Računarstvo i informatika
Brojni sistemi i kodiranje znakova

Institut za matematiku i informatiku, PMF Kragujevac

Januar 2013. godine

1 Uvod

Savremeni računarski sistemi funkcionišu na bazi elektronskih komponenti koje operišu podacima u digitalizovanoj formi. Digitalizovana forma podataka podrazumeva da su podaci predstavljeni diskretnom vrednošću neprekidne fizičke veličine kao što su napon ili jačina struje. Istraživanja su pokazala da kada se odgovarajuća fizička veličina koja reprezentuje podatke podeli na veći broj uzastopnih bliskih vrednosti **funkcionisanje računara postaje nepouzdanije**. Zato je usvojeno da se u računarskim komponentama primenjuje binarni sistem predstave podataka koji podrazumeva samo dve vrednosti (dva stanja) napona na pomenutim komponentama. Osnovna jedinica računarske memorije je binarna cifra, zvana bit (eng. ***bi****nary digi****t***). Bit može da ima vrednosti 0 ili 1, što u fizičkom smislu označava dve vrednosti napona na elektronskoj komponenti računarskog sistema.

2 Brojevi konačne tačnosti

Kako je memorija računara fizički ograničena u smislu da postoji tačno određen broj elektronskih komponenti koje je sačinjavaju, standardima u računarskoj industriji je utvrđeno koliko memorije je namenjeno za čuvanje svakog tipa podatka.

Fiksna tačnost numeričkih podataka za posledicu ima konačnu tačnost istih. Jednostavno, fiksan broj bitova namenjen memorisanju numeričkih podataka nas primorava da radimo samo sa brojevima koji se mogu predstaviti fiksnim brojem cifara. Takvi brojevi se zovu **brojevi konačne tačnosti** (eng. *finite-precision numbers*).

Pretpostavimo da je potrebno da u polje koje je podeljeno na dva kvadrata treba da upišemo pozitivan ceo broj.

X	X

U polje sa raspoloživa dva kvadrata je moguće upisati tačno 100 brojeva i to 00, 01, 02, ..., 99. Očigledno je da je u pomenuto polje nemoguće upisati određene vrste brojeva:

- Brojeve veće od 99,
- Negativne brojeve i
- Razlomke.

Na primer, potrebno je sabrati 80 i 90 i dobijeni zbir upisati u dva kvadrata. U ovom slučaju zbir bi iznosio 170, dakle bio bi trocifren. Očigledno je da je trocifren broj nemoguće upisati u raspoložive kvadrate.

1	7	0

Druga vrsta problema nastaje kada je potrebno u respoloživa dva kvadrata upisati negativan broj koji predstavlja razliku dva broja iz posmatranog skupa celih, pozitivnih, dvocifrenih brojeva. Na primer, potrebno je izračunati razliku broja 50 i 70 i dobijenu razliku upisati u raspoložive kvadrate. Kako je razlika ova dva broja -20 očigledno je da ne postoji mogućnost da se izračunata razlika unese u dva kvadrata.

-	2	0

Dalji primer, potrebno je izračunati proizvod broja 50 i broja 60 i dobijeni proizvod upisati u raspoložive kvadrate.

3	0	0	0

Zbog toga kažemo da **skup brojeva konačne tačnosti nije zatvoren za operacije sabiranja, oduzimanja i množenja**. Činjenica da računari imaju konačne memorije i da zbog toga manipulišu podacima konačne tačnosti za posledicu ima opasnost da rezultati pojedinih izračunavanja budu pogrešni. Zbog toga savremeni računari poseduju specijalan hardver koji otkriva takozvane greške prekoračenja.

3 Zapis podataka

Zapis podataka predstavlja način na koji će podaci bliski čoveku (alfanumerički i numerički) biti prevedeni u binarnu formu koju prepoznaje računar.

Generalno sve podatke možemo podeliti u dve grupe:

- znakovne (alfanumeričke) podatke i
- brojčane (numeričke) podatke.

Međutim, ni zapisivanje numeričkih podataka nije jedinstveno i zavisi od tipa podataka, odnosno vrste brojeva koje treba predstaviti. Zbog toga je u skladu sa metodologijom zapisivanja izvršena dalja podela numeričkih podataka na:

- cele brojeve,
- binarno kodirane dekadne brojeve,
- realne brojeve zapisane u nepokretnom zarezu i
- realne brojeve u pokretnom zapisu.

4 Brojni sistemi date osnove

U opštem slučaju vrsta brojnog sistema je definisana brojem cifara koje se koriste za generisanja brojeva. Broj različitih cifara brojnog sistema se naziva osnova brojnog sistema. Brojčana vrednost X u brojnom sistemu sa osnovom N piše se u obliku niza cifara u kojem svaka cifra ima svoju težinu. Brojčana vrednost cifre x_i u pomenutom nizu cifara zavisi od vrednosti ali i položaja cifre i jednaka je $V(x_i) = x_i \cdot N^i$.

Na taj način ukupna brojčana vrednost X u brojnom sistemu sa osnovom N se može dobiti kao:

$$(X)_N = \sum_{i=-m}^{n-1} x_i N^i = x_{n-1}N^{n-1} + x_{n-2}N^{n-2} + \cdots + x_0 N^0 + x_{-1}N^{-1} + \cdots x_{-m}N^{-m},$$

pri čemu m predstavlja broj cifara u razlomljenom delu, a n broj cifara u celobrojnom delu.

5 Vrste brojčanih sistema

5.1 Dekadni sistem

Brojni sistem kod koga je osnova $N = 10$ zove se **dekadni sistem**. Primer broja zapisanog u ovom sistemu je 24864.687. Ako primenimo prethodni izraz, ovaj broj možemo napisati u sledećem obliku:

$$(24864.687)_{10} = 2 \cdot 10^4 + 4 \cdot 10^3 + 8 \cdot 10^2 + 6 \cdot 10^1 + 4 \cdot 10^0 + 6 \cdot 10^{-1} + 8 \cdot 10^{-2} + 7 \cdot 10^{-3}$$

5.2 Binarni sistem

Brojni sistem kod koga je osnova $N = 2$ zove se **binarni sistem**. Funkcionisanje savremenih računara je bazirano na ovom sistemu. Primer broja zapisanog u ovom sistemu je 1011101. Ako primenimo izraz ovaj broj možemo napisati u sledećem obliku:

$$(1011101)_2 = 1 \cdot 2^6 + 0 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = (93)_{10}$$

Sabiranjem članova u izrazu na jednostavan način možemo odrediti vrednost binarnog broja 1011101 u dekadnom sistemu.

5.3 Oktalni sistem

Brojni sistem kod koga je osnova $N = 8$ zove se **oktalni sistem**. Cifre koje se koriste u ovom sistemu su: 0, 1, 2, 3, 4, 5, 6, 7.

U prethodnim generacijama računara oktalni sistem je imao značajnu ulogu. Primer broja zapisanog u ovom sistemu je 365,142. Ako primenimo izraz ovaj broj možemo napisati u sledećem obliku:

$$(365.142)_8 = 3 \cdot 8^2 + 6 \cdot 8^1 + 5 \cdot 8^0 + 1 \cdot 8^{-1} + 4 \cdot 8^{-2} + 2 \cdot 8^{-3}$$

5.4 Heksadekadni sistem

Brojni sistem kod koga je osnova $N = 16$ zove se **heksadekadni sistem**. Očigledno je da je za ovaj brojni sistem pored 10 decimalnih cifara potrebno još 6 simbola. Cifre koje se koriste u heksadecimalnom sistemu su: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F. Odgovarajuće "cifre" A, B, C, D, E, F u decimalnom sistemu su: 10, 11, 12, 13, 14 i 15. Primer broja zapisanog u ovom sistemu je A35EF.

6 Binarni zapis celih brojeva

U binarnom sistemu celi brojevi se zapisuju pomoću niza cifara 0 i 1. Svaka cifra, 0 ili 1, se u računaru zapisuje u jedan bit koji predstavlja najmanju jedinicu računarske memorije. Kao što je opisano ranije broj bitova namenjenih za binarni zapisa numeričkih podataka je ograničen i određen je veličinom registra na procesoru računara. Na najvećem broju procesora taj broj iznosi 32 što znači da se celi brojevi zapisuju kao 32-cifreni binarni brojevi. U opštem slučaju forma binarnog broja u n-bitnom zapisu ima sledeći oblik:

$$a_{n-1}a_{n-2}a_{n-3}\cdots a_1a_0$$

Očigledno dužina broja je n. Mesto (bit) u zapisu broja na poziciji 0 se naziva mesto najmanje težine a mesto (bit) na poziciji $n-1$ se naziva mesto najveće težine.

6.1 Zapis neoznačenih brojeva

Broj čiji binarni zapis ne sadrži znak naziva se **neoznačen broj**. Binarni zapis neoznačenih celih brojeva je identičan njihovoj reprezentaciji u binarnom brojčanom sistemu.

Ako pretpostavimo da za binarni zapis proizvoljnog neoznačenog celog broja I imamo n bitova (cifara) onda izraz možemo napisati u sledećem obliku:

$$I = \sum_{i=0}^{n-1} a_i 2^i = a_{n-1}2^{n-1} + a_{n-2}2^{n-2} + \cdots + a_1 2^1 + a_0 2^0,$$

gde a_i predstavlja cifru binarnog zapisa decimalnog broja I na poziciji i. U opštem slučaju sa n cifara moguće je zapisati bilo koji neoznačeni decimalni ceo broj I za koji važi da je $I \leq 2^n - 1$. Tako se u 8-bitnom binarnom zapisu mogu zapisati neoznačenih celi decimalni brojevi u intervalu [0,255] dok se u 16-bitnom zapisu mogu zapisati brojevi u intervalu [0,65535]. U tabeli ispod dati su primeri zapisa nekih neoznačenih brojeva.

Decimalna vrednost	**Binarni zapis**
0	0000000000000000
1	0000000000000001
12	0000000000001100
64	0000000001000000
4108	0001000000001100
32768	1000000000000000
65536	1111111111111111

6.2 Pretvaranje brojeva iz decimalnog brojnog sistema u binarni

Neoznačene cele brojeve iz decimalnog brojnog sistema je moguće pretvoriti u binarne na dva načina.

Prvi način pretvaranja neoznačenog decimalnog broja se može primeniti samo na cele decimalne brojeve, a svodi se na deljenje sa decimalnim brojem 2. Dobijeni količnik se upisuje odmah ispod originalnog broja. Prilikom deljenja bilo kog celog decimalnog broja sa 2 ostatak može da bude 0 (nema ostatka) ili najviše 1. Dobijeni ostatak se piše odmah pored dobijenog količnika. Postupak se ponavlja dalje primenjući se na količnik dobijen u prethodnom koraku sve dok se na kraju ne dobije količnik jednak 0. Rezultat postupka su dve kolone sa količnicima i odgovarajućim ostacima. Binarni broj koji odgovara početnom decimalnom celom broju predstavlja niz ostataka gledano odozdo:

187	*ostaci*	*pozicija*
93	1	0
46	1	1
23	0	2
11	1	3
5	1	4
2	1	5
1	0	6
0	1	7

Drugi način potiče direktno iz definicije binarnih brojeva i sastoji se u sledećem. Od decimalnog broja se oduzme decimalni broj koji odgovara najvećem stepenu dvojke a koji je manji od samog broja. Postupak se zatim ponavlja nad dobijenom razlikom pri čemu se svaki put od razlike oduzima broj koji predstavlja stepen dvojke koji je za jedan manji od stepena

korišćenog u prethodnom koraku. Binarni broj se dobija tako što se cifra 1 stavlja na one pozicije koje odgovaraju iskorišćenim stepenima dvojke a cifra 0 na one pozicije u kojima odgovarajući stepen dvojke nije korišćen. Rešenje: $(187)_{10} = (10111011)_2$.

$2^8 > 187 > 2^7$	$187 - 2^7 = 187 - 128 = 59$	1(cifra najveće težine na poziciji 7)
$2^6 > 59$		0
$59 > 2^5$	$59 - 2^5 = 59 - 32 = 27$	1
$27 > 2^4$	$27 - 2^4 = 27 - 16 = 11$	1
$11 > 2^3$	$11 - 2^3 = 11 - 8 = 3$	1
$2^2 > 3$		0
$3 > 2^1$	$3 - 2^1 = 3 - 2 = 1$	1
$1 = 2^0$		1

6.3 Zapis označenih brojeva

Broj čiji zapis uključuje i njegov znak se naziva **označen broj**. Navedena specifičnost naizgled ne predstavlja poseban problem, međutim ograničenost zapisivanja u računarstvu stvara izrazite probleme prilikom zapisivanja znaka broja. Generalno postoji 4 načina čije će prednosti i nedostaci biti analizirane u tekstu koji sledi.

6.3.1 Znak i apsolutna vrednost

Najprirodniji način zapisivanja označenih brojeva je zapisivanje u kojem bi se jedan bit "žrtvovao" za označavanje znaka. Rešenje koje se nameće jeste da u n-bitnom zapisu krajnji levi bit tj. bit najveće težine (bit na poziciji $n-1$) označava znak, a preostalih $n-1$ bitova označava vrednost broja.

Na taj način se menja i skup označenih decimalnih brojeva koji se mogu predstaviti u binarnom zapisu. U opštem slučaju za označen ceo decimalni broj I zapisan pomoću znaka i apsolutne vrednosti u n-cifarskom binarnom zapisu važi $I \in [-2^{n-1}+1, 2^{n-1}-1]$. Tako se u 8-bitnom binarnom zapisu mogu zapisati označeni celi decimalni brojevi u intervalu $[-127, 127]$ dok se u 16-bitnom zapisu mogu zapisati brojevi u intervalu $[-32767, 32767]$.

Dva osnovna nedostatka ovog načina zapisa označenih celobrojnih vrednosti su:

1. Postoje dva različita binarna zapisa koji reprezentuju nulu. Na primer u 8-bitnom zapisu: 00000000 i 10000000,

2. Za otkrivanje eventualnog prekoračenja prilikom računskih operacija potrebno je ispitivati znak i apsolutnu vrednost oba operanda.

Primer 6.1

$$+61 = 00111101$$

$$-61 = 10111101$$

6.3.2 Nepotpuni komplement (Komplement jedinice)

U zapisu u nepotpunom komplementu označenog celog broja I krajnja leva cifra se takođe koristi za definisanje znaka broja kao što je to slučaj i u binarnom zapisu pomoću znaka i apsolutne vrednosti. Na potpuno isti način vrednost 0 krajnje leve cifre označava znak +, a vrednost 1 označava znak −.

Ostalih $n-1$ cifara određuju vrednost broja I, i to:

- Ukoliko je označeni ceo broj I pozitivan onda je binarni zapis u preostalih $n-1$ cifara praktično binarni zapis apsolutne vrednosti broja I.

- Ukoliko je označeni ceo broj I negativan onda se binarni zapis u preostalih $n-1$ cifara dobija tako što se svaka cifra binarnog zapisa apsolutne vrednosti broja I zameni njenim komplementom.

U opštem slučaju za označen ceo decimalni broj I zapisan pomoću nepotpunog komplementa u n-cifarskom binarnom zapisu važi $I \in [-2^{n-1}+1, 2^{n-1}-1]$. Tako se u 8-bitnom binarnom zapisu mogu zapisati označeni celi decimalni brojevi u intervalu $[-127, 127]$ dok se u 16-bitnom zapisu mogu zapisati brojevi u intervalu $[-32767, 32767]$.

Izvršavanje osnovnih računskih operacija sa brojevima zapisanim u zapisu nepotpunog komplementa je znatno jednostavnije i efikasnije u odnosu na zapise pomoću znaka i apsolutne vrednosti. Međutim, neki nedostaci prethodnog načina zapisivanja su nasleđeni i u ovoj vrsti zapisa. Tako na primer, nula se i u ovom zapisu može zapisati na dva načina:

$$+0 = 00000000$$

$$-0 = 11111111$$

PRIMER 6.2 *[Broj −61]*

Zapis u formi znaka i apsolutne vrednosti:	$-61 = 10111101$
Zamena cifara komplementom:	$-61 = 11000010$

Takođe, može se koristiti i svojevrsi „trik“ pomoću koga je moguće dobiti nepotpuni komplement broja bez konverzije u binarni zapis, tj. direktno u heksadekadnom formatu. Sistem se sastoji u tome da se pojedinačne heksadekadne cifre oduzimaju od F (heksadekadno 15). Na primer, broj $+61$ je u heksadekadnom zapisu $3D$. Da bi se našao nepotpuni komplement, dovoljno je svaku njegovu cifru oduzeti od F, pri čemu se dobija $C2$ u heksadekadnom zapisu.

6.3.3 Potpuni komplement (Komplement dvojke)

Jedan od najosetljivijih nedostataka je činjenica da se nula u binarnom zapisu nepotpunog komlementa može zapisati na dva načina. U cilju prevazilaženja tog problema razvijen je binarni zapis potpunog komplementa.

U slučaju da je označeni ceo broj I pozitivan onda je binarni zapis u preostalih $n-1$ cifara praktično binarni zapis apsolutne vrednosti broja I.

Ukoliko je označeni ceo broj I negativan onda se binarni zapis u preostalih $n-1$ cifara dobija iz dva koraka. Prvo se svaka cifra zameni svojim komplementom kao u postupku dobijanja nepotpunog komlementa, a zatim se na tako dobijen binarni zapis (nepotpuni komplement) na mesto najmanje težine doda 1.

Glavne prednosti binarnog zapisa potpunog komplementa su:

- jednostavnije izvođenje računskih operacija i
- postoji samo jedan zapis nule čime se olakšava realizacija poređenja sa nulom.

U opštem slučaju za označen ceo decimalni broj I zapisan pomoću potpunog komplementa u n-cifarskom binarnom zapisu važi $I \in [-2^{n-1}, 2^{n-1}-1]$. Tako se u 8-bitnom binarnom zapisu mogu zapisati označeni celi decimalni brojevi u intervalu $[-128, 127]$ dok se u 16-bitnom zapisu mogu zapisati brojevi u intervalu $[-32768, 32767]$. U tabeli 1 dati su primeri predstavljanja označenih brojeva binarnim zapisom.

PRIMER 6.3 *Odrediti 8-cifarske binarne zapise potpunog komplementa decimalnih brojeva +61 i -61.*

$$+61 = 00111101$$

Zapis u formi znaka i apsolutne vrednosti:	$-61 = 10111101$
Zamena cifara komplementom:	$-61 = 11000010$
Dodavanje jedinice na mesto najmanje težine:	$+00000001$
	$-61 = 11000011$

Tabela 1: 8-cifarski zapis označenih celih brojeva u obliku znaka i apsolutne vrednosti, nepotpunog komplementa i potpunog komplementa

Decimalna vrednost	Znak i apsolutna vrednost	Nepotpuni komplement	Potpuni komplement
+127	01111111	01111111	01111111
+64	01000000	01000000	01000000
+32	00100000	00100000	00100000
+16	00010000	00010000	00010000
+15	00001111	00001111	00001111
+10	00001010	00001010	00001010
+9	00001001	00001001	00001001
+8	00001000	00001000	00001000
+7	00000111	00000111	00000111
+6	00000110	00000110	00000110
+5	00000101	00000101	00000101
+4	00000100	00000100	00000100
+3	00000011	00000011	00000011
+2	00000010	00000010	00000010
+1	00000001	00000001	00000001
+0	00000000	00000000	00000000
-0	10000000	11111111	———
-1	10000001	11111110	11111111
-2	10000010	11111101	11111110
-3	10000011	11111100	11111101
-4	10000100	11111011	11111100
-5	10000101	11111010	11111011
-6	10000110	11111001	11111010
-7	10000111	11111000	11111001
-8	10001000	11110111	11111000
-9	10001001	11110110	11110111
-10	10001010	11110101	11110110
-15	10001111	11110000	11110001
-16	10010000	11101111	11110000
-32	10100000	11011111	11100000
-64	11000000	10111111	11000000
-127	11111111	10000000	10000001
-128	———	———	10000000

6.4 Celobrojna aritmetika

U ovom poglavlju je izložena metodologija izvođenja računskih operacija promene znaka, sabiranja i oduzimanja binarnih zapisa decimalnih brojeva.

6.4.1 Promena znaka

Primer 6.4 (Znak i apsolutna vrednost) *U zapisu pomoću znaka i apsolutne vrednosti promena znaka broja se vrši jednostavnim komplementiranjem cifre (bita) za označavanje znaka.*

$$(+13)_{10} \Leftrightarrow (00001101)_2$$
$$(-13)_{10} \Leftrightarrow (10001101)_2$$

Primer 6.5 (Nepotpuni komplement) *U zapisu nepotpunog komplementa promena znaka broja se vrši komplementiranjem svih cifara u binarnom zapisu uključujući i cifru (bit) za znak.*

$$(+13)_{10} \Leftrightarrow (00001101)_2$$
$$(-13)_{10} \Leftrightarrow (11110010)_2$$

Primer 6.6 (Potpuni komplement) *U binarnom zapisu potpunog komplementa promena znaka broja se realizuje u dva koraka:*

1. *U prvom koraku se vrši komplementiranje svake cifre binarnog zapisa uključujući i cifru za znak.*
2. *U drugom koraku se dobijeni binarni broj sabere sa jedinicom, tj. na cifru najmanje težine dobijenog binarnog zapisa se doda 1. Primer:*

$$-9 = 11110111$$

- *Prvi korak: komplementiranje svih cifara* $11110111 \rightarrow 00001000$
- *Drugi korak: sabiranje sa 1*

$$\begin{array}{rr} & 00001000 \\ + & 00000001 \\ \hline = & 00001001 \\ +9 = & 00001001 \end{array}$$

6.4.2 Sabiranje i oduzimanje binarnih brojeva

Sabiranje i oduzimanje brojeva u binarnom sistemu se vrši na način koji je u potpunosti analogan sabiranju i oduzimanju brojeva u decimalnom brojnom sistemu. Kao što je to slučaj u decimalnom brojnom sistemu i u binarnom brojnom sistemu se sabiraju cifre iste težine. U decimalnom brojnom sistemu ukoliko je zbir dve cifre iste težine jednak ili veći od 10 vrši se prenos jedinice na sledeći par cifara. Analogno tome, prilikom sabiranja dva broja u binarnom brojnom sistemu ukoliko je zbir dve cifre iste težine jednak 2 cifra zbira te težine će biti 0 a na sledeći par cifara se vrši prenos jedinice.

Postupak sabiranja i oduzimanja u binarnom brojnom sistemu će biti opisan na primeru 3-cifrenih binarnih brojeva.

Primer 6.7 (Trocifreni brojevi)

$$\begin{array}{rr} & 011 \\ + & 010 \\ \hline = & 101 \end{array}$$

6.4.3 Sabiranje i oduzimanje binarnih brojeva u formi znak i apsolutna vrednost

Prilikom sabiranja i oduzimanja binarnih brojeva u zapisu znak i apsolutna vrednost posebno se analizira znak a posebno apsolutne vrednosti brojeva.

Neka su A i B binarni brojevi zapisani u formi znaka i apsolutne vrednosti. Prilikom sabiranja brojeva A i B možemo razlikovati dva slučaja:

1. **Brojevi A i B su istog znaka.** Ako su brojevi A i B istog znaka tada i njihov zbir ima isti znak. Apsolutna vrednost zbira brojeva A i B je jednaka zbiru njihovih apsolutnih vrednosti.

2. **Brojevi A i B su različitog znaka.** Ako su brojevi A i B različitog znaka, njihov zbir ima znak sabirka čija je apsolutna vrednost veća. Apsolutna vrednost zbira je jednaka razlici njihovih apsolutnih vrednosti pri čemu se oduzima manja apsolutna vrednost od veće.

Oduzimanje $A - B$ se se svodi na sabiranje uz promenu znaka umanjioca.

Primer 6.8

$$+61 - (+14) = +61 + (-14)$$

$$\begin{array}{rcl} +61 & = & 0|0111101 \\ -14 & = & 1|0001110 \\ \hline +47 & = & 0|0101111 \end{array}$$

Primer 6.9

$$-61 - 75$$

$$\begin{array}{rcl} -61 & = & 1|0111101 \\ -75 & = & 1|1001011 \\ \hline *** & = & 1|10001000 \end{array}$$

U poslednjem primeru došlo je do prekoračenja koje savremeni procesori mogu da detektuju i o tome obaveste pozivajući program. Važno je napomenuti da do prekoračenja prilikom sabiranja binarnih brojeva može doći samo ukoliko su brojevi istog znaka.

6.4.4 Sabiranje i oduzimanje binarnih brojeva zapisu potpunog komplementa

Realizacija računskih operacija sa binarnim brojevima u potpunom komplementu je znatno jednostavnija nego sa binarnim brojevima u formi znaka i apsolutne vrednosti. Takođe, binarni zapis u potpunom komplementu ima samo jednu reprezentaciju nule. Zbog toga u savremenim računarima se označeni celi brojevi zapisuju u potpunom komplementu.

Sabiranje n-cifarskih binarnih brojeva A i B u zapisu potpunog komplemena se sastoji u sledećem. Binarni brojevi u potpunom komplementu se sabiraju sabiranjem cifara iste težine. Pri tome se sabiraju i cifre njaveće težine (cifre na pozicijama $n-1$) koje označavaju znak broja. Eventualni prenos sa pozicije za znak (pozicije $n-1$) se eliminiše i ne učestvuje u formiranju rezultata.

Oduzimanje $A - B$ se svodi na sabiranje uz prethodnu promenu znaka umanjiocu u skladu sa pravilima za promenu znaka binarnim brojevima u potpunom komplementu.

Primer 6.10

$$+61 + 14$$

$$\begin{array}{rcl} +61 & = & 0|0111101 \\ +14 & = & 0|0001110 \\ \hline +75 & = & 0|1001011 \end{array}$$

Primer 6.11

$$+61 - (+14) = +61 + (-14)$$

$$\begin{array}{rcl} +61 & = & 0|0111101 \\ -14 & = & 1|1110010 \\ \hline +47 & = & 0|0101111 \end{array}$$

7 Zadaci

1. Prevesti broj $(-112)_{10}$ u binarni zapis znak i apsolutna vrednost, nepotpuni i potpuni komplement.
2. Prevesti broj $(-89)_{10}$ u binarni zapis znak i apsolutna vrednost, nepotpuni i potpuni komplement.
3. Izračunati $67-112$ u binarnom zapisu znak i apsolutna vrednost i potpunom komplementu.
4. Izračunati $118-37$ u binarnom zapisu znak i apsolutna vrednost i potpunom komplementu.
5. Izračunati $-15118-3199$ u binarnom zapisu znak i apsolutna vrednost i potpunom komplementu.

8 Kodiranje znakova

8.1 ASCII

ASCII tabela ili ASCII kod (engl. American Standard Code for Information Interchange) je Američki standardni kod za razmenjivanje unformacija. ASCII kod obuhvata 128 znakova pri čemu je svaki znak predstavljen 7-bitnim binarnim zapisom. Bez obzira što se za kodiranje koristi niz od 7 binarnih cifara znakovi u ASCII kodu se čuvaju i prenose u 8-bitnom zapisu pri čemu se krajnji levi bit koristi za kontrolu parnosti. Znakovi koji su obuhvaćeni ASCII kodom se mogu podeliti na grafičke znakove (znakove koji se štampaju) i kontrolne (upravljačke) znakove. Znakovi koji se štampaju su velika i mala slova abecede, decimalne cifre, znakovi interpunkcije i nekoliko matematičkih simbola, kao što je to dato na Slici 8.1.

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040		Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	(	(	72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051	)	)	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[	[	123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174			|
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135	]	]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177	&#127;	DEL

Slika 1: ASCII kodna tabela

8.2 UNICODE

Iz strukture tabele ASCII kodova jasno se može zaključiti da je namenjena engleskom govornom području. Razlog je taj što je ova tabela nastala u Americi gde je razvoj računarske

industrije bio najizrazitiji.

Internacionalizacijom računarske tehnologije primena ASCII koda je postala neadekvatna. Broj znakova koje je ASCII kod obuhvatao je bio neprimeren raznovrsnosti koju poseduju drugi jezici. Tako na primer, u francuskom jeziku postoje akcenti koje takođe treba kodirati, a u nemačkom jeziku dijakritici koji takođe moraju imati svoj zapis u računaru.

Navedeni problemi su bili prevaziđeni uvođenjem standarda IS 10646 koji je sa sobom doneo novi kodni sistem nazvan UNICODE (UNIversal enCODE). UNICODE se zasniva na ideji da se svakom znaku dodeli 16-bitni binarni zapis koji je nazvan kodna tačka (engl. code point).

S obzirom da se bazira na 16-bitnom binarnom zapisu UNICODE poseduje 65536 kodnih tačaka odnosno mogućnost da se na ovaj način kodira 65536 različitih znakova. Iako na prvi pogled deluje da je kapacitet UNICODE sistema dovoljan za obuhvatanje svih poznatih svetskih znakova i simbola već polovina kapaciteta je već potrošena. UNICODE sistem je inkorporiao perthodno opšte prihvaćeni ASCII kod tako što je prvih 256 kodnih tačaka dodeljeno skupu `Latin-1` nasledniku izvornog ASCII koda. Time je postignuto brzo prihvatanje UNICODE kao novog kodnog sistema.

Literatura

[1] ICT tehnologije, Ekonomski fakultet Kragujevac, 2010.