Binarno kodirani dekadni brojevi i aritmetika

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	84-2-1	višak 3	ciklički
0	0000	0000	0000	0000	0000	0011	0001
1	0001	0001	0001	1001	0111	0100	0101
2	0010	0010	0010	0111	0110	0101	0111
3	0011	0011	0011	0010	0101	0110	1111
4	0100	0100	0100	1011	0100	0111	1110
5	0101	1011	1000	0100	1011	1000	1100
6	0110	1100	1001	1101	1010	1001	1000
7	0111	1101	1010	1000	1001	1010	1001
8	1000	1110	1011	0110	1000	1011	1011
9	1001	1111	1100	1111	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

Konverzija izmedju binarnog zapisa i Grejovog koda

Neka je $A = a_{n-1}a_{n-1}...a_1a_0$ binarni broj i $G = g_{n-1}g_{n-2}...g_1g_0$ odgovarajući Grejov kod. Konverzija izmedju njih je definisana na sledeći način:

- U slučaju konverzije binarnog broja A u odgovarajući Grejov kod G:

$$g_i = \begin{cases} a_i & i = n-1 \\ a_i \oplus a_{i+1} & i \in [0, n-2] \end{cases}$$

- U slučaju konverzije Grejovog koda G u odgovarajući binarni broj A:

$$a_i = \begin{cases} g_i & i = n-1 \\ g_i \oplus a_{i+1} & i \in [0, n-2] \end{cases}$$

gde je sa $\oplus$ označena operacija ekskluzivne disjunkcije.

Primeri:

1. Konvertovati binarni broj $A = 100011$ u odgovarajući Grejov kod.

$$\begin{array}{lllllll} g_5 & = & a_5 & = & & & 1 \\ g_4 & = & a_4 \oplus a_5 & = & 0 \oplus 1 & = & 1 \\ g_3 & = & a_3 \oplus a_4 & = & 0 \oplus 0 & = & 0 \\ g_2 & = & a_2 \oplus a_3 & = & 0 \oplus 0 & = & 0 \\ g_1 & = & a_1 \oplus a_2 & = & 1 \oplus 0 & = & 1 \\ g_0 & = & a_0 \oplus a_1 & = & 1 \oplus 1 & = & 0 \end{array}$$

   Dobijeni Grejov kod je G=110010

2. Konvertovati $(234)_{10}$ u Grejov kod

   $(234)_{10} = (11101010)_2$. Cifre odgovarajućeg Grejovog koda su

$$\begin{array}{lllllll} g_7 & = & a_7 & = & & & 1 \\ g_6 & = & a_6 \oplus a_7 & = & 1 \oplus 1 & = & 0 \\ g_5 & = & a_5 \oplus a_6 & = & 1 \oplus 1 & = & 0 \\ g_4 & = & a_4 \oplus a_5 & = & 0 \oplus 1 & = & 1 \\ g_3 & = & a_3 \oplus a_4 & = & 1 \oplus 0 & = & 1 \\ g_2 & = & a_2 \oplus a_3 & = & 0 \oplus 1 & = & 1 \\ g_1 & = & a_1 \oplus a_2 & = & 1 \oplus 0 & = & 1 \\ g_0 & = & a_0 \oplus a_1 & = & 0 \oplus 1 & = & 1 \end{array}$$

   Dobijeni Grejov kod je G=10011111

3. Konvertovati Grejov kod G=1011001 u binarni broj.

$$\begin{array}{lllllll} a_6 & = & g_6 & = & & & 1 \\ a_5 & = & g_5 \oplus a_6 & = & 0 \oplus 1 & = & 1 \\ a_4 & = & g_4 \oplus a_5 & = & 1 \oplus 1 & = & 0 \\ a_3 & = & g_3 \oplus a_4 & = & 1 \oplus 0 & = & 1 \\ a_2 & = & g_2 \oplus a_3 & = & 0 \oplus 1 & = & 1 \\ a_1 & = & g_1 \oplus a_2 & = & 0 \oplus 1 & = & 1 \\ a_0 & = & g_0 \oplus a_1 & = & 1 \oplus 1 & = & 0 \end{array}$$

   Dobijeni binarni broj je A=1101110

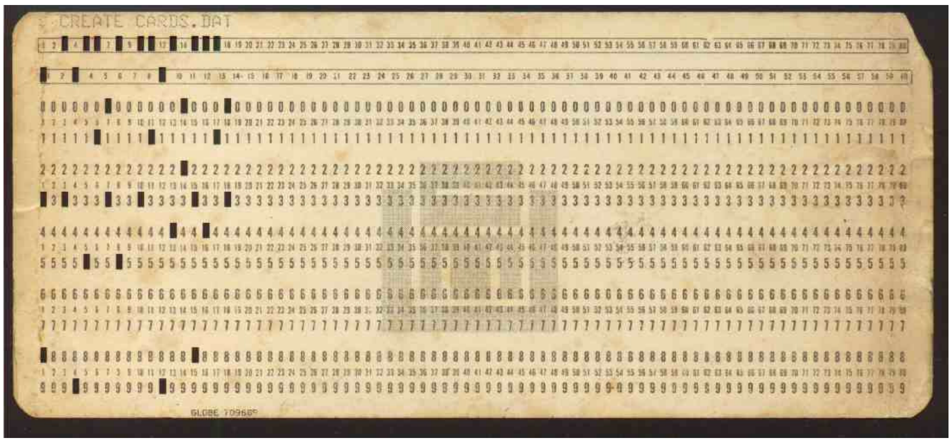
Zapis binarno kodiranih dekadnih brojeva

- 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.

Na najvećem broju računara koristi se zapis u obliku znak i apsolutna vrednost. Isti zapis ce biti korišćen u narednim primerima.

BCD zapis dekadnih brojeva

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



Slika 1: Bušena kartica

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

Slika 2: Zonsko i cifarsko ubušenje na kartici

257314

F 2	F 5	F 7	F 3	F 1	F 4	EBCDIC
bajt	bajt	bajt	bajt	bajt	bajt	

3 2	3 5	3 7	3 3	3 1	3 4	ASCII
bajt	bajt	bajt	bajt	bajt	bajt	

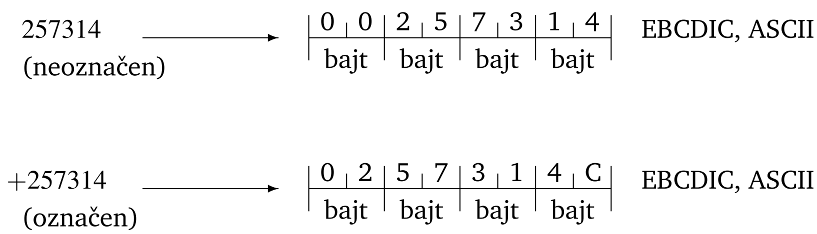
Slika 3: Nepakovani (zonski) BCD zapis neoznačenog dekadnog broja

$+257314$

F 2	F 5	F 7	F 3	F 1	C 4	EBCDIC
bajt	bajt	bajt	bajt	bajt	bajt	

3 2	3 5	3 7	3 3	3 1	C 4	ASCII
bajt	bajt	bajt	bajt	bajt	bajt	

Slika 4: Nepakovani (zonski) BCD zapis označenog dekadnog broja



Slika 5: Pakovani BCD zapis označenog i neoznačenog dekadnog broja sa parnim brojem cifara

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

Čen-Ho kodiranje

- Kodiranje su definisali 1971. godine Tien Či Čen i Irving Ho (*Tien Tien Chi Chen i Irving T. Ho*)
- Varijanta Hofmanovog kodiranja
- Kodira tri dekadne cifre u 10 bita
- 20% efikasnije kodiranje od BCD zapisa
- Deli cifre na male (0-7) i velike (9,9) na osnovu vrednosti prvog bita u BCD kodu
- Male cifre zahtevaju tri bita a velike jedan bit da bi se medjusobno razlikovale
- Kodiranje razmatra svaku od kombinacija tri cifre
  - Sve tri cifre su male (51.2% slučajeva): potrebno je 10 bitova (3+3+3 za cifre, 1 bit da označi ovu kombinaciju)
  - Dve cifre su male (38.4% slučajeva): potrebno je 7 bitova za cifre (3+3+1); preostala 3 bita označavaju kombinaciju
  - Jedna cifra je mala (9.6% slučajeva): potrebno je 5 bitova za cifre (3+1+1); preostalih 5 bitova označavaju kombinaciju
  - Sve cifre su velike (0.8% slučajeva): potrebno je 3 bita za cifre (1+1+1); preostalih 7 (potrebno je samo 5) označavaju kombinaciju

Shematski prikaz kodiranja

Cifre $C_1C_2C_3$ zapisane u BCD zapisu (abcd)(efgh)(ijkl) postaju (p)(qrs)(tuv)(wxy) gde je

aei	p	qrs	tuv	wxy
000	0	bcd	fgh	jkl
100	1	00d	fgh	jkl
010	1	01d	bch	jkl
001	1	10d	fgh	bcl
011	1	11d	00h	bcl
101	1	11d	01h	fgl
110	1	11d	10h	jkl
111	1	11d	11h	00l

Realizacija

- Hardversko kodiranje
- Softversko kodiranje pomoću logičkih funkcija

$$
\begin{aligned}
p &= a \vee e \vee i \\
q &= (b \wedge \neg\, e) \vee i \vee (a \wedge e) \\
r &= (c \wedge \neg i) \vee e \vee (a \wedge i) \\
s &= d \\
t &= (a \wedge e) \vee (f \wedge (\neg a \wedge \neg i)) \vee (b \wedge e \wedge \neg i) \\
u &= (a \wedge i) \vee (c \wedge e \wedge \neg i) \vee (g \wedge \neg e) \\
v &= h \\
w &= j \vee (b \wedge i) \vee (f \wedge a \wedge i) \\
x &= k \vee (c \wedge i) \vee (g \wedge a \wedge i) \\
y &= l
\end{aligned}
$$

Shematski prikaz dekodiranja

Cifre (p)(qrs)(tuv)(wxy) postaju (abcd)(efgh)(ijkl) koje predstavljaju BCD zapis cifara $C_1C_2C_3$

pqrtu	abcd	efgh	ijkl
0....	0qrs	0tuv	0wxy
100..	100s	0tuv	0wxy
101..	0tus	100v	0wxy
110..	0wxs	0tuv	100y
11100	0wxs	100v	100y
11101	100s	0wxv	100y
11110	100s	100v	0wxy
11111	100s	100v	100y

gde tačka ('.') označava da je sadržaj na toj poziciji nebitan.

Realizacija

- Hardversko dekodiranje
- Softversko dekodiranje pomoću logičkih funkcija

$$
\begin{array}{lcl}
a & = & (p \wedge \neg q \wedge \neg r) \vee (p \wedge q \wedge r \wedge (t \vee u)) \\
b & = & (q \wedge \neg p) \vee (t \wedge p \wedge \neg q \wedge r) \vee (w \wedge q \wedge (\neg r \vee (\neg t \wedge \neg u))) \\
c & = & (r \wedge (\neg p \vee (\neg q \wedge u))) \vee (x \wedge p \wedge q \wedge (\neg r \vee (\neg t \wedge \neg u))) \\
d & = & s \\
e & = & (p \wedge r \wedge (\neg q \vee \neg u \vee t)) \\
f & = & (t \wedge (\neg p \vee \neg r)) \vee (p \wedge q \wedge r \wedge \neg t \wedge u \wedge w) \\
g & = & (u \wedge (\neg p \vee \neg r)) \vee (p \wedge q \wedge r \wedge \neg t \wedge u \wedge x) \\
h & = & v \\
i & = & (p \wedge q \wedge (\neg r \vee \neg t \vee u)) \\
j & = & (w \wedge (\neg p \vee \neg q \vee (r \wedge t))) \\
k & = & (x \wedge (\neg p \vee \neg q \vee (r \wedge t))) \\
l & = & y
\end{array}
$$

Kodiranje gustim pakovanjem dekadnih cifara

- Definisao ga je Mike Cowlishaw Densely Packed Decimal kodiranje
- Slično Čen-Ho kodiranju, ali umesto Hofmanovog koda koristi drugačije preuredjenje bitova
- Kodira tri dekadne cifre u 10 bita
- 20% efikasnije kodiranje od BCD zapisa
- Deli cifre na male (0-7) i velike (9,9) na osnovu vrednosti prvog bita u BCD kodu
- Male cifre zahtevaju tri bita a velike jedan bit da bi se medjusobno razlikovale
- Kodiranje razmatra svaku od kombinacija tri cifre
  - Sve tri cifre su male (51.2% slučajeva): potrebno je 10 bitova (3+3+3 za cifre, 1 bit da označi ovu kombinaciju)
  - Dve cifre su male (38.4% slučajeva): potrebno je 7 bitova za cifre (3+3+1); preostala 3 bita označavaju kombinaciju
  - Jedna cifra je mala (9.6% slučajeva): potrebno je 5 bitova za cifre (3+1+1); preostalih 5 bitova označavaju kombinaciju
  - Sve cifre su velike (0.8% slučajeva): potrebno je 3 bita za cifre (1+1+1); preostalih 7 (potrebno je samo 5) označavaju kombinaciju

Shematski prikaz kodiranja

Cifre $C_1C_2C_3$ zapisane u BCD zapisu (abcd)(efgh)(ijkl) postaju (p)(qrs)(tuv)(wxy) gde je

aei	pqr stu v wxy	Komentar
000	bcd fgh 0 jkm	Sve cifre su male
001	bcd fgh 1 00m	Krajnje desna cifra je velika
010	bcd jkh 1 01m	Srednja cifra je velika
100	jkd fgh 1 10m	Krajnje leva cifra je velika
110	jkd 00h 1 11m	Krajnje desna cifa je mala (ostale dve su velike)
101	fgd 01h 1 11m	Srednja cifra je mala (ostale dve su velike)
011	bcd 10h 1 11m	Krajnje leva cifra je mala (ostale dve su velike)
111	00d 11h 1 11m	Sve cifre su velike; dva bita se ne koriste

Realizacija

- Hardversko kodiranje
- Softversko kodiranje pomoću logičkih funkcija

$$
\begin{aligned}
p &= b \vee (a \wedge j) \vee (a \wedge f \wedge i) \\
q &= c \vee (a \wedge k) \vee (a \wedge g \wedge i) \\
r &= d \\
s &= (f \wedge (\neg a \vee \neg i)) \vee (\neg a \wedge e \wedge j) \vee (e \wedge i) \\
t &= g \vee (\neg a \wedge e \wedge k) \vee (a \wedge i) \\
u &= h \\
v &= a \vee e \vee i \\
w &= a \vee (e \wedge i) \vee (\neg e \wedge j) \\
x &= e \vee (a \wedge i) \vee (\neg a \wedge k) \\
y &= m
\end{aligned}
$$

Shematski prikaz dekodiranja

Cifre (p)(qrs)(tuv)(wxy) postaju (abcd)(efgh)(ijkl) koje predstavljaju BCD zapis cifara $C_1C_2C_3$

vwxst	abcd efgh ijkm
0....	0pqr 0stu 0wxy
100..	0pqr 0stu 100y
101..	0pqr 100u 0sty
110..	100r 0stu 0pqy
11100	100r 100u 0pqy
11101	100r 0pqu 100y
11110	0pqr 100u 100y
11111	100r 100u 100y

gde tačka ('.') označava da je sadržaj na toj poziciji nebitan.

Realizacija

- Hardversko dekodiranje
- Softversko dekodiranje pomoću logičkih funkcija

$$
\begin{aligned}
a &= (v \wedge w) \wedge (\neg s \vee t \vee \neg x) \\
b &= p \wedge (\neg v \vee \neg w \vee (s \wedge \neg t \wedge x)) \\
c &= q \wedge (\neg v \vee \neg w \vee (s \wedge \neg t \wedge x)) \\
d &= r \\
e &= v \wedge ((\neg w \wedge x) \vee (\neg t \wedge x) \vee (s \wedge x)) \\
f &= (s \wedge (\neg v \vee \neg x)) \vee (p \wedge \neg s \wedge t \wedge v \wedge w \wedge x) \\
g &= (t \wedge (\neg v \vee \neg x)) \vee (q \wedge \neg s \wedge t \wedge w) \\
h &= u \\
i &= v \wedge ((\neg w \wedge \neg x) \vee (w \wedge x \wedge (s \vee t))) \\
j &= (\neg v \wedge w) \vee (s \wedge v \wedge \neg w \wedge x) \vee (p \wedge w \wedge (\neg x \vee (\neg s \wedge \neg t))) \\
k &= (\neg v \wedge x) \vee (t \wedge \neg w \wedge x) \vee (q \wedge v \wedge w \wedge (\neg x \vee (\neg s \wedge \neg t))) \\
m &= y
\end{aligned}
$$

Prednosti DPD u odnosu na Čen-Ho kodiranje

DPD kodiranje je po ideji slično Čen-Ho kodiranju ali umesto Hofmanovog koda koristi drugačije preuredjenje bitova što donosi sledeće prednosti u odnosu na Chen-Ho kodiranje:

1. Kompresija jedne ili dve dekadne cifre ((u optimalnih 4 ili 7 bitova respektivno) se dobija kao podskup kodiranja 3-cifrenog dekadnog broja. Posledica ovoga je da se proizvoljan broj dekadnih cifara kodira efikasnije. Na primer, 38 cifara se može kodirati pomoću 127 bita(130 bita za Čen-Ho), a 71 dekadna cifra u 237 bita (240 Čen-Ho)

2. Kodiranje jedne ili dve dekadne cifre je uvek desno poravnato u grupi od 10 bita, dok su ostali bitovi 0. Posledica je da kodirane dekadne cifre mogu da se zapišu polju veće dužine dopunjavanjem nula sa leve strane.

3. Pozicija i izbor indikator bita (bit p u Chen-Ho kodiranju, odnosno bit v u DPD kodiranju) dopuštaju da se svi jednocifreni brojevi (preciznije svi brojevi u intervalu [0,79] kodiraju desno poravnato na isti način kao u BCD (8421) kodu (Čen-Ho preslikava brojeve iz intervala [0-7])

Primer poredjenja Chen-Ho i DPD kodiranja

Cifre	BCD	Chen-Ho	Densely Packed
005	0000 0000 0101	000 000 0101	000 000 0101
009	0000 0000 1001	110 000 0000	000 000 1001
055	0000 0101 0101	000 010 1101	000 101 0101
099	0000 1001 1001	111 000 1001	000 101 1111
555	0101 0101 0101	010 110 1101	101 101 0101
999	1001 1001 1001	111 111 1001	001 111 1111

Primer: Kodirati brojeve 825 i 294 u DPD zapis.

```
  8    2    5          2    9    4
abcd efgh ijkm       abcd efgh ijkm
1000 0010 0101       0010 1001 0100

100 010 1  101       010 101 1  010
pqr stu v  wxy       pqr stu v  wxy
```

Dekadni zapis	
Bitovi u BCD zapisu	
BCD zapis	
DPD zapis	
Bitovi u DPD zapisu	

Decimalna aritmetika

Promena znaka

$$|\,8\,|\,2\,|\,1\,|\,6\,|\,4\,|\,C\,| \longrightarrow |\,8\,|\,2\,|\,1\,|\,6\,|\,4\,|\,D\,|$$

Slika 6: 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

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

Sabiranje se realizuje u dve faze:

1. Odredi se medjurezultat $C'_\alpha = A_\alpha + B_\alpha$:

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

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

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

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

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					
$C'\alpha$	=		0010	0001	1000	1010	1100

Druga faza			Korak 4	Korak 3	Korak 2	Korak 1	Korak 0
$C'\alpha$	=		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 $A = 945$ i $B = 86$ u kodu 8421.

$A\alpha$	=	0000	0000	1001	0100	0101
$B\alpha$	=	0000	0000	0000	1000	0110
C'_α	=	0000	0000	1000	1011	1111
$K\alpha$	=	0000	0000	0000	0110	0110
$C\alpha$	=	0000	0000	1000	0101	1001

6. 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*.

$A\alpha$	=		0100	1011	0110	0111	1000
$B\alpha$	=		0011	1100	1000	1001	1010
P'	=	0	1	0	1	1	0
C'_α	=		1000	0111	1111	0001	0010
$K\alpha$	=		1101	0011	1101	0011	0011
$C\alpha$	=		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.

$A\alpha$	=		1100	1100	0011	0011	0100
$B\alpha$	=		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_α	=		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_α	=		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_α	=		0011	0011	1101	0011	0011
C_α	=		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.