Binarno kodirani dekadni brojevi

Koriste se radi tačnog zapisa mešovitih brojeva u računarskom sistemu. Princip zapisa je da se svaka dekadna cifra kodira odredjenim binarnim zapisom. Za uspešno kodiranje neophodno je da dužina kodne reči bude bar četiri.

Pri kodiranju treba da bude ispunjen uslov jednoznačnosti, odnosno da sve binarne reči koje ulaze u kod moraju da budu medjusobno različite.

Osobine koje omogućuju jednostavije izvodjenje operacija su:

- Najvećoj dekadnoj cifri (9) treba pridružiti reč koja ima najveću vrednost (posmatrana kao binarni broj).
- Parnim i neparnim dekadnim ciframa treba da odgovaraju parni odnosno neparni binarni brojevi.
- Kod je *komplementaran* ako su kodovi dekadnih cifara a i b za koje važi uslov $a+b=9$ komplementarni (u smislu da su cifre na odgovarajućim pozicijama komplementarne).
- Kod je *težinski* ako je i-toj poziciji kodne reči pridružen broj p_i, tako da za dekadnu cifru q i njenu kodnu reč $y_3y_2y_1y_0$ važi jednakost $q = p_3y_3 + p_2y_2 + p_1y_1 + p_0y_0$

Dekadna cifra	Binarni kod					
	8421	2421	5421	753-6	višak 3	ciklički
0	0000	0000	0000	0000	0011	0001
1	0001	0001	0001	1001	0100	0101
2	0010	0010	0010	0111	0101	0111
3	0011	0011	0011	0010	0110	1111
4	0100	0100	0100	1011	0111	1110
5	0101	1011	1000	0100	1000	1100
6	0110	1100	1001	1101	1001	1000
7	0111	1101	1010	1000	1010	1001
8	1000	1110	1011	0110	1011	1011
9	1001	1111	1100	1111	1100	0011

Tabela 1: Binarni kodovi dekadnih cifara

Grejov kod

Grejov kod dužine $n \geq 0$ je funkcija $G(n,i)$ koja vrši 1-1 preslikavanje celog broja $i \in [0, 2^n - 1]$ pri čemu važi da se binarne reprezentacije $G(n,i)$ i $G(n,i+1)$ razlikuju tačno na jednom mestu.

Karakteristike Grejovog koda su:

- Funkcija koja vrši preslikavanje nije jedinstvena tako da postoji više Grejovih kodova dužine n.

- Jedna od najčešće korišćenih funkcija se može definisati na sledeći način:

  $G(n,i) =_n i \oplus\ _n[i/2]$ gde $n > 0, i \in [0, 2^n - 1]$, $_n i$ označava i zapisano u binarnom sistemu kao neoznačen ceo broj u polju dužine n, a $\oplus$ ekskluzivnu disjunkciju.

Heksadekadna cifra	Binarna vrednost	Grejov kod	Heksadekadna cifra	Binarna vrednost	Grejov kod
0	0000	0000	8	1000	1100
1	0001	0001	9	1001	1101
2	0010	0011	A	1010	1111
3	0011	0010	B	1011	1110
4	0100	0110	C	1100	1010
5	0101	0111	D	1101	1011
6	0110	0101	E	1110	1001
7	0111	0100	F	1111	1000

Tabela 2: Grejov kod dužine 4

- Ista funkcija $G(n,i)$ se može definisati i rekurentno:

$$
\begin{array}{lcll}
G(n+1,i) & = & 0G(n,i) & n>0, i \in [0,\ldots,2^n-1] \\
G(n+1,i) & = & 1G(n,2^{n+1}-1-i) & n>0, i \in [2^n,\ldots,2^{n+1}-1] \\
G(1,0) & = & 0 & \\
G(1,1) & = & 1 &
\end{array}
$$

Heksadekadna cifra	Binarna vrednost	Grejov kod dužine 1	dužine 2	dužine 3	dužine 4
0	0000	0	00	000	0000
1	0001	1	01	001	0001
2	0010		11	011	0011
3	0011		10	010	0010
4	0100			110	0110
5	0101			111	0111
6	0110			101	0101
7	0111			100	0100
8	1000				1100
9	1001				1101
A	1010				1111
B	1011				1110
C	1100				1010
D	1101				1011
E	1110				1001
F	1111				1000

Tabela 3: Grejovi kodovi dužina 1, 2, 3 i 4

Zapis binarno kodiranih dekadnih brojeva

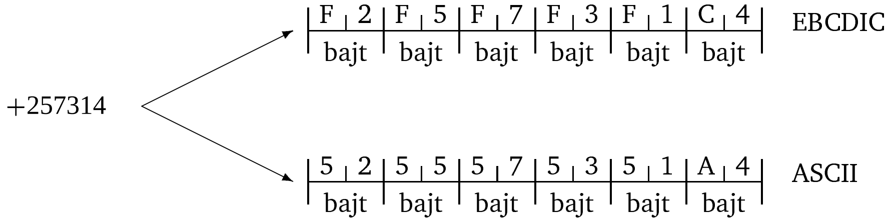
- Binarno kodirani zapis dekadnog broja u nekom kodu se dobija tako što se binarno kodira svaka od njegovih cifara.
- Označeni binarno kodirani dekadni brojevi poseduju dodatnu (dekadnu) cifru u koju se upisuje znak broja. Za zapis označenih brojeva se koriste zapisi:
  - Znak i apsolutna vrednost. Vrednosti cifre za znak broja mogu da budu proizvoljne i zavise od konkretne implementacije na računaru.
  - 10-ti komplement (tj. komplement osnove, N-ti komplement gde je $N = 10$). U ovom slučaju kod najmanje cifre (nule) označava pozitivne, a kod najveće cifre (devetke) negativne brojeve.

BCD zapis dekadnih brojeva

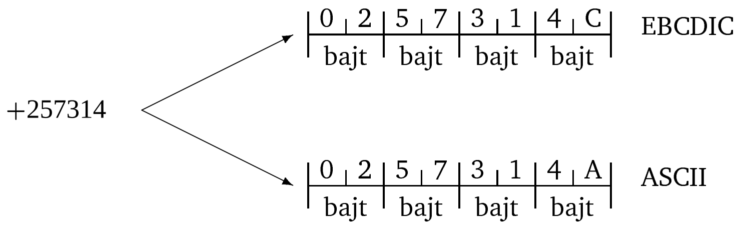
Vodi poreklo od Holeritove kartice kao i termini 'zonsko' i 'cifarsko' ubušenje.

```
                        ____________
                       /
               /  12 / O                  ⟵   red Y (12)
Zonsko        /   11|  O                  ⟵   red X (11)
ubušenje       \/  0|   O                 ⟵   red 0 (10)
               /   1|    O                ⟵   red 1
              /    2|     O               ⟵   red 2
             /     3|      O              ⟵   red 3
cifarsko    /      4|       O             ⟵   red 4
ubušenje    \      5|        O            ⟵   red 5
             \     6|         O           ⟵   red 6
              \    7|          O          ⟵   red 7
               \   8|           O         ⟵   red 8
                \  9|            O        ⟵   red 9
                    |______________
```

Slika 1: Zonsko i cifarsko ubušenje na kartici



Slika 2: Nepakovani (zonski) zapis dekadnog broja



Slika 3: Pakovani (BCD) zapis dekadnog broja

Binarni kod	Heksadekadni kod	Značenje na mestu	
		cifre	znaka
0000	0	0	greška
0001	1	1	greška
0010	2	2	greška
0011	3	3	greška
0100	4	4	greška
0101	5	5	greška
0110	6	6	greška
0111	7	7	greška
1000	8	8	greška
1001	9	9	greška
1010	A	greška	plus
1011	B	greška	minus
1100	C	greška	plus (preporučeno)**
1101	D	greška	minus (preporučeno)**
1110	E	greška	plus
1111	F	greška	plus (zonsko)
**Primedba: ove kodove za znak generišu mašinske instrukcije za rad sa BCD podacima.			

Tabela 4: Cifarski i zonski kodovi u EBCDIC kodu

Decimalna aritmetika

Promena znaka

$$\lfloor 8 \mid 2 \mid 1 \mid 6 \mid 4 \mid C \rfloor \longrightarrow \lfloor 8 \mid 2 \mid 1 \mid 6 \mid 4 \mid D \rfloor$$

Slika 4: Promena znaka dekadnog broja u pakovanom zapisu

Sabiranje i oduzimanje

Neka su A i B dekadni brojevi sa n cifara $A = a_{n-1}a_{n-2}...a_1a_0$ i $B = b_{n-1}b_{n-2}...b_1b_0$, i neka je α funkcija kodiranja koja svakoj cifri u broju pridružuje binarnu kodnu reč

$$\begin{aligned} A\alpha = &\quad \alpha(a_{n-1})\alpha(a_{n-2})...\alpha(a_1)\alpha(a_0) \\ B\alpha = &\quad \alpha(b_{n-1})\alpha(b_{n-2})...\alpha(b_1)\alpha(b_0) \end{aligned}$$

Sabiranje se realizuje u dve faze:

1. Odredi se medjurezultat $C'_\alpha = A\alpha + B\alpha$:

$$\begin{array}{cccccccc} A\alpha & = & \alpha(a_{n-1}) & \alpha(a_{n-2}) & ... & \alpha(a_1) & \alpha(a_0) \\ B\alpha & = & \alpha(b_{n-1}) & \alpha(b_{n-2}) & ... & \alpha(b_1) & \alpha(b_0) \\ \hline C'_\alpha & = & \alpha(c'_{n-1}) & \alpha(c'_{n-2}) & ... & \alpha(c'_1) & \alpha(c'_0) \end{array}$$

2. Dobijeni medjurezultat C'_α se koriguje zbog specifičnosti zapisa binarno kodiranih dekadnih brojeva.

   Konačan rezultat je jednak zbiru medjurezultata i korekcije: $C\alpha = C'_\alpha + K\alpha$:

$$\begin{array}{cccccccc} C'_\alpha & = & \alpha(c'_{n-1}) & \alpha(c'_{n-2}) & ... & \alpha(c'_1) & \alpha(c'_0) \\ K\alpha & = & \alpha(k_{n-1}) & \alpha(k_{n-2}) & ... & \alpha(k_1) & \alpha(k_0) \\ \hline C\alpha & = & \alpha(c_{n-1}) & \alpha(c_{n-2}) & ... & \alpha(c_1) & \alpha(c_0) \end{array}$$

Oduzimanje binarno kodiranih dekadnih brojeva može da se realizuje na dva načina:

1. Po sličnom principu kao i sabiranje, pri čemu se u obe faze umesto sabiranja vrši oduzimanje brojeva.

2. Kao sabiranje brojeva u potpunom komplementu.

Sabiranje i oduzimanje u kodu 8421

Funkcija kodiranja je definisana kao prevodjenje cifre u binarni sistem, tj. $\alpha(c) \longrightarrow c_3c_2c_1c_0$ gde $\forall c \in \{0,1,2,3,4,5,6,7,8,9\}$ važi $c = c_3 * 2^3 + c_2 * 2^2 + c_1 * 2^1 + c_0 * 2^0, c_i \in \{0,1\}, i \in [0,3]$

Prva faza je nezavisna od funkcije kodiranja tako da se primenjuje prethodni algoritam.

U prikazu druge faze se uvode sledeće oznake:

- $\alpha(c_i')$ označava zbir dobijen sabiranjem kodova za dekadno cifarsko mesto i
- p_i' označava binarni prenos izmedju zbirova $\alpha(c_i')$ i $\alpha(c_{i+1}')$ u medjurezultatu prve faze sabiranja
- $\alpha(k_i)$ označava korekciju na dekadnom cifarskom mestu i.
- p_i'' označava binarni prenos u drugoj fazi sabiranja sa dekadnog cifarskog mesta $i-1$ na dekadno cifarsko mesto i. Važi $p_0'' = 0$.

Druga faza se izvodi u n koraka (n maksimum broja cifara dekadnih brojeva koji se sabiraju. Sabiranje se vrši zdesna u levo; postupak u i-tom koraku je sledeći:

1. Odredjuje se privremeni zbir $t_i = \alpha(c_i') + p_i''$.
2. Na osnovu vrednosti t_i i p_i' odredjuje se korekcija $\alpha(k_i)$.
3. Krajnja vrednost $\alpha(c_i)$ se dobija kao zbir $t_i + \alpha(k_i)$. Pri tome se odredjuje i p_{i+1}''.

Korekcija medjurezultata je:

1. $p_{i+1}' = 1 \Rightarrow \alpha(k_i) = (0110)_2$
2. $t_i \geq (1010)_2 \Rightarrow \alpha(k_i) = (0110)_2$.
3. $\alpha(k_i) = (0000)_2$.

Prekoračenje se javlja kada je $p_n' = 1$ ili $p_n'' = 1$.

Oduzimanje se realizuje po sličnom algoritmu kao sabiranje, ili kao sabiranje brojeva u potpunom komplementu.

Primeri

1. Odrediti zbir $A = 18345$ i $B = 9567$ u kodu 8421.

Prva faza			Korak 4	Korak 3	Korak 2	Korak 1	Korak 0
$A\alpha$	=		0001	1000	0011	0100	0101
$B\alpha$	=		0000	1001	0101	0110	0111
p'_0	=						0
p'_1	=					0	
p'_2	=				0		
p'_3	=			0			
p'_4	=		1				
p'_5	=	0					
C'_α	=		0010	0001	1000	1010	1100

Druga faza			Korak 4	Korak 3	Korak 2	Korak 1	Korak 0
C'_α	=		0010	0001	1000	1010	1100
p''_0	=						0
t'_0	=						1100
$\alpha(k_0)$	=						0110
$\alpha(c_0)$	=						0010
p''_1	=					1	
t'_1	=					1011	
$\alpha(k_1)$	=					0110	
$\alpha(c_1)$	=					0001	
p''_2	=				1		
t'_2	=				1001		
$\alpha(k_2)$	=				0000		
$\alpha(c_2)$	=				1001		
p''_3	=			0			
t'_3	=			0001			
$\alpha(k_3)$	=			0110			
$\alpha(c_3)$	=			0111			
p''_4	=		0				
t'_4	=		0010				
$\alpha(k_4)$	=		0000				
$\alpha(c_4)$	=		0010				
p''_5	=	0					
$C\alpha$	=		0010	0111	1001	0001	0010

2. Odrediti zbir $A = 259$ i $B = 938$ u kodu 8421.

A_α	$=$		0000	0000	0010	0101	1001
B_α	$=$		0000	0000	1001	0011	1000
P'	$=$	0	0	0	0	1	0
C'_α	$=$		0000	0000	1011	1001	0001
P''	$=$	0	0	1	0	0	0
K_α	$=$		0000	0000	0110	0000	0110
C_α	$=$		0000	0001	0001	1001	0111

3. Odrediti zbir $A = 9001$ i $B = 999$ u kodu 8421.

A_α	$=$		0000	1001	0000	0000	0001
B_α	$=$		0000	0000	1001	1001	1001
P'	$=$	0	0	0	0	0	0
C'_α	$=$		0000	1001	1001	1001	1010
P''	$=$	0	1	1	1	1	0
K_α	$=$		0000	0110	0110	0110	0110
C_α	$=$		0001	0000	0000	0000	0000

4. Odrediti zbir $A = 99001$ i $B = 999$ u kodu 8421.

$A\alpha$	$=$		1001	1001	0000	0000	0001
$B\alpha$	$=$		0000	0000	1001	1001	1001
P'	$=$	0	0	0	0	0	0
C'_α	$=$		1001	1001	1001	1001	1010
P''	$=$	***1	1	1	1	1	0
$K\alpha$	$=$		0110	0110	0110	0110	0110
$C\alpha$	$=$		0000	0000	0000	0000	0000

Prekoračenje se javlja zbog pojave prenosa $p''_5 = 1$.

5. Odrediti razliku $A = 1275$ i $B = 452$ u kodu 8421.

$B\alpha$	$=$	0000	0000	0100	0101	0010
$[-B\alpha]_{nk}$	$=$	1001	1001	0101	0100	0111
$+1$						0001
$[-B\alpha]_{pk}$	$=$	1001	1001	0101	0100	1000

$C = A + [B]_{pk}$

$A\alpha$	$=$		0000	0001	0010	0111	0101
$[-B\alpha]_{pk}$	$=$		1001	1001	0101	0100	1000
P'	$=$	0	0	0	0	0	0
C'_α	$=$		1001	1010	0111	1011	1101
P''	$=$	1	1	0	1	1	0
$K\alpha$	$=$		0110	0110	0000	0110	0110
$C\alpha$	$=$		0000	0000	1000	0010	0011

U ovom slučaju, u skladu sa pravilima za sabiranje brojeva u potpunom komplementu pojava prenosa $p''_5 = 1$ ne označava prekoračenje.

Sabiranje i oduzimanje u kodu višak 3

Funkcija kodiranja definisana je kao $\alpha(c) \longrightarrow c_3c_2c_1c_0$ gde $\forall c \in \{0,1,2,3,4,5,6,7,8,9\}$ važi $c = c_3 * 2^3 + c_2 * 2^2 + c_1 * 2^1 + c_0 * 2^0 - 3, c_i \in \{0,1\}, i \in [0,3]$

Prva faza je nezavisna od funkcije kodiranja tako da se primenjuje prethodni algoritam.

U kodu *višak 3* druga faza sabiranja se izvodi u n koraka gde je n maksimum broja cifara dekadnih brojeva koji se sabiraju. Sabiranje se vrši zdesna u levo; postupak u i-tom koraku je:

1. Na osnovu vrednosti p'_{i+1} odredjuje se korekcija $\alpha(k_i)$.
2. Krajnja vrednost $\alpha(c_i)$ se dobija kao zbir $\alpha(c'_i) + \alpha(k_i)$. Pojava prekoračenja ('prenosa') u ovoj fazi sabiranja se ingnoriše.

Korekcija:

1. $p'_{i+1} = 1 \Rightarrow \alpha(k_i) = (0011)_2$.
2. $p'_{i+1} = 0 \Rightarrow \alpha(k_i) = (1101)_2$.

Prekoračenje pri sabiranju se javlja kada je $p'_n = 1$.

Oduzimanje u kodu višak 3 se izvodi kao sabiranje u potpunom komplementu.

Primeri

1. Odrediti zbir $A = 18345$ i $B = 9567$ u kodu *višak 3*.

A_α	=		0100	1011	0110	0111	1000
B_α	=		0011	1100	1000	1001	1010
P'	=	0	1	0	1	1	0
C'_α	=		1000	0111	1111	0001	0010
K_α	=		1101	0011	1101	0011	0011
C_α	=		0101	1010	1100	0100	0101

   Dobijeni rezultat sabiranja je broj 27912.

2. Odrediti zbir $A = 99001$ i $B = 999$ u kodu *višak 3*. Pri sabiranju se dobija prekoračenje (označeno sa ***) zbog prenosa na cifarskom mestu najveće težine ($p'_5 = 1$) u prvoj fazi sabiranja.

A_α	=		1100	1100	0011	0011	0100
B_α	=		0011	0011	1100	1100	1100
P'	=	***1	1	1	1	1	0
C'_α	=		0000	0000	0000	0000	0000

3. Odrediti razliku $A = 1275$ i $B = 452$ u kodu *višak 3*. Rezultat $C = A - B = 823$ se dobija primenom sabiranja u potpunom komplementu:

$B\alpha$	$=$		0011	0011	0111	1000	0101
$[-B\alpha]_{nk}$	$=$		1100	1100	1000	0111	1010
$+1$							0001
$[-B\alpha]_{pk}$	$=$		1100	1100	1000	0111	1011

$C = A + [B]_{pk}$							
$A\alpha$	$=$		0011	0100	0101	1010	1000
$[-B\alpha]_{pk}$	$=$		1100	1100	1000	0111	1011
P'	$=$	1	1	0	1	1	0
C'_α	$=$		0000	0000	1110	0010	0011
$K\alpha$	$=$		0011	0011	1101	0011	0011
$C\alpha$	$=$		0011	0011	1011	0101	0110

U skladu sa pravilima za sabiranje brojeva u potpunom komplementu pojava prenosa $p'_5 = 1$ ne označava prekoračenje.

Množenje i deljenje

1. Odrediti proizvod brojeva $A = -38460$ i $B = -321$ u kodu *8421*.

   $A = 000038460D, B = 321D$. Rezultat $C = A * B = 012345660C$ može da se dobije formiranjem delimičnih proizvoda.

```
2 * 000038460
            0
           12
          08
         16
        06
        0
       0
      0
     0
    000076920

2 * 000038460
            0
           12   <--- 2 * 6 = 0C_16 = 12_10
          08    <--- 2 * 4 = 08_16 = 08_10
         16     <--- 2 * 8 = 10_16 = 16_10
        06      <--- 2 * 3 = 06_16 = 06_10
    000076920
```

   Sabiranjem delimičnih dobija se ukupan proizvod:

```
000038460 * 321
000038460
00076920
0115380
012345660
```

2. Odrediti količnik i ostatak brojeva $A = +12345678$ i $B = -321$ u kodu *8421*.

   U pakovanom zapisu ovu operaciju možemo da zapišemo kao
   C= A / B= 012345678C / 321D = 38460D uz ostatak 018C

   Cifre količnika se odredjuju uporedjivanjem delioca sa početnim delom deljenika. Redosled koraka je:

   (a) Upotrebom operacije poredjenja dobijamo

```
012   < 321   da
0123  < 321   da
01234 < 321   ne
```

Primenom operacije množenja dobija se da je prva cifra količnika 3.

(b) Kako je $321 * 3 = 963$, dobijeni proizvod se oduzima od (početka) deljenika i prelazi na odredjivanje naredne cifre.

```
012345678 / 321 = 3
  963
  2715         2715  < 321   ne
```

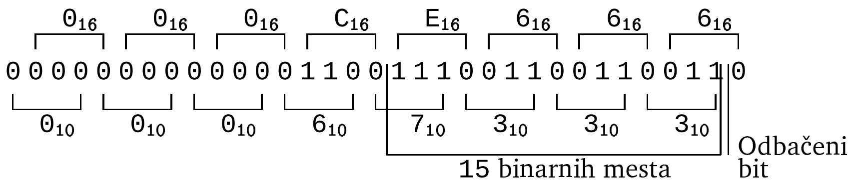
Naredna cifra je 8. Ostale cifre se dobijaju na isti način.

(c)
```
012345678 / 321 = 38460
  963
  2715         2715 > 321,   [2715/321] = 8,
  2568                        8 * 321 = 2568
   1476        1476 > 321,   [1476/321] = 4,
   1284                       4 * 321 = 1284
    1927       1927 > 321,   [1927/321] = 6,
    1926                      6 * 321 = 1926
      018      018  < 321,   [ 018/321] = 0
```

Pošto nema više cifara u deljeniku, količnik je -38460 a ostatak +18.

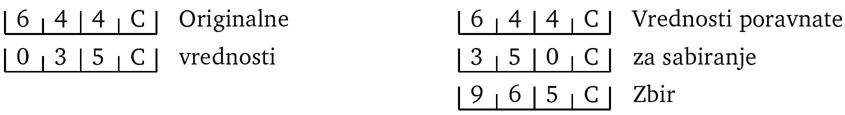
Realni brojevi u nepokretnom zarezu

1. Nekorektno smeštanje tačke osnove.



Slika 5: Uticaj nekorektne deklaracije na tačnost zapisa

2. Nekorektno skaliranje pri aritmetičkim operacijama.



Slika 6: Poravnanje pri sabiranju pakovanih brojeva