A

BINARNI BROJEVI

Aritmetika koju koristi računar u izvesnom pogledu se razlikuje od aritmetike koju koriste ljudi. Najvažnija razlika je to što računari rade s brojevima konačne i fiksne tačnosti. Druga razlika je to što računari za predstavljanje brojeva umesto decimalnog sistema po pravilu koriste binarni brojčani sistem. To su teme ovog dodatka.

A.1 BROJEVI KONAČNE TAČNOSTI

Dok obavljamo neku aritmetičku operaciju, obično malo mislimo na to koliko je decimalnih mesta potrebno da se broj predstavi. Fizičari mogu da izjave kako u svemiru ima 10^{78} elektrona, ne brinući o tome što se taj broj u razvijenom obliku piše sa 79 decimalnih mesta. Neko ko „pešice“ izračunava neku funkciju s tačnošću od šest značajnih cifara, međurezultate će izračunavati sa sedam ili sa osam cifara, ili sa onoliko cifara koliko je potrebno. Nikada se neće pojaviti problem što list hartije nije dovoljno širok za zapisivanje sedmocifrenih brojeva.

Kod računara stvari stoje sasvim drugačije. U većini računara, količina memorije raspoložive za smeštanje brojeva obično se fiksira u trenutku projektovanja računara. Uz određen napor, programer može da prikaže dva, tri i više puta preciznije brojeve nego što to diktira ta fiksna veličina, ali to ne menja prirodu ove teškoće. Konačna priroda resursa računara prisiljava nas da radimo samo s brojevima koji se mogu predstaviti fiksnim brojem cifara. Takve brojeve zovemo **brojevi konačne tačnosti** (engl. *finite-precision numbers*).

Da bismo proučili svojstva brojeva konačne tačnosti, ispitajmo skup pozitivnih celih brojeva koji se mogu predstaviti s tri decimalne cifre, dakle, bez decimalnog zareza i bez znaka. Ovaj skup ima tačno 1000 članova: 000, 001, 002, 003, ..., 999. U njemu je nemoguće izraziti određene vrste brojeva, na primer:

1. Brojeve veće od 999.
2. Negativne brojeve.
3. Razlomke.
4. Iracionalne brojeve.
5. Kompleksne brojeve.

Važno aritmetičko svojstvo skupa svih celih brojeva jeste **zatvorenost** u odnosu na operacije sabiranja, oduzimanja i množenja. Drugim rečima, za svaki par celih brojeva i i j, $i + j$, $i - j$ i $i \times j$ takođe su celi brojevi. Skup celih brojeva nije zatvoren u odnosu na deljenje jer postoje vrednosti i i j za koje se i/j ne može izraziti celim brojem (npr. 7/2 i 1/0).

Brojevi konačne tačnosti nisu zatvoreni ni za jednu od četiri osnovne operacije, kao što se vidi iz sledećeg primera s trocifrenim brojevima:

600 + 600 = 1200 (prevelik)
003 - 005 = - 2 (negativan)
050 × 050 = 2500 (prevelik)
007 / 002 = 3,5 (nije ceo broj)

Neodgovarajući rezultati mogu se svrstati u dve klase koje se međusobno isključuju: rezultat je veći od najvećeg broja iz skupa (greška prekoračenja gornje granice) ili je manji od najmanjeg broja iz skupa (greška prekoračenja donje granice), ili rezultat nije ni jedno ni drugo, već jednostavno ne spada u skup. U četiri gornja primera, prva tri rezultata spadaju u prvu klasu, a poslednji u drugu.

Pošto računari imaju konačne memorije i stoga moraju raditi s brojevima konačne tačnosti, rezultati određenih izračunavanja biće pogrešni, s tačke gledišta klasičnog matematičara. Uređaj za računanje koji daje pogrešan odgovor, a tehnički je u savršenom stanju, može na prvi pogled izgledati besmisleno, ali je greška logična posledica njegove konačne prirode. Neki računari imaju specijalan hardver koji otkriva greške prekoračenja.

Algebra brojeva konačne tačnosti razlikuje se od uobičajene algebre. Kao primer, razmotrite pravilo asocijativnosti:

$$a + (b - c) = (a + b) - c$$

Izračunajmo obe strane jednačine za $a = 700$, $b = 400$, $c = 300$. Da biste izračunali levu stranu jednačine, najpre izračunajte $(b - c)$, što daje 100, a zatim to dodajte vrednosti a, da biste dobili 800. Da biste izračunali desnu stranu jednačine, prvo izračunajte $(a + b)$, što daje prekoračenje u konačnoj aritmetici trocifrenih celih brojeva. Rezultat može zavisiti od korišćenog računara, ali neće biti 1100. Oduzimanje 300 od broja koji nije 1100 neće dati 800. Prema tome, pravilo asocijativnosti ne važi, već je važan redosled operacija.

Kao drugi primer, razmotrite pravilo distributivnosti:

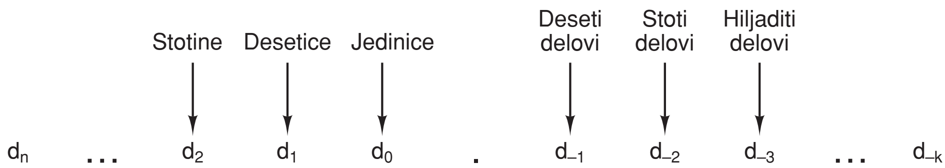
$a \times (b - c) = a \times b - a \times c$

Izračunajmo obe strane jednačine za $a = 5$, $b = 210$, $c = 195$. Leva strana će dati 5×15, dakle, 75. Desna strana neće dati 75 jer operacija $a \times b$ izaziva prekoračenje.

Sudeći po ovim primerima, moglo bi se zaključiti da računari, kao uređaji opšte namene, nisu baš pogodni za aritmetičke operacije. Ovakav zaključak je, naravno, pogrešan, ali smo ga namerno izveli da pokažemo koliko je važno razumeti kako računar radi i koja su mu ograničenja.

A.2 BROJČANI SISTEMI DATE OSNOVE

Običan decimalan broj koji svi poznajemo sastoji se od niza decimalnih cifara i (možda) decimalnog zareza. Njegov opšti oblik i uobičajeno tumačenje prikazani su na slici A-1. Broj 10 izabran je kao **osnova** (engl. *radix*) za stepenovanje zato što mi koristimo decimalne brojeve sa osnovom 10. U računarima je često pogodnije koristiti drugačiju osnovu. Najvažnije takve osnove su 2, 8 i 16. Brojčani sistemi na tim osnovama su **binarni**, **oktalni** i **heksadecimalni**.



$$\text{Broj} = \sum_{i=-k}^{n} d_i \times 10^i$$

Slika A-1. Opšti oblik decimalnog broja.

Brojčani sistem osnove k mora imati k različitih simbola za predstavljanje cifara od 0 do k - 1. Decimalni brojevi su sastavljeni od 10 decimalnih cifara

0 1 2 3 4 5 6 7 8 9

Za razliku od toga, u binarnim brojevima se ne pojavljuje ovih deset cifara. Svi su oni sastavljeni samo od dve binarne cifre

0 1

Oktalni brojevi su sastavljeni od osam oktalnih cifara

0 1 2 3 4 5 6 7

Za heksadecimalne brojeve potrebno je šesnaest cifara. Znači, treba nam šest dodatnih simbola. Obično se za šest cifara koje dolaze posle cifre 9 koriste velika slova od A do F. Prema tome, heksadecimalni brojevi su sastavljeni od cifara

0 1 2 3 4 5 6 7 8 9 A B C D E F

„Binarna cifra“ (dakle, 1 ili 0) obično se naziva **bit**. Na slici A-2, decimalan broj 2001 izražen je u binarnom, oktalnom, decimalnom i heksadecimalnom obliku. Broj 7B9 je očigledno heksadecimalan jer se simbol B može pojaviti samo među heksadecimalnim brojevima. Međutim, broj 111 može poticati iz bilo kog od četiri pominjana sistema. Da bi se izbegla dvosmislenost kada brojčani sistem nije jasan iz konteksta, u indeks se stavljaju osnove: 2, 8, 10 ili 16.

Binarni 1 1 1 1 1 0 1 0 0 0 1

$$1 \times 2^{10} + 1 \times 2^9 + 1 \times 2^8 + 1 \times 2^7 + 1 \times 2^6 + 0 \times 2^5 + 1 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$$

$$1024 + 512 + 256 + 128 + 64 + 0 + 16 + 0 + 0 + 0 + 1$$

Oktalni 3 7 2 1

$$3 \times 8^3 + 7 \times 8^2 + 2 \times 8^1 + 1 \times 8^0$$

$$1536 + 448 + 16 + 1$$

Decimalni 2 0 0 1

$$2 \times 10^3 + 0 \times 10^2 + 0 \times 10^1 + 1 \times 10^0$$

$$2000 + 0 + 0 + 1$$

Heksadecimalni 7 D 1

$$7 \times 16^2 + 13 \times 16^1 + 1 \times 16^0$$

$$1792 + 208 + 1$$

Slika A-2. Broj 2001 u binarnom, oktalnom, decimalnom i heksadecimalnom obliku.

Kao primer binarnog, oktalnog, decimalnog i heksadecimalnog označavanja, pogledajte sliku A-3, gde su odabrani nenegativni brojevi prikazani u svakom od četiri sistema. Možda će neki budući arheolozi jednom naleteti na nju i postupati s njom kao s trojezičnom stelom iz Rozete da bi saznali kakvi su se to brojčani sistemi koristili na prelazu iz drugog u treći milenijum.

Decimalni	Binarni	Oktalni	Heksadecimalni
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	7
8	1000	10	8

Slika A-3. Decimalni brojevi i njihovi binarni, oktalni i heksadecimalni ekvivalenti.

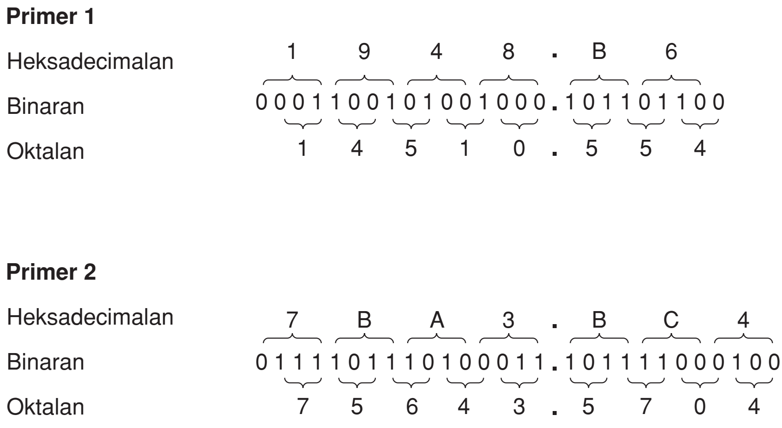
Decimalni	Binarni	Oktalni	Heksadecimalni
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F
16	10000	20	10
20	10100	24	14
30	11110	36	1E
40	101000	50	28
50	110010	62	32
60	111100	74	3C
70	1000110	106	46
80	1010000	120	50
90	1011010	132	5A
100	11001000	144	64
1000	1111101000	1750	3E8
2989	101110101101	5655	BAD

Slika A-3. Decimalni brojevi i njihovi binarni, oktalni i heksadecimalni ekvivalenti. *(nastavak)*

A.3 PRETVARANJE BROJEVA IZ JEDNOG BROJČANOG SISTEMA U DRUGI

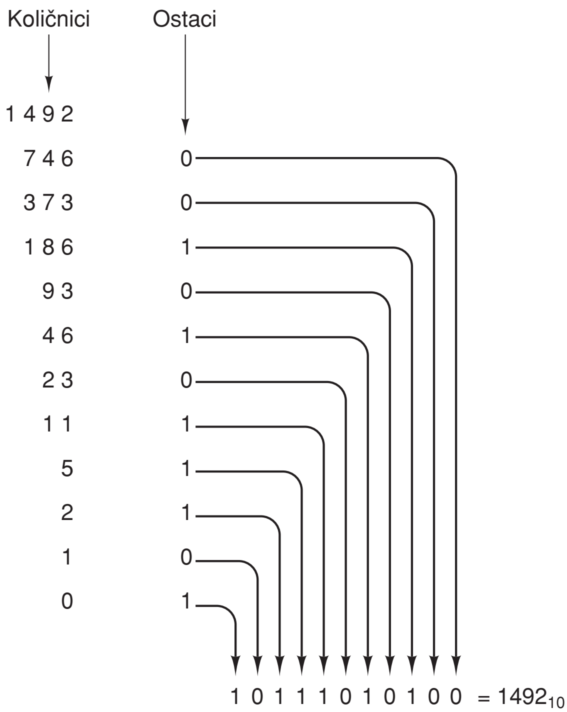
Nije teško pretvoriti brojeve iz oktalnog ili heksadecimalnog sistema u binarni i obrnuto. Da biste binarni broj preveli u oktalni, podelite ga u grupe od po 3 bita – prvu grupu čine 3 bita levo (ili desno) od decimalnog zareza (često zvanog binarni zarez), a drugu grupu 3 levo od nje. Svaka grupa od po 3 bita može se direktno prevesti u jednu oktalnu cifru (od 0 do 7), prema primerima iz prvih redova sa slike A-3. Pri podeli na grupe možda će biti potrebno da se neke od njih dopune vodećim ili dodatnim nulama do 3 cifre. Pretvaranje oktalnog broja u binarni isto je tako jednostavno. Svaku oktalnu cifru treba zameniti ekvivalentnim trocifrenim binarnim brojem. Pretvaranje heksadecimalnih brojeva u binarne suštinski je isto kao i pretvaranje oktalnih u binarne, s tim što se svaka heksadecimalna cifra zamenjuje grupom od po 4 bita. Na slici A-4 su neki primeri.

Pretvaranje decimalnih brojeva u binarne može se obaviti na dva načina. Prvi način sledi direktno iz definicije binarnih brojeva. Od broja se oduzme najveći stepen dvojke koji je manji od broja. Postupak se zatim ponavlja nad razlikom. Kada se broj rastavi na stepene dvojke, binarni broj se sklapa tako što se cifra 1 stavlja na pozicije bitova koji odgovaraju stepenima dvojke iskorišćenim za razlaganje a cifra 0 na ostala mesta.



Slika A-4. Primeri pretvaranja brojeva iz oktalnog i heksadecimalnog sistema u binarni.

Drugi način (važi samo za cele brojeve) svodi se na deljenje broja sa 2. Količnik se upisuje direktno ispod originalnog broja, a ostatak (0 ili 1) pored količnika. Postupak se ponavlja sve dok se na kraju ne dobije 0. Rezultat ovog postupka su dve kolone brojeva – količnika i ostataka. Binaran broj se sada direktno može očitati iz kolone sa ostacima, čitajući odozdo. Slika A-5 je primer pretvaranja decimalnog broja u binarni.

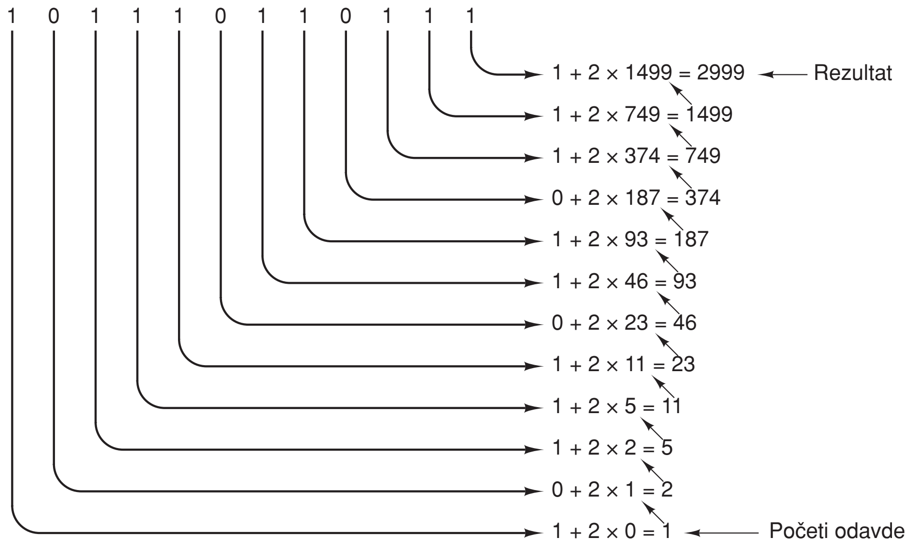


Slika A-5. Pretvaranje decimalnog broja 1492 u binaran uzastopnim deljenjem, počinjući od vrha i idući ka dnu. Na primer, 93 podeljeno sa 2 daje 46 uz ostatak 1, što se upisuje u red ispod.

Binarni celi brojevi mogu se pretvarati u decimalne na dva načina. Jedan način je sabiranje stepena dvojke koji odgovaraju bitovima 1 u broju. Na primer,

$10110 = 2^4 + 2^2 + 2^1 = 16 + 4 + 2 = 22$

Prema drugom načinu, binarni broj se piše vertikalno, po jedan bit u svakom redu, s krajnjim levim bitom u dnu stupca. Najniži red je prvi, onaj iznad njega drugi itd. Decimalni broj se gradi u paralelnom stupcu. Počinje se upisivanjem cifre 1 u prvi red. Upis u red n sastoji se od dva upisa iz reda n - 1 i bita iz reda n (0 ili 1). Upis u najviši red daje odgovor. Slika A-6 prikazuje primer pretvaranja binarnog broja u decimalan na ovaj način.



Slika A-6. Pretvaranje binarnog broja 101110110111 u decimalan uzastopnim udvostručavanjem, počevši od dna. Svaki red se dobija tako što se onaj ispod njega udvostruči i na to doda odgovarajući bit. Na primer, 749 je dva puta 374 plus bit 1 u istom redu gde je i 749.

Pretvaranje iz decimalnog u oktalni ili iz decimalnog u heksadecimalni sistem može se izvesti tako što se broj pretvori prvo u binarni, a zatim u broj željenog sistema ili tako što se oduzimaju stepeni broja 8, odnosno 16.

A.4 NEGATIVNI BINARNI BROJEVI

Tokom istorije digitalnih računara, u njima su korišćena četiri različita sistema za predstavljanje negativnih brojeva. Prvo su bili **označeni moduli** (engl. *signed magnitude*). U ovom sistemu krajnji levi bit bio je bit znaka (0 je značilo +, a 1 je značilo -), dok su ostali bitovi sadržali apsolutnu vrednost (modul) broja.

Drugi sistem, zvan **komplement jedinice** (engl. *one's complement*), takođe je imao bit za znak, gde je 0 značilo plus, a 1 minus. Kada od pozitivnog želite da napravite negativan broj, samo jedinice zamenite nulama, a nule jedinicama, što važi i za bit znaka. Sistem s komplementom jedinice je zastareo.

Treći sistem, zvan **komplement dvojke** (engl. *two's complement*), takođe ima bit za znak, gde 0 stoji za plus, a jedan za minus. Pretvaranje pozitivnog u negativan broj ovde je dvostepen postupak. Prvo se svaka jedinica zameni nulom, a svaka nula jedinicom, kao u sistemu komplementa jedinice. Zatim se rezultatu doda 1. Binarni brojevi se sabiraju isto kao i decimalni, s tim što se prenos na više mesto događa kada zbir premaši 1, a ne 9, što bi bilo u decimalnom sistemu. Na primer, pretvaranje broja 6 u -6 u komplementu dvojke obavlja se u dva koraka:

00000110 (+6)
11111001 (- 6 u komplementu jedinice)
11111010 (- 6 u komplementu dvojke)

Ako se operacijom na krajnjem levom bitu stvore uslovi za prenos, on se zanemaruje.

U četvrtom sistemu, koji se za *m*-bitne brojeve zove **višak 2^{m-1}** (engl. *excess* 2^{m-1}), broj se čuva kao zbir dobijen njegovim uvećanjem za 2^{m-1}. Na primer, za 8-bitne brojeve ($m = 8$), sistem se zove višak 128 i brojevi se čuvaju uvećani za 128. Prema tome, - 3 postaje - 3 + 128 = 125, a - 3 se predstavlja 8-bitnim binarnim ekvivalentom broja 125 (01111101). Brojevi između - 128 i +127 preslikavaju se u interval od 0 do 255 i svi se mogu izraziti kao 8-bitni pozitivni brojevi. Zanimljivo je da je ovaj sistem identičan sistemu komplementa dvojke kod koga je bit znaka invertovan. Slika A-7 prikazuje primere negativnih brojeva u sva četiri sistema.

Sistem označenog modula i sistem komplementa jedinice predstavljaju nulu na dva načina: kao pozitivnu nulu i kao negativnu nulu. Takvo stanje nije poželjno. Sistem komplementa dvojke nema taj nedostatak jer je u njemu pozitivna nula u komplementu dvojke takođe pozitivna nula. Sistem komplementa dvojke, međutim, ima drugu jedinstvenu osobinu. Niz bitova koji počinje jedinicom, a iza nje su nule, predstavlja svoj sopstveni komplement. Zbog toga opsezi pozitivnih i negativnih brojeva nisu simetrični; postoji negativan broj koji nema svog pozitivnog parnjaka.

Nije teško pronaći otkud ovakvi problemi: želimo da napravimo sistem kodiranja koji ima sledeća dva svojstva:

1. Predstavljanje nule na samo jedan način.
2. Tačno isti broj pozitivnih i negativnih brojeva.

Problem nastaje zato što svaki skup sa istim brojem pozitivnih i negativnih brojeva i samo jednom nulom ima neparan broj članova, dok *m* bitova omogućava paran broj rasporeda bitova. Uvek će jedan specifičan raspored bitova biti višak ili će nedostajati, bez obzira na način prikazivanja. Taj jedan suvišan raspored bitova može se upotrebiti za predstavljanje - 0 ili velikog negativnog broja (ili nečega sasvim drugog) ali će uvek zadavati glavobolje.

N decimalno	N binarno	-N označeni modul	-N komplement jedinice	-N komplement dvojke	-N višak 128
1	00000001	10000001	11111110	11111111	01111111
2	00000010	10000010	11111101	11111110	01111110
3	00000011	10000011	11111100	11111101	01111101
4	00000100	10000100	11111011	11111100	01111100
5	00000101	10000101	11111010	11111011	01111011
6	00000110	10000110	11111001	11111010	01111010
7	00000111	10000111	11111000	11111001	01111001
8	00001000	10001000	11110111	11111000	01111000
9	00001001	10001001	11110110	11110111	01110111
10	00001010	10001010	11110101	11110110	01110110
20	00010100	10010100	11101011	11101100	01101100
30	00011110	10011110	11100001	11100010	01100010
40	00101000	10101000	11010111	11011000	01011000
50	00110010	10110010	11001101	11001110	01001110
60	00111100	10111100	11000011	11000100	01000100
70	01000110	11000110	10111001	10111010	00111010
80	01010000	11010000	10101111	10110000	00110000
90	01011010	11011010	10100101	10100110	00100110
100	01100100	11100100	10011011	10011100	00011100
127	011111111	111111111	10000000	10000001	00000001
128	Ne postoji	Ne postoji	Ne postoji	10000000	00000000

Slika A-7. Negativni 8-bitni brojevi u četiri sistema.

A.5 BINARNA ARITMETIKA

Tabela za sabiranje binarnih brojeva prikazana je na slici A-8.

Prvi sabirak	0	0	1	1
Drugi sabirak	+0	+1	+0	+1
Zbir	0	1	1	0
Prenos	0	0	0	1

Slika A-8. Tabela binarnog sabiranja.

Dva binarna broja se sabiraju tako što se – počevši od krajnjeg desnog bita – sabiraju odgovarajući bitovi dva sabirka. Eventualni višak se prenosi jedno mesto ulevo, kao u aritmetici decimalnih brojeva. U aritmetici komplementa jedinice, višak stvoren pri sabiranju krajnjih levih bitova prenosi se u krajnji desni bit. Ovaj postupak se zove cirkularni prenos. U aritmetici komplementa dvojke, zanemaruje se višak stvoren sabiranjem dva krajnja leva bita. Primeri binarne aritmetike prikazani su na slici A-9.

Decimalni sistem	Komplement jedinice	Komplement dvojke
10	00001010	00001010
+ (-3)	11111100	11111101
+7	1 00000110	1 00000111
	prenos 1	odbačeno
	00000111	

Slika A-9. Sabiranje u sistemu komplementa jedinice i u sistemu komplementa dvojke.

Kad su dva sabirka različitog znaka, ne nastaje prekoračenje. Ako su oni istog znaka, a rezultat je suprotnog znaka, nastaje prekoračenje i dobija se pogrešan odgovor. U sistemu komplementa jedinice i u sistemu komplementa dvojke, prekoračenje se javlja samo ako se prenos u bit znaka razlikuje od prenosa iz bita znaka. Većina računara čuva prenos generisan iz bita za znak, ali se prenos u bit znaka ne vidi iz rezultata. Stoga se obično koristi i specijalan bit prekoračenja.

VEŽBANJA

1. Napišite binarne ekvivalente sledećih decimalnih brojeva: 1984, 4000, 8192.

2. Koji je decimalni ekvivalent binarnog broja 1001101001, a koji su njegovi oktalni i heksadecimalni ekvivalenti?

3. Koji su od ovih zapisa punovažni heksadecimalni brojevi? BED, CAB, DEAD, DECADE, ACCEDED, BAG, DAD.

4. Decimalni broj 100 izrazite u brojčanim sistemima sa osnovama od 2 do 9.

5. Koliko se različitih pozitivnih brojeva može izraziti pomoću *k* cifara u brojčanom sistemu osnove *r*?

6. Većina ljudi ume da računa do 10 na prste; međutim, kompjuteraši umeju i više. Ako svaki prst posmatrate kao jedan binarni bit, pa ispružen prst znači 1, a savijen prst znači 0, do koliko možete brojati sa obe ruke? A sa obe ruke i obe noge? Sada upotrebite obe ruke i obe noge, a palac vaše leve noge neka bude bit za znak u komplementu dvojke. Koliko brojeva možete na taj način predstaviti?

7. Izvršite sledeća izračunavanja sa 8-bitnim brojevima u komplementu dvojke.

00101101	11111111	00000000	11110111
+ 01101111	+ 11111111	- 11111111	- 11110111

8. Ponovite računicu iz prethodnog vežbanja, ali sada u komplementu jedinice.

9. Razmotrite sledeća sabiranja 3-bitnih binarnih brojeva u komplementu dvojke. Za svaki zbir navedite
a. Da li je bit znaka u rezultatu jednak 1.
b. Da li su najmanje značajna tri bita 0.
c. Da li je nastalo prekoračenje.

$$\begin{array}{r} 000 \\ +\ \underline{001} \end{array} \qquad \begin{array}{r} 000 \\ +\ \underline{111} \end{array} \qquad \begin{array}{r} 111 \\ +\ \underline{110} \end{array} \qquad \begin{array}{r} 100 \\ +\ \underline{111} \end{array} \qquad \begin{array}{r} 100 \\ +\ \underline{100} \end{array}$$

10. Označeni decimalni brojevi sa *n* cifara mogu se predstaviti sa *n* + 1 cifara bez znaka. Krajnja leva cifra pozitivnih brojeva je 0. Negativni brojevi se formiraju tako što se svaka cifra oduzme od 9. Tako se od broja 014725 dobija njegov negativni parnjak 985274. Takvi brojevi su „komplementi devetke" i analogni su binarnim brojevima u komplementu jedinice. Izrazite sledeće brojeve kao trocifrene komplemente devetke: 6, - 2, 100, - 14, - 1, 0.

11. Formulišite pravilo za sabiranje brojeva u komplementu devetke a zatim obavite sledeća sabiranja.

$$\begin{array}{r} 0001 \\ +\ \underline{9999} \end{array} \qquad \begin{array}{r} 0001 \\ +\ \underline{9998} \end{array} \qquad \begin{array}{r} 9997 \\ +\ \underline{9996} \end{array} \qquad \begin{array}{r} 9241 \\ +\ \underline{0802} \end{array}$$

12. Komplement desetke analogan je komplementu dvojke. Negativan broj u komplementu desetke formira se dodavanjem jedinice na odgovarajući broj u komplementu desetke, uz zanemarivanje eventualnog prenosa. Kako glasi pravilo za sabiranje komplemenata desetke?

13. Konstruišite tablice množenja brojeva sa osnovom 3.

14. Pomnožite binarno 0111 i 0011.

15. Napišite program koji prihvata označen decimalni broj kao znakovni ASCII niz i štampa binarni broj u komplementu dvojke, kao oktalan i kao heksadecimalan broj.

16. Napišite program koji preuzima dva ASCII niza od po 32 znaka koji sadrže nule i jedinice (svaki predstavlja 32-bitni binarni broj u komplementu dvojke). Program treba da odštampa njihov zbir kao znakovni ASCII niz sa 32 nule i(ili) jedinice.