmanje dva načina. Prikazati svaki korak, koji se izvršava prilikom operacije množenja.

1011×1101, 0011×1001.

b) Izvršiti operaciju množenja dva četvorobitna binarana broja data u komplementu do dva. Prikazati svaki korak operacije množenja.
1011×1101, 0011×1001.

REŠENJE:

a) Prvi način množenja neoznačenih binarnih brojeva zahteva najpre formiranje svih parcijalnih proizvoda koji predstavljaju množenik (za bit množioca jednak jedinici) ili vrednost 0 (za bit množioca jednak nuli). Parcijalni proizvodi su pomereni u levo za vrednost pozicije onog bita množioca, koji se koristi u formiranju parcijalnog proizvoda. Ovakav način množenja nije pogodan za računarsku primenu obzirom da zahteva najpre određivanje svih parcijalnih proizvoda, a samim tim i odgovarajuće memorijske resurse, i zatim operaciju sabiranja svih predhodno određenih parcijalnih proizvoda. Rezultat operacije množenja n-to bitnog i m-to bitnog binarnog broja može se uvek predstaviti sa maksimalno $n+m$ bita.
Za binarne brojeve iz postavke zadatka, operacija množenja izgledaće:

```
      1011 × 1101 = 10001111
      1011
     0000
    1011
+  1011
   10001111

      0011 × 1001 = 00011011
      0011
     0000
    0000
+  0011
   00011011
```

Drugi način množenja neoznačenih binarnih brojeva, podrazumeva formiranje parcijalnih proizvoda na sličan način. Razlika je u odnosu na prvi način množenja ogleda se u činjenici da se vrednost parcijalnog proizvoda u svakom koraku dodaje na vrednost proizvoda sabiranja. Postupak množenja podrazumeva da se najpre odredi parcijalni proizvod koji odgovara *LSB* bitu množioca, itd. Početna vrednost rezultata množenja postavlja se na vrednost 0000.
Za binarne brojeve iz postavke zadatka, operacija množenja, korak po korak, izgledaće:

```
        1011 × 1101
        0000
+       1011
       01011
+      0000
      001011
+     1011
     0110111
+    1011
    10001111
```

```
          0011 × 1001
          0000
+         0011
         00011
+        0000
        000011
+       0000
       0000011
+      0011
      00011011
```

b) Postupak množenja označenih binarnih brojeva, datih u drugom komplementu, je sličan drugom načinu množenja neoznačenih brojeva, osim što treba imati u vidu da svaki parcijalni proizvod predstavlja označen binarni broj. Proširenje broja bita prilikom operacije sabiranja parcijalnih proizvoda vrši se kako ne bi došlo do prekoračenja. Treba voditi računa da se proširenja broja bita svodi na dodavanje bita znaka na *MSB* poziciji. Takođe, prilikom određivanja parcijalnog proizvoda koji odgovara *MSB* bitu množioca, treba voditi računa da ukoliko je *MSB* bit množioca jednak jedinici, za vrednost parcijalnog proizvoda uzima se negativna vrednost množenika predstavljena u drugom komplementu.
Množenje dva četvorobitna binarana broja, iz postavke zadatka, data u komplementu do dva dato je u formi:

```
          1011 × 1101
         00000
+        11011
        111011 (proširuje se broj bita)
+       00000
       1111011 (proširuje se broj bita)
+      11011
      11100111 (proširuje se broj bita)
+     00101      (−1011 u drugom komplementu)
     100001111
```

Obzirom da se prilikom operacije sabiranja ignoriše vodeća jedinica, rezultat operacije množenja biće 00001111 ili decimalni broj 15.
Na sličan način, određujemo rezultat druge operacije množenja iz postavke zadatka.

```
          0011 × 1001
         00000
+        00011
        000011
+       00000
       0000011
+      00000
      00000011
+     11101
      11101011
```

Zadatak 12

Izvršiti sledeće operacije deljenja dva neoznačena binarna broja: 11011001/1011, 110101010/1001. Operaciju deljenja izvršiti korak po korak, odrediti celobrojni rezultat deljenja kao i ostatak deljenja.

REŠENJE:

Operacija deljenja binarnih brojeva obavlja se na isti kao u slučaju deljenja decimalnih brojeva.

```
   11011001 :  1011 = 10011
-  1011        (1) (MSB bit rezultata deljenja)
   0010
   00101       (dopisana 1)
-   0000       (0)
    0101
    01010      (dopisana 0)
-    0000      (0)
     1010
     10100     (dopisana 0)
-     1011     (1)
      1001
      10011    (dopisana 1)
-      1011    (1) (LSB bit rezultata deljenja)
       1000    (ostatak deljenja)
```

Na sličan način dobijamo:

```
   110101010 :  1001 = 101111
-  1001        (1) (MSB bit rezultata deljenja)
   0100
   01000       (dopisana 0)
-   0000       (0)
    1000
    10001      (dopisana 1)
-    1001      (1)
     1000
     10000     (dopisana 0)
-     1001     (1)
      0111
      01111    (dopisana 1)
-      1001    (1)
       0110
       01100   (dopisana 0)
-       1001   (1) (LSB bit rezultata deljenja)
        0011   (ostatak deljenja)
```

Zadatak 13

a) Predstaviti decimalne cifre $0-9$ u BCD[5] kôdu, kôdu više tri, *Gray*-ovom BCD kôdu i BCD 2421 kôdu.
b) Dat je 8-bitni neoznačeni binarni broj u BCD kôdu, kôdu više tri i *Gray*-ovom BCD kôdu. Odrediti težinske koeficijente bita (pozicije bita od 0 do 7).
c) Izvršiti sledeće operacije sabiranja dva neoznačena 8 bitna broja data u BCD kôdu:

$$01011001 + 01111001 \text{ i } 01010111 + 00111001$$

d) Sledeće neoznačene binarne brojeve predstaviti u *Gray*-ovom binarnom kôdu:

01101010, 10101111

REŠENJE:

a) Odgovarajuća predstava decimalnih brojeva u BCD kôdu, kôdu više tri i *Gray*-ovom BCD kôdu data je u tabeli 13.1.

Tabela 13.1: Predstava decimalnih brojeva 0-9 u traženim kodovima iz tačke a

Decimalni broj	BCD (8421)	BCD 2421	Više 3	*Gray* BCD
0	0000	0000	0011	0000
1	0001	0001	0100	0001
2	0010	0010	0101	0011
3	0011	0011	0110	0010
4	0100	0100	0111	0110
5	0101	1011	1000	0111
6	0110	1100	1001	0101
7	0111	1101	1010	0100
8	1000	1110	1011	1100
9	1001	1111	1100	1000

b) Kôd više tri i *Gray*-ov BCD kôd nisu težinski kodovi, već su BCD kôd i BCD 2421 težinski kôdovi. Klasičan BCD kôd je ujedno i BCD 8421 kôd, tako da su težinski koeficijenti odgovarajućih bita BCD kôda, 80, 40, 20, 10, 8, 4, 2, 1, počevši od pozicije MSB bita (težinski koeficijent MSB bita je 80), dok su težinski koeficijenti odgovarajućih bita BCD 2421 kôda 20, 40, 20, 10, 2, 4, 2, 1.
Gray-ov BCD kôd je interesantan obzirom da se dva susedna binarna broja data u *Gray*-ovom BCD kôdu razlikuju u vrednosti samo jednog bita kôda, dok je kôd više tri interesantan jer ima osobinu autokomplementiranja, odnosno prvi komplement binarnog broja datog u kôdu više tri dobija se komplementiranjem svih bita datog broja. Npr. prvi komplement broja 0 (0011 u kôdu više tri) je broj 9 (1100 u kôdu više tri). Ovu osobinu nema standardni BCD 8421 kôd.
c) Operacija sabiranja brojeva datih u BCD kôdu, obavlja se tako što se najpre sabiraju biti najniže BCD cifre (najnižih 4-bita). Treba voditi računa, da u slučaju kada je rezultat operacije sabiranja, na poziciji bilo koje cifre, veći od decimalne vrednosti 9, odnosno 1001, na dobijeni rezultat sabiranja (na posmatranoj poziciji cifre) se dodaje vrednost korekcije, odnosno decimalna vrednost 6 ili 0110 binarno.
Operacija sabiranja 8-bitnih BCD brojeva iz postavke zadatka izgledaće:

[5] BCD - *Binary Coded Decimal Digit*

```
       0101 1001 (59)
+      0111 1001 (79)
          1 0010 rezultat sabiranja je veći od 1001
+         0 0110 dodaje se korekcija 0110
          1 1000 rezultat sabiranja nakon korekcije
       0101
+      0111
       1101 1000 (korekcija + 0110)
+      0110
     1 0011 1000 (138)
```

```
     0101 0111 (57)
+    0011 1001 (39)
        1 0000 (korekcija + 0110)
+       0 0110
        1 0110
     0101
+    0011
     1001 0110 (96)
```

d) Binarni broj u *Gray*-ovom binarnom kôdu dobija se direktno od polaznog binarnog broja na sledeći način:
Ako je dat n-to bitni binarni broj $(b_{n-1} \dots b_0)$, odgovarajući bit u *Gray*-ovom binarnom kôdu na poziciji i biće jednak jedinici, tj. $g_i = 1$ ako važi $b_i \neq b_{i+1}$, u suprotnom $g_i = 0$. Prilikom određivanja vrednosti *MSB* bita u *Gray*-ovom kôdu, tj. bita g_{n-1}, podrazumeva se da je vrednost bita b_n jednaka nuli.
Prema postavci zadatka iz tačke **d**, binarni brojevi u *Gray*-ovom binarnom kôdu biće jednaki:

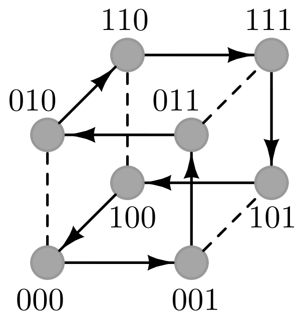
$$01101010_{BIN} \rightarrow 01011111_{GRAY},$$
$$10101111_{BIN} \rightarrow 11111000_{GRAY}.$$

Zadatak 14

a) Za trobitni neoznačeni binarni broj, dat u *Gray*-ovom kôdu, nacrtati odgovarajuću predstavu u obliku 3D kocke sa naznačenim smerom prelaza (naznačiti strelicom) između susednih kodnih reči.
b) Definisati pojam *Hamming*-ovog rastojanja i odrediti *Hamming*-ovo rastojanje kôdnih reči 100 i 001, kao i 001 i 101. Za koju vrstu kôdova je neophodno poznavanje termina *Hamming*-ovo rastojanje.

REŠENJE:

a) Na slici 14.1 prikazani su, u formi 3D kocke, svi prelazi između susednih kodnih reči, datih u Gray-ovom kodu. Temena kocke predstavljaju kodne reči, dok strelice predstavljaju prelaze između dve susedne kodne reči.



Slika 14.1: Prelazi između susednih kôdnih reču u slučaju *Gray*-ovog kôda

b) *Hamming*-ovo rastojanje između dve n-bitne kôdne reči (u binarnoj formi) predstavlja broj pozicija na kojima se odgovarajući biti, kôdnih reči, razlikuju. U slučaju 3D-kocke iz tačke **a**, *Hamming*-ovo rastojanje predstavlja najkraću putanju (broj prelaza) između dve kodne reči. *Gray*-ov kôd je takav kôd za koji važi da je *Hamming*-ovo rastojanje između susednih kôdnih reči jednako 1. Dakle, u slučaju *Gray*-ovog kôda nemamo mogućnost detekcije ili korekcije grešaka, osim ako se sukcesivno šalju informacije koje su rezultat inkrementiranja ili dekrementiranja. U ovom slučaju se može detektovati greška kao situacija kada se dve uzastopno primljene kôdne reči razlikuju u više od jednog bita.
Za slučaj proizvoljnog kôda, *Hamming*-ovo rastojanje između kôdnih reči 100 i 001 je 2, dok je između kôdnih reči 001 i 101 jednako 1. *Hamming*-ovo rastojanje je pojam koji se koristi kod kôdova koji imaju mogućnost detekcije i korekcije grešaka.

<u>Zadatak 15</u>

a) Dat je 3-bitni binarni broj D. Predstaviti dati broj D u kôdu sa neparnom i parnom parnosti. Koliki broj bita ima kôdovana informacija?
b) Odrediti minimalno *Hamming*-ovo rastojanje između dve kôdne reči u slučaju kôda sa neparnom ili parnom parnosti iz tačke **a**.
c) Da li kôd sa neparnom parnosti ima mogućnost detekcije i/ili korekcije greške? Obrazložiti odgovor.

REŠENJE:

a) Kôdovana informacija ima ukupno 4 bita, dakle na tri informaciona bita, koja predstavljaju binarni broj D, dodaje se jedan kontrolni bit koji se naziva bit parnosti. Bit parnosti može biti takav, da ukupan broj jedinica kod kôdovane informacije (zajedno se posmatraju informacioni i kontrolni biti) bude paran (dobija se kôd sa parnom parnosti, kontrolni bit se naziva bit parne parnosti), ili neparan (dobija se kôd sa neparnom parnosti, kontrolni bit se naziva bit neparne parnosti). Kôd sa neparnom ili parnosti se može odrediti za proizvoljan broj informacionih bita.
Izgled kôdnih reči za određenu vrednost binarnog broja D data je u tabeli 15.1. Biti $d_2d_1d_0$ predstavljaju informacione bite, dok je bit c_0 kontrolni bit ili bit parnosti.

Tabela 15.1: Izgled kôdnih reči za različite vrednosti binarnog broja D

Binarni broj D $d_2d_1d_0$	Kôd sa parnom parnosti $\overline{d_2d_1d_0c_0}$	Kôd sa neparnom parnosti $\overline{d_2d_1d_0c_0}$
000	000**0**	000**1**
001	001**1**	001**0**
010	010**1**	010**0**
011	011**0**	011**1**
100	100**1**	100**0**
101	101**0**	101**1**
110	110**0**	110**1**
111	111**1**	111**0**

Tabelom 15.1 definisane su **kôdne reči** za slučaj kôda sa parnom i neparnom parnosti. Treba napomenuti da sve kombinacije bita $d_2d_1d_0c_0$ koje nisu obuhvaćene tabelom 15.1, predstavljaju **nekodne reči**.

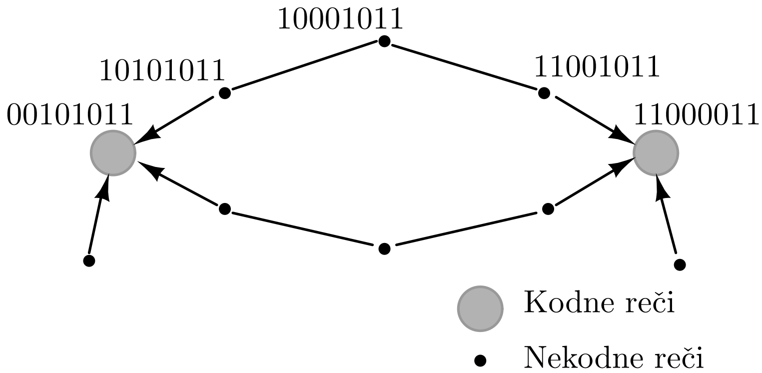
b) Minimalno rastojanje između kôdnih reči kod kôda sa parnom parnosti i neparnom parnosti je 2, što znači da su oba kôda kôdovi koji mogu da detektuju proizvoljnu jednobitnu grešku.

c) Interesantno je da kôd sa parnom ili neparnom parnosti ima mogućnost detekcije neparnog broja grešaka, odnosno detektuje se pojavljivanje $1, 3, 5 \ldots$ grešaka.
Dakle, kôdovi sa bitom parnosti nemaju mogućnost korekcije greške obzirom da je minimalno *Hamming*-ovo rastojanje između proizvoljnih kodnih reči 2. U opštem slučaju, da bi kôd imao mogućnost korekcije bar jedne greške, neophodno je da minimalno *Hamming*-ovo rastojanje između kôdnih reči, bude veće ili jednako 3.

Zadatak 16

Na slici 16.1 dat je fragment n-dimenzione kocke sa naznačenim kôdnim i nekôdnim rečima. Minimalno *Hamming*-ovo rastojanje između kôdnih reči kôda, čiji je fragment prikazan, je 4.

a) Odrediti mogućnost korekcije i detekcije 1-bitnih grešaka.
b) Odrediti mogućnost korekcije i detekcije 2-bitnih grešaka.
c) Odrediti mogućnost korekcije i detekcije 3-bitnih grešaka.



Slika 16.1: Fragment n-dimenzione kocke sa naznačenim kôdnim i nekôdnim rečima

REŠENJE:

a, b, c) U opštem slučaju, minimalno *Hamming*-ovo rastojanje određuje broj grešaka koje je moguće detektovati, odnosno korigovati prema izrazu:

$$\text{Minimalno } Hamming - \text{ovo rastojanje} = 2c + d + 1 \tag{16.1}$$

U izrazu 16.1, c je broj pogrešnih bita u kôdu koje je moguće korigovati, dok je d dodatni broj bita greške koje je moguće detektovati. Obzirom da za bite greške, koje je moguće korigovati, očigledno važi, da ih je moguće i detektovati, ukupan broj bita koje je moguće detektovati jednak je zbiru $d + c$.

U našem slučaju, minimalno *Hamming*-ovo rastojanje između proizvoljnih kôdnih reči iznosi 4, odnosno prema jednačini 16.1 imamo:

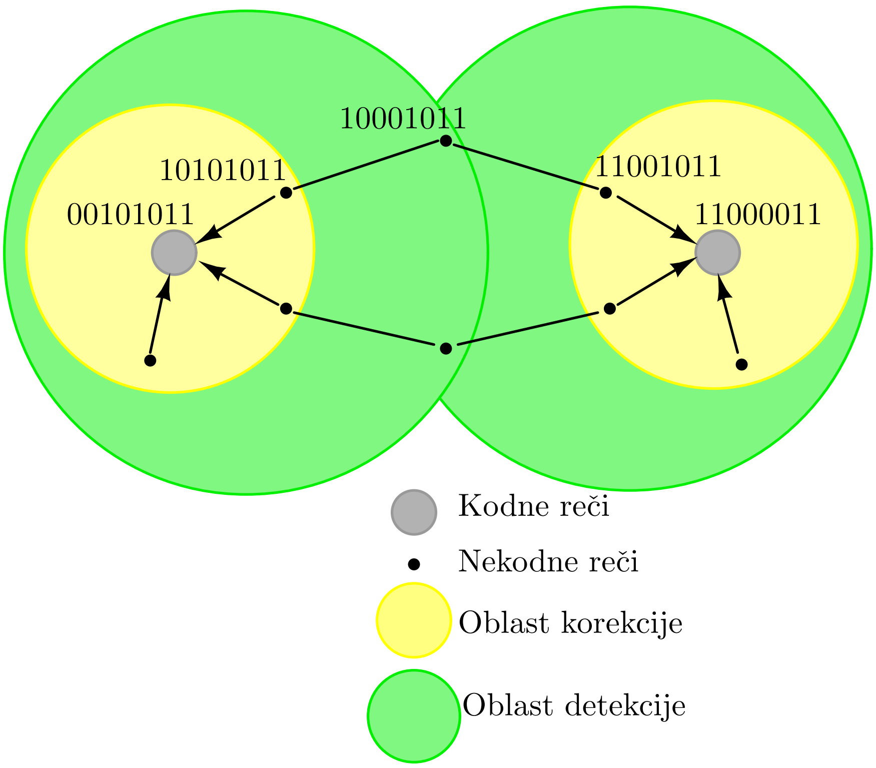
$$4 = 2c + d + 1 \tag{16.2}$$

U izrazu 16.2, c predstavlja broj bita greške koju je moguće korigovati, dok je d dodatni broj bita koje je moguće detektovati.

To bi značilo, da u slučaju datog kôda, možemo korigovati najviše 1-bitnu grešku i detektovati proizvoljnu 2-bitnu grešku. Mogućnost korekcije 2-bitne greške (slučaj npr. nekôdne reči 10001011 sa slike 16.1) na najbližu kôdnu reč može dovesti do greške u korekciji, obzirom da, postoje dve kôdne reči koje su podjednako udaljene.

Takođe kôd može detektovati 3-bitnu grešku, međutim greška će biti protumačena kao jednobitna greška koja će biti korigovana na pogrešnu vrednost. Srećom, verovatnoća pojave trobitnih grešaka, je najčešće, zanemarljiva u odnosu na verovatnoću pojave jednobitnih grešaka.

Na slici 16.2, dat je segment kôda sa obeleženim oblastima korekcije i detekcije grešaka, koje definišu nekôdne reči, tj. greške, koje je moguće detektovati, odnosno korigovati.



Slika 16.2: Fragment n-dimenzione kocke sa naznačenim oblastima korekcije i detekcije grešaka

---

Zadatak 17

a) Dat je 4-bitni binarni broj, odrediti *Hamming*-ov kôd sa minimalnim *Hamming*-ovim rastojanjem 3.
b) Ako je primljena sekvenca bita $d_7d_6d_5d_3c_4c_2c_1 = 0101011$, gde su d_i informacioni biti a c_i kontrolni biti sekvence date u *Hamming*-ovom kôdu sa rastojanjem 3, izvršiti korekciju greške u prijemu ako je poznato da je samo jedan bit pogrešan.
c) Odrediti *Hamming*-ov kôd sa minimalnim rastojanjem 4 u slučaju prenosa 4-bitnog podatka.
d) Koliko je maksimalni broj informacionih bita u kôdnoj reči datoj u *Hamming*-ovom kôdu sa minimalnim *Hamming*-ovim rastojanjem 3, ako imamo ukupno 5 kontrolnih bita.

REŠENJE:

a) Prema tekstu zadatka, treba obezbediti da minimalno *Hamming*-ovo rastojanje bude 3, tako da je ukupan broj bita u kôdnoj reči *Hamming*-ovog kôda sa minimalnim *Hamming*-ovim rastojanjem 3, dat izrazom $2^i - 1$, gde je i broj kontrolnih bita u *Hamming*-ovom kôdu. Dakle u slučaju *Hamming*-ovog kôda sa minimalnim rastojanjem 3 imamo:

$$2^i \geq i + j \tag{17.1}$$

, gde je i broj kontrolnih a j broj informacionih bita u *Hamming*-ovom kodu.
Za slučaj dat u postavci zadatka, $j = 4$, pa je za $i = 3$, uslov dat izrazom 17.1 zadovoljen, tako da u *Hamming*-ovom kôdu imamo ukupno 3 kontrolna bita.
Bite kôda ćemo numerisani od 1 (*LSB* bit) do je $2^i - 1$ (*MSB* bit). Dakle, u kôdnoj reči imamo 4 informaciona bita (obzirom da je dat ulazni 4-bitni binarni broj) i 3 bita za proveru greške (kontrolni biti). Pozicije kontrolnih bita su 2^0, 2^1, 2^2, dakle 1, 2 i 4. U opšem slučaju imamo i kontrolnih bita i maksimalno $2^i - 1 - i$ informacionih bita.

Tabela 17.1: Izgled kontrolnih bita u slučaju *Hammingovog* kôda sa minimalnim rastojanjem 3 i 4