UVOD U ORGANIZACIJU I ARHITEKTURU RAČUNARA 1
(skripta za M, N, V i L smer)

Stefan Mišković
Matematički fakultet
Beograd

1 Brojni sistemi

Svi brojni sistemi se dele u dve grupe – nepozicioni i pozicioni. Kod nepozicionih brojnih sistema svaka cifra ukazuje na istu vrednost bez obzira na kom se mestu nalazi, dok kod pozicionih može imati različite vrednosti. Na primer, kod dekadnog broja 2012 (pozicioni brojni sistem) cifra 2 može imati vrednost 2 ili vrednost 2000, dok kod rimskog broja MMXII (nepozicioni brojni sistem) cifra I uvek ima vrednost 1, a cifra M uvek 1000. U sledećoj tabeli su prikazani neki pozicioni brojni sistemi sa odgovarajućim osnovama, skupom cifara i primerom zapisa jednog broja.

Naziv brojnog sistema	**Osnova**	**Skup cifara**	**Primer**
dekadni	10	$0, 1, 2, 3, 4, 5, 6, 7, 8, 9$	$(31483)_{10}$
binarni	2	$0, 1$	$(101101)_2$
oktalni	8	$0, 1, 2, 3, 4, 5, 6, 7$	$(70163)_8$
heksadekadni	16	$0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F$	$(C1A34)_{16}$
troični	3	$0, 1, 2$	$(20112)_3$
balansirani troični	3	$-1, 0, 1$	$(10(-1)10(-1)01)_{bt}$
binarni	2	$0, 1$	$(101101)_2$
sa osnovom -10	-10	$0, 1, 2, 3, 4, 5, 6, 7, 8, 9$	$(5296)_{-10}$
negabinarni	-2	$0, 1$	$(1010)_2$
sa osnovom 0.5	0.5	$0, 1, 2, 3, 4, 5, 6, 7, 8, 9$	$(439)_2$

1. Prevesti sledeće cele brojeve iz naznačenog brojnog sistema u dekadni: (a) $(10110)_2$, (b) $(3127)_8$, (c) $(3129)_{16}$, (d) $(1A3)_{16}$, (e) $(212001)_3$, (f) $(10(-1)(-1)00)_{bt}$, (g) $(101101)_{-2}$, (h) $(145)_{0.5}$.

Rešenje: (a) Broj se prevodi tako što se svakoj cifri dodeli vrednost pozicije, koja zavisi od datog sistema. U ovom slučaju cifre sleva nadesno imaju vrednosti 2^4, 2^3, 2^2, 2^1 i 2^0, pa je $(10110)_2 = 1\cdot2^4+0\cdot2^3+1\cdot2^2+1\cdot2^1+0\cdot2^0 = 16+4+2 = (22)_{10}$. (b) $(3127)_8 = 2\cdot8^3+1\cdot8^2+2\cdot8^1+7\cdot8^0 = 3\cdot512+1\cdot64+2\cdot8+7\cdot1 = (1623)_{10}$. (c) $(3129)_{16} = 3\cdot16^3+1\cdot16^2+2\cdot16^1+9\cdot16^0 = 3\cdot4096+1\cdot256+2\cdot16+9\cdot1 = (12585)_{10}$. (d) Kod heksadekadnog sistema se cifri A dodeljuje vrednost 10 (a ciframa B, C, D, E i F vrednosti 11, 12, 13, 14 i 15), pa je $(1A3)_{16} = 1\cdot16^2+10\cdot16^1+3\cdot16^0 = 256+160+3 = (419)_{10}$. (e) $(212001)_3 = 2\cdot3^5+1\cdot3^4+2\cdot3^3+1\cdot3^0 = (622)_{10}$. (f) $(10(-1)(-1)00)_{bt} = 1\cdot3^5+0\cdot3^4+(-1)\cdot3^3+(-1)\cdot3^2+0\cdot3^1+0\cdot3^0 = (207)_{10}$. (g) $(101101)_{-2} = 1\cdot(-2)^5+0\cdot(-2)^4+1\cdot(-2)^3+1\cdot(-2)^2+0\cdot(-2)^1+1\cdot(-2)^0 = (-35)_{10}$. (h) $(145)_{0.5} = 1\cdot0.5^2+4\cdot0.5^1+5\cdot0.5^0 = 0.25+2+5 = (7.25)_{10}$.

Druga varijanta algoritma za prevođenje brojeva iz drugih brojnih sistema u dekadni je Horneova šema. Na primer, za broj $abcde$ u osnovi 7, važi da je

$$\begin{aligned}(abcde)_7 &= a\cdot7^4+b\cdot7^3+c\cdot7^2+d\cdot7+e\\ &= (a\cdot7^3+b\cdot7^2+c\cdot7+d)\cdot7+e\\ &= ((a\cdot7^2+b\cdot7+c)\cdot7+d)\cdot7+e\\ &= (((a\cdot7+b)\cdot7+c)\cdot7+d)\cdot7+e,\end{aligned}$$

pa se on može prevesti u dekadni sistem koristeći jednakost $(abcde)_7 = a\cdot7^4+b\cdot7^3+c\cdot7^2+d\cdot7+e$ ili pomoću Horneove šeme, tj. koristeći jednakost $(abcde)_7 = (((a\cdot7+b)\cdot7+c)\cdot7+d)\cdot7+e$.

2. Prevesti sledeće cele brojeve u dekadni sistem pomoću Horneove šeme: (a) $(3127)_8$, (b) $(3129)_{16}$.

Rešenje: (a) $(3127)_8 = ((3 \cdot 8 + 1) \cdot 8 + 2) \cdot 8 + 7 = (25 \cdot 8 + 2) \cdot 8 + 7 = 202 \cdot 8 + 7 = (1623)_8$. (b) $(3129)_{16} = ((3 \cdot 16 + 1) \cdot 16 + 2) \cdot 16 + 9 = (12585)_{10}$.

3. Prevesti sledeće cele brojeve iz dekadnog u navedeni brojni sistem: (a) 3129 u sistem sa osnovom 4, (b) 3129 u heksadekadni, (c) 23 u binarni, (d) 76 u sistem sa osnovom 3, (e) 146222 u heksadekadni.

Rešenje: (a) Broj se iz dekadnog prevodi u sistem sa osnovom n tako što se u svakom koraku deli brojem n, pri čemu se pamti količnik i ostatak. Zatim se njegov količnik deli brojem m i postupak se ponavlja sve dok količnik ne postane 0. Na kraju ostaci spojeni u obrnutom redosledu daju cifre broja. U prvom redu tabele u primeru su početni broj u dekadnom sistemu i količnici, a u drugom odgovarajući ostaci pri deljenju.

3129	782	195	48	12	3
1	2	3	0	0	3

Iz tabele sledi da je $(3129)_{10} = (300321)_4$. (b) Slično iz

3129	195	12
9	3	C

sledi da je $(3129)_{10} = (C39)_{16}$. Treba primetiti da se ostaci u ovom slučaju tretiraju kao heksadekadne cifre. (c) $(23)_{10} = (10111)_2$. (d) $(76)_{10} = (2211)_3$. (e) $(146222)_{10} = (23B2E)_{16}$.

4. Prevesti sledeće mešovite brojeve iz naznačenog sistema u dekadni: (a) $(1101.101)_2$, (b) $(233.12)_8$, (c) $(0.42)_5$.

Rešenje: (a) Mešoviti broj 1101.101 se prevodi tako što se ciframa desno od decimalne tačke dodele vrednosti pozicija 2^{-1}, 2^{-2} i 2^{-3}, nakon se dobijeni rezultat sabere sa revodom binarnog celog broja 1101. Odatle je $(1101.101)_2 = 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 + 1 \cdot 2^{-1} + 0 \cdot 2^{-2} + 1 \cdot 2^{-3} = 8 + 4 + 1 + 0.5 + 0.125 = (13.625)_{10}$. (b) $(233.12)_8 = 2 \cdot 8^2 + 3 \cdot 8^1 + 3 \cdot 8^0 + 1 \cdot 8^{-1} + 2 \cdot 8^{-2} = 128 + 24 + 3 + 0.125 + 0.03125 = (155.15625)_{10}$. (c) $(0.42)_5 = 4 \cdot 5^{-1} + 2 \cdot 2^{-2} = 0.8 + 0.08 = (0.88)_{10}$.

5. Prevesti sledeće razlomljene i mešovite brojeve iz dekadnog u naznačeni sistem: (a) 0.84375 u binarni, (b) 12.375 u binarni, (c) 0.4 u binarni, (d) 77.44 u sistem sa osnovom 5.

Rešenje: (a) Razlomljeni broj 0.84375 se iz dekadnog prevodi u sistem sa osnovom m tako što se u svakom koraku množi sa m, pri čemu se pamte celi i razlomljeni deo. U narednom koraku se sledeći razlomljeni deo množi sa m i postupak se ponavlja sve dok se ne dostigne vrednost 0. Rezultat je niz dobijenih celih delova. Na primer, iz

0.84375	0.6875	0.375	0.75	0.5	0
0	1	1	0	1	1

sledi da je $(0.84375)_{10} = (0.11011)_2$. (b) Mešoviti broj 12.375 se prevodi u binarni sistem spajanjem prevođenja celog i razlomljenog broja 12 i 0.375. Kako je $(12)_{10} = (1100)_2$ i $(0.375)_{10} = (0.011)_2$, to je $(12.375)_{10} = (1100.011)_2$. (c) U nekim primerima se može dogoditi da se ovim postupkom ne može stići do nule, jer se uočava ponavljanje odgovarućeg razlomljenog, odnosno celog, dela. U takvim primerima vrednost u jednom sistemu se može zapisati konačnim brojem cifara, a u drugom je zapis periodičan. Tako za $(0.4)_{10}$ važi

0.4	0.8	0.6	0.2	0.4	0.8	0.6	0.2	0.4	0.8
0	0	1	1	0	0	1	1	0	0

pa je $(0.4)_{10} = (0.011001100...)_2 = (0.\overline{0110})_2$. Kako je

0.44	0.2	0
0	2	1

i $(77)_{10} = (302)_5$, to je $(77.44)_{10} = (302.21)_5$.

Nekad se jednostavno može izvršiti prevođenje iz sistema sa osnovom n u sistem sa osnovom m bez međuprevođenja u dekadni sistem. U tom slučaju broj m mora biti stepen broja n ili broj n mora biti stepen broja m. Na primer, za $n = 2$ i $m = 16$ ako bismo želeli da broj $(101101.01)_2$ prevedemo u heksadekadni sistem, potrebno je da pocevši od decimalne tačke i krećući se levo i desno izdvajamo grupe od po 4 cifre, pri čemu ukoliko odgovarajuća poslednja grupa ima manje od 4 cifre, potrebno ih je dopuniti nulama u istom smeru i zatim svaku grupu od po 4 cifre ponaosob pretvoriti u heksadekadnu cifru. Drugim rečima, $(101101.01)_2 = (10|1101.01)_2 = (0010|1101.0100)_2 = (2D.4)_{16}$. U obrnutom smeru je potrebno, nakon pretvaranja heksadekadnih cifara u četvorke binarnih, izbrisati eventualne nule na početku i kraju broja. U ovom primeru je $(2D.4)_{16} = (0010|1101.0100)_2 = (101101.01)_2$. Analogno se princip može uopštiti, pri čemu, ako je $m = n^k$, uvek se posmatraju grupe od po k cifara (u ovom slučaju 4, jer je $16 = 2^4$).

6. Izvršiti sledeća prevođenja brojeva bez međuprevođenja u dekadni sistem: (a) $(10110001.0101101)_2 = (...)_8$, (b) $(C1.F1F9)_{16} = (...)_4$, (c) $(D2.EA5)_{16} = (...)_2$.

Rešenje: (a) $(10110001.0101101)_2 = (010|110|001.010|110|100)_2 = (261.264)_8$. (b) $(C1.F1F9)_{16} = (30|01.33|01|33|21)_4$ $= (3001.33013321)_4$. (c) $(D2.EA5)_{16} = (1101|0010.1110|1010|0101)_2 = (11010010.111010100101)_2$.

Primetimo da se ne može na ovaj način direktno prevoditi iz osnove 16 u osnovu 8 (ni obratno), budući da 16 nije stepen broja 8. Moguće je izvršiti međuprevođenje u sistem sa osnovom 2.

2 Zapis označenih brojeva u računaru

Označeni brojevi se u računaru mogu zapisivati na više načina, a za svaki zapis je zajedničko da se mora voditi računa o znaku i apsolutnoj vrednosti broja. Znak se u svakom zapisu piše pomoću jedne cifre na isti način − ukoliko je broj pozitivan, dodeljuje mu se najmanja cifra sistema (tj. 0), a ukoliko je negativan najveća (za binarne 1, za dekadne 9, za heksadekadne F). Apsolutna vrednost broja se u zavisnosti od sistema tretira na različite načine. U nastavku će biti pomenuta četiri načina zapisa označenih brojeva u računaru: znak i apsolutna vrednost, nepotpuni komplement, potpuni komplement i višak k.

Ako je broj zapisan u znaku i apsolutnoj vrednosti, tada je potrebno nakon znaka broja dopisati njegovu apsolutnu vrednost. Ukoliko je naglašeno da se broj zapisuje na više mesta nego što je potrebno, apsolutnu vrednost broja treba dopuniti nulama. Na primer, ako je potrebno broj $(27)_{10}$ napisati takođe u dekadnom sistemu u zapisu znak i apsolutna vrednost na 3 i 5 mesta, zapisi bi bili $(027)^3_{10}$ i $(00027)^3_{10}$. Slično, broj $(-27)_{10}$ se zapisuje kao $(927)^3_{10}$ i $(90027)^3_{10}$. Osnovna mana ovog sistema je što se nula može zapisati na dva načina, na primer na tri mesta u osnovi 6 se piše kao $(000)^3_6$ ili $(500)^3_6$.

Ako je broj zapisan u nepotpunom komplementu, tada je različit postupak za zapis pozitivnih i negativnih brojeva. Pozitivni brojevi se pišu na način na koji se zapisuju u znaku i apsolutnoj vrednosti, dok se negativni brojevi najpre zapišu kao odgovarajući pozitivni a zatim se izvrši njihovo komplementiranje do najveće cifre sistema (tj. oduzimanje od najveće cifre sistema). Na primer, broj $(27)_{10}$ se u nepotpunom komplementu na 3 i 5 mesta piše, analogno prethodnom, kao $(027)^3_{10}$ i $(00027)^3_{10}$. Ako bismo želeli da zapišemo broj $(-27)_{10}$ na 3 mesta, najpre ga zapišemo kao odgovarajući pozitivan, odnosno kao $(027)^3_{10}$. Zatim svaku cifru komplementiramo u odnosu na cifru 9, prema opisanom postupku. Dobijeni broj je $(972)^3_{10}$. Analogno, ako bismo želeli da zapišemo broj $(-27)_{10}$ na 5 mesta, preko broja $(00027)^3_{10}$ se dobija rešenje $(99972)^3_{10}$. Kod nepotpunog komplementa se nula takođe može zapisati na dva načina, na primer u osnovi 6 na tri mesta se piše kao $(000)^3_6$ ili $(555)^3_6$.

U potpunom komplementu se takođe posebno razmatraju slučajevi pozitivnih i negativnih brojeva. Pozitivni brojevi se pišu isto kao u prethodna dva zapisa, a negativni tako što se na poslednju cifru nepotpunog komplementa doda cifra 1. Ako pri tom sabiranju dođe do prekoračenja, poslednji prenos se ignoriše. Na primer, pozitivan broj $(27)_{10}$ se na 3 i 5 mesta u osnovi 10 piše kao $(027)^3_{10}$ i $(00027)^3_{10}$, a negativan $(-27)_{10}$ kao $(973)^3_{10}$ i $(99973)^3_{10}$. Primetimo da se kod ovog zapisa nula piše na jedinstven način. Kao pozitivan broj u osnovi 6 se na tri mesta piše kao $(000)^3_6$, dok se u negativnom slučaju, prema opisanom algoritmu, vrši dodavanje jedinice na broj $(555)^3_6$, pa se dobija zapis $(000)^3_6$ (kao što je naglašeno, poslednja jedinica u prenosu se ignoriše).

1. Zapisati sledeće brojeve u znaku i apsolutnoj vrednosti, nepotpunom komplementu i potpunom komplementu: (a) $(732)_{10} \to (...)_7$, (b) $(-1045)_{10} \to (...)_{16}$.

Rešenje: (a) Kako je broj dat u dekadnom sistemu, a u navednim zapisima je potrebno zapisati ga u osnovi 7, najpre je potrebno izvršiti prevođenje celog broja 732 iz osnove 10 u osnovu 7. Primenom poznatog algoritma se dobija da je $(732)_{10} = (2064)_7$. Zbog toga je zapis u znaku i apsolutnoj vrednosti oblika $(02064)^5_7$. Kako je broj pozitivan, isto se zapisuje i u nepotpunom i u potpunom komplementu. (b) Najpre je $(-1045)_{16} = (-415)_{16}$. Zapis u znaku i apsolutnoj vrednosti je, kako je najveća cifra heksadekadnog sistema F, $(F415)^4_{16}$. Za zapis u nepotpunom komplementu je potrebno svaku cifru broja $(0415)^4_{16}$ komplementirati do F, pa je zapis $(FBEA)^4_{16}$. U potpunom komplementu je $(FBEB)^4_{16}$.

2. Zapisati sledeće brojeve na navedeni broj mesta u znaku i apsolutnoj vrednosti, nepotpunom komplementu i potpunom komplementu: (a) $(197)_{10} \rightarrow (...)^{10}_2$, (b) $(-125)_{10} \rightarrow (...)^8_6$.

Rešenje: Prema opisanom algoritmu zapisi na najmanji potreban broj mesta su prikazani sledećom tabelom:

Broj	Znak i apsolutna vrednost	Nepotpuni komplement	Potpuni komplement
$(197)_{10}$	$(011000101)^9_2$	$(011000101)^9_2$	$(011000101)^9_2$
$(-125)_{10}$	$(5325)^4_6$	$(5230)^4_6$	$(5231)^4_6$

Svako od ovih rešenja je potrebno proširiti na potreban broj mesta. U znaku i apsolutnoj vrednosti, potrebno je dodeliti vodeće nule iza cifre za znak, a u ostalim zapisima je dovoljno cifru za znak kopirati dovoljan broj puta. Na taj način se dobijaju sledeća rešenja:

Broj	Znak i apsolutna vrednost	Nepotpuni komplement	Potpuni komplement
$(197)_{10}$	$(0011000101)^{10}_2$	$(0011000101)^{10}_2$	$(0011000101)^{10}_2$
$(-125)_{10}$	$(50005325)^8_6$	$(55555230)^8_6$	$(55555231)^8_6$

3. Zapisati mešoviti broj $(-84.6875)_10$ na 8 mesta u osnovi 8 u znaku i apsolutnoj vrednosti, nepotpunom komplementu i potpunom komplementu.

Rešenje: Najpre se prevođenjem u osnovu 8 dobija $(-84.6875)_{10} = (-124.54)_8$. Cifre razlomljenog dela broja se tretiraju kao i cifre celog dela. Zapis u znaku i apsolutnoj vrednosti je tako $(700124.54)^8_8$. Komplementiranjem cifara broja $(000124.54)^8_8$ dobija se da je zapis u nepotpunom komplementu $(777653.23)^8_8$. Za zapis u potpunom komplementu se cifra 1 dodaje na poslenju cifru zapisa, a ne nužno na poslednju cifru celog dela. Odatle je zapis u potpunom komplementu $(777653.24)^8_8$.

Kod sabiranja u potpunom komplementu se sabiraju sve cifre iz zapisa broja uključujući i cifru za znak. Eventualni prenos sa pozicije najveće težine se ignoriše. Prekoračenje se može prepoznati promenom cifre za znak rezultata: ako se sabiraju pozitivni brojevi i dobije negativan rezultat ili ako se sabiraju negativni brojevi i dobije pozitivan rezultat. Oduzimanje brojeva svodimo na sabiranje.

1. Izvršiti sledeće operacije u potpunom komplementu: (a) $(04321)^5_5 - (02013)^5_5$, (b) $(01101)^5_2 + (00110)^5_2$.

Rešenje: Rezultati su prikazani na donjoj slici. U primeru pod (a) se oduzimanje svodi na sabiranje, tj. važi $(04321)^5_5 - (02013)^5_5 = (04321)^5_5 + (42432)^5_5$, a u primeru pod (b) je došlo do prekoračenja jer se sabiranjem dva pozitivna broja nije dobio pozitivan broj (u ovom slučaju se dobio negativan broj, a nekad rezultat može biti i nekorektan zapis), a rezultat je $^*(10011)^5_4$.

$$\begin{array}{c} 04321 \\ 42432 \\ \hline 02303 \end{array} \qquad \begin{array}{c} 01101 \\ 00110 \\ \hline 10011 \end{array}$$

3 Množenje i deljenje

Neka su data dva neoznačena cela broja i neka je potrebno izračunati njihov proizvod. Ukoliko pretpostavimo da je za njihovo zapisivanje dovoljno 8 bitova (brojevi su po pretpostavci zapisani kao neoznačeni u binarnom sistemu), za zapis njihovog proizvoda će biti dovoljno 16 bitova. Nazovimo prvi činilac množenikom, a drugi množiocem. Raspolažemo sa tri osmobitna registra, A, M i P (registar je zapravo niska date dužine) i jednobitnim registrom C. Sada algoritam izgleda ovako:
(1) U registar M se upisuje množenik (neoznačen ceo broj zapisan sa 8 cifara), množilac u registar P, dok se u registre A i C upisuje 00000000 i 0, redom.
(2) Posmatramo bit množioca u registru P na mestu najmanje težine (krajnja desna cifra). Ako je taj bit 0, ne vrši se nikakva akcija. Inače ukoliko je taj bit 1, tada se vrši sabiranje brojeva koji su u registrima A i M i dobijeni rezultat postaje nova vrednost registra A.
(3) Vrši se logičko pomeranje udesno registra CAP (dakle, registri C, A i P se posmatraju kao jedinstven registar tako što se nadovežu jedan na drugi). Logičko pomeranje znači da će se svi bitovi osim poslednjeg pomeriti za jedno mesto udesno, poslednji će se izgubiti, a na mesto prvog bita sleva novodobijenog broja se dodaje 0. Na primer, logičko pomeranje udesno niza od šest bitova 100101 daje kao rezultat nisku 010010.

(4) Koraci (2) i (3) se ponavljaju sve dok se ne obrade svi bitovi u množiocu. (U našem slučaju, gde su činioci osmobitni, algoritam će se vršiti u 8 koraka.)
(5) Vrednost proizvoda je smešten u registar AP, koji posmatramo kao jedinstven registar nadovezanih registara A i P.

1. Izvršiti množenje $112 \cdot 9$ ako su brojevi zapisani kao neoznačeni celi brojevi u binarnom sistemu na 8 mesta.

Rešenje: Najpre je $(112)_{10} = (0111000)_2^8$ i $(9)_{10} = (00001001)_2^8$. Prvi činilac nazovimo množenikom i označimo ga sa M, a drugi množiocem i označimo ga sa P. U sledećoj tabeli je prikazano stanje registara C, A i P, pri čemu se podrazumeva da registar M uvek ima istu vrednost.

Korak	C	A	P	Komentar
Početak	0	00000000	00001001	Na početku je $A = 00000000$, $C = 0$, $M = 01110000$ i $P = 00001001$.
1	0	01110000	00001001	Kako je poslednja cifra P bila 1, vrši se sabiranje $A = A + M$.
	0	00111000	00000100	Registar CAP se logički pomera udesno.
2	0	00111000	00000100	Kako je poslednja cifra P bila 0, ne vrši se nikakva akcija.
	0	00011100	00000010	Registar CAP se logički pomera udesno.
3	0	00011100	00000010	Kako je poslednja cifra P bila 0, ne vrši se nikakva akcija.
	0	00001110	00000001	Registar CAP se logički pomera udesno.
4	0	01111110	00000001	Kako je poslednja cifra P bila 1, vrši se sabiranje $A = A + M$.
	0	00111111	00000000	Registar CAP se logički pomera udesno.
5	0	00111111	00000000	Kako je poslednja cifra P bila 0, ne vrši se nikakva akcija.
	0	00011111	10000000	Registar CAP se logički pomera udesno.
6	0	00011111	10000000	Kako je poslednja cifra P bila 0, ne vrši se nikakva akcija.
	0	00001111	11000000	Registar CAP se logički pomera udesno.
7	0	00001111	11000000	Kako je poslednja cifra P bila 0, ne vrši se nikakva akcija.
	0	00000111	11100000	Registar CAP se logički pomera udesno.
8	0	00000111	11100000	Kako je poslednja cifra P bila 0, ne vrši se nikakva akcija.
	0	00000011	11110000	Registar CAP se logički pomera udesno.

Rezultat množenja je sadržan u registru AP. Njegova dekadna vrednost iznosi $(1111110000)_2 = 2^9+2^8+2^7+2^6+2^5+2^4 = 512 + 256 + 128 + 64 + 32 + 16 = 1008$.

Algoritam za množenje označenih brojeva zapisanih u potpunom komplementu je poznat još i kao Butov algoritam. Za zapis ćemo koristiti osmobitne registre A, M, P i jednobitni P_{-1}. Vršićemo množenje dva označena cela broja zapisana u potpunom komplementu u binarnom sistemu sa 8 bitova. Algoritam se odvija u sledećih nekoliko koraka:
(1) Najpre se u registar M upisuje množenik, u registar P množilac, dok se registrima A i P_{-1} dodeljuju vrednosti 00000000 i 0, redom.
(2) Poredi se vrednost P_0 u bitu najmanje težine registra R (krajnji desni bit) i bita P_{-1}: (a) ako su te dve vrednosti jednake (kombinacije 00 i 11), ne vrši se nikakva akcija; (b) ako se bitovi razlikuju i kombinacija je 01, vrši se sabiranje $A = A + M$; (c) ako se bitovi razlikuju i kombinacija je 10, vrši se oduzimanje $A = A – M$.
(3) Vrši se aritmetičko pomeranje udesno registra APP_{-1}. Aritmetičko pomeranje udesno se razlikuje od logičkog jedino u tome što se na mesto krajnje leve cifre rezultata pomeranja dopisuje ona cifra koja je bila prva cifra broja čije pomeranje vršimo. (Dakle nije obavezno nula, kao što je to bio slučaj kod logičkog pomeranja.) Na primer, aritmetičko pomeranje udesno niza od šest bitova 100101 daje kao rezultat nisku 110010.
(4) Koraci (2) i (3) se ponavljaju sve dok se ne obrade svi bitovi u množiocu. (U našem slučaju, gde su činioci osmobitni, algoritam će se izvršavati u 8 koraka.)
(5) Vrednost proizvoda je smešten u registar AP, koji se posmatra kao jedinstven registar nadovezanih registara A i P.

2. Izvršiti množenje $103 \cdot (-13)$ ako su brojevi zapisani kao označeni celi brojevi u potpunom komplementu u binarnom sistemu na 8 mesta.

Rešenje: Ako sa M označimo množenik, a sa P množilac i brojeve 103 i -13 zapišemo u potpunom komplementu u binarnom sistemu na 8 mesta, dobija se $M = 01100111$ i $P = 11110011$, redom. Tada se, prema algoritmu, sadržaj datih registara menja na sledeći način:

Korak	A	P	P_{-1}	Komentar
Početak	00000000	11110011	0	Na početku je $A = 00000000$, $M = 01100111$, $P = 11110011$ i $P_{-1} = 0$.
1	10011001 11001100	11110011 11111001	0 1	Kako je $P_0P_{-1} = 10$, vrši se oduzimanje $A = A - M$. Registar APP_{-1} se aritmetički pomera udesno.
2	11001100 11100110	11111001 01111100	1 1	Kako je $P_0P_{-1} = 11$, ne vrši se nikakva akcija. Registar APP_{-1} se aritmetički pomera udesno.
3	01001101 00100110	01111100 10111110	1 0	Kako je $P_0P_{-1} = 01$, vrši se sabiranje $A = A + M$. Registar APP_{-1} se aritmetički pomera udesno.
4	00100110 00010011	10111110 01011111	0 0	Kako je $P_0P_{-1} = 00$, ne vrši se nikakva akcija. Registar APP_{-1} se aritmetički pomera udesno.
5	10101100 11010110	01011111 00101111	0 1	Kako je $P_0P_{-1} = 10$, vrši se oduzimanje $A = A - M$. Registar APP_{-1} se aritmetički pomera udesno.
6	11010110 11101011	00101111 00010111	1 1	Kako je $P_0P_{-1} = 11$, ne vrši se nikakva akcija. Registar APP_{-1} se aritmetički pomera udesno.
7	11101011 11110101	00010111 10001011	1 1	Kako je $P_0P_{-1} = 11$, ne vrši se nikakva akcija. Registar APP_{-1} se aritmetički pomera udesno.
8	11110101 11111010	10001011 11000101	1 1	Kako je $P_0P_{-1} = 11$, ne vrši se nikakva akcija. Registar APP_{-1} se aritmetički pomera udesno.

Dakle, rezultat je $AP = 1111101011000101$. Potrebno je dobijeni rezultat prevesti iz potpunog komplementa u dekadni zapis. Najlakši način da se izvrši prevođenje je da se vrednosti stepena dvojke na mestima gde su jedinice saberu, osim u slučaju gde cifra 1 odgovara znaku (prva cifra sleva), gde se vrši oduzimanje. Drugim rečima, rezultat je $2^0+2^2+2^6+2^7+2^9+2^{11}+2^{12}+2^{13}+2^{14}-2^{15} = 2^0+2^2+2^6+2^7+2^9-2^{11} = 1+4+64+128+512-2048 = -1339$. (Ako je broj pozitivan, na mestu za znak bi se nalazila cifra 0, pa se nikakvo oduzimanje ne bi vršilo.)

Kod algoritma za deljenje neoznačenih brojeva se koriste tri osmobitna registra, A, M i P. Inicijalno se u registar P upisuje deljenik, u registar M delilac, a u registar A 00000000. Zatim se sledeća dva koraka ponavljaju osam puta:
(1) Pomera se sadržaj registra AP ulevo za jedan bit. (Kod pomeranja ulevo se na krajnje desno mesto postavlja cifra 0).
(2) Ako je $A \geq M$, vrši se oduzimanje $A = A - M$, a u najniži bit registra P se upisuje cifra 1. Inače, ako je $A < M$, ne vrši se nikakva akcija.
Na kraju algoritma se količnik nalazi u registru P, a ostatak u registru A.
3. Izvršiti deljenje $24 \div 9$ ako su brojevi zapisani kao neoznačeni celi brojevi u binarnom sistemu na 8 mesta.

Rešenje: Najpre je $P = (24)_{10} \rightarrow (00011000)_2^8$ i $P = (9)_{10} \rightarrow (00001001)_2^8$. Registri A i P se tokom izvršavanja algoritma menjaju na sledeći način:

Korak	A	P	Komentar
Početak	00000000	00011000	Na početku je $A = 00000000$, $M = 00001001$ i $P = 00011000$.
1	00000000 00000000	00110000 00110000	Registar AP se pomera ulevo. Kako je $A < M$, ne vrši se nikakva akcija.
2	00000000 00000000	01100000 01100000	Registar AP se pomera ulevo. Kako je $A < M$, ne vrši se nikakva akcija.
3	00000000 00000000	11000000 11000000	Registar AP se pomera ulevo. Kako je $A < M$, ne vrši se nikakva akcija.
4	00000001 00000001	10000000 10000000	Registar AP se pomera ulevo. Kako je $A < M$, ne vrši se nikakva akcija.
5	00000011 00000011	00000000 00000000	Registar AP se pomera ulevo. Kako je $A < M$, ne vrši se nikakva akcija.
6	00000110 00000110	00000000 00000000	Registar AP se pomera ulevo. Kako je $A < M$, ne vrši se nikakva akcija.
7	00001100 00000011	00000000 00000001	Registar AP se pomera ulevo. Kako je $A \geq M$, izvršilo se oduzimanje $A = A - M$, a najniži bit P je 1.
8	00000110 00000110	00000010 00000010	Registar AP se pomera ulevo. Kako je $A < M$, ne vrši se nikakva akcija.

Prema algoritmu, količnik je u registru P i njegova vrednost je 2, a ostatak u registru A i iznosi 6.

Za deljenje brojeva zapisanih u binarnom sistemu u potpunom komplementu se koriste tri registra, A, M i P. Algoritam se odvija na sledeći način:
(1) U registar AP se upisuje deljenik kao broj zapisan u potpunom komplementu dužine 16. (Kod prethodna tri algoritma u registar A se upisivala vrednost 00000000, što ovde nije uvek slučaj. Ukoliko je broj negativan, registar A će počinjati nulama, a ako je pozitivan jedinicama.) Delilac se upisuje u registar M.
(2) Sadržaj registra AP se pomera ulevo za jedan bit.
(3) Ako je $|A| < |M|$, ne vrši se nikakva akcija. Ako je $|A| \geq |M|$, tada se vrši sabiranje $A = A + M$ ukoliko su brojevi A i M različitog znaka, a oduzimanje $A = A - M$ ukoliko su brojevi A i M istog znaka, dok se najniža cifra registra P postavlja na 1.
(4) Koraci (2) i (3) se ponavljaju onoliko puta koliko iznosi dužina registra P (u ovom slučaju 8). Po završetku algoritma ostatak se nalazi u registru A. Ako je znak deljenika i delioca isti, tada je vrednost količnika u registru P, a ako je različit, tada za količnik treba uzeti vrednost u registru P sa promenjenim znakom.

4. Izvršiti deljenje $24 \cdot (-9)$ ako su brojevi zapisani kao označeni celi brojevi u potpunom komplementu u binarnom sistemu.

Rešenje: Nakon zapisa brojeva 24 i -9 u potpunom komplementu na odgovarajući broj mesta, prema algoritmu, dobija se da je deljenik jednak $AP = 0000000000011000$, a delilac $M = 11110111$. Registri A i P se dalje menjaju na sledeći način:

Korak	A	P	Komentar
Početak	00000000	00011000	Na početku je $A = 00000000$, $M = 11110111$ i $P = 00011000$.
1	00000000	00110000	Registar AP se pomera ulevo.
	00000000	00110000	Kako je $\|A\| < \|M\|$, ne vrši se nikakva akcija.
2	00000000	01100000	Registar AP se pomera ulevo.
	00000000	01100000	Kako je $\|A\| < \|M\|$, ne vrši se nikakva akcija.
3	00000000	11000000	Registar AP se pomera ulevo.
	00000000	11000000	Kako je $\|A\| < \|M\|$, ne vrši se nikakva akcija.
4	00000001	10000000	Registar AP se pomera ulevo.
	00000001	10000000	Kako je $\|A\| < \|M\|$, ne vrši se nikakva akcija.
5	00000011	00000000	Registar AP se pomera ulevo.
	00000011	00000000	Kako je $\|A\| < \|M\|$, ne vrši se nikakva akcija.
6	00000110	00000000	Registar AP se pomera ulevo.
	00000110	00000000	Kako je $\|A\| < \|M\|$, ne vrši se nikakva akcija.
7	00001100	00000000	Registar AP se pomera ulevo.
	00000011	00000001	Kako je $\|A\| \geq \|M\|$ i $\mathrm{sgn}(A) \neq \mathrm{sgn}(M)$, sledi $A = A + M$ i $P_0 = 1$.
6	00000110	00000010	Registar AP se pomera ulevo.
	00000110	00000010	Kako je $\|A\| < \|M\|$, ne vrši se nikakva akcija.

Ostatak deljenja je sadržan u registru A i iznosi 6. Vrednost registra P je 2. Kako je znak deljenika i delioca različit, prema algoritmu se za vrednost količnika uzima vrednost registra P sa promenjenim znakom, pa je količnik jednak -2. (Zaista je $24 = (-9) \cdot (-2) + 6$.)

4 Predstavljanje teksta i slike u računaru

Tekst se obično zamišlja kao dvodimenzionalni objekat, ali se u računaru predstavlja kao jednodimenzionalni niz karaktera. Pritom je potrebno uvesti i specijalne karaktere koji označavaju prelazak u novi red, tabulator, kraj teksta, itd. Ti specijalni karakteri se u računaru tretiraju na isti način kao slova, cifre i ostali karakteri. Osnovna ideja kod zapisivanja teksta u računaru je da se svakom karakteru pridruži odgovarajući ceo broj na unapred dogovoren način. Uređena lista karaktera predstavljena svojim kodovima se naziva kodna strana.

Primeri kodnih strana (jednobajtni strandard):

- **ASCII.** U ASCII kodu se može zapisati 128 različitih karaktera. Svakom karakteru se dodeljuje odgovarajuća sedmobitna niska. ASCII je jednobajtni standard, pa se vodeća cifra zapisa svakog karaktera ne koristi. Na prva 33 mesta (od 0 do 32) predstavljeni su kontrolni karakteri, velika slova engleske abecede (od A do Z) na mestima od 65 do 90, mala slova engleske abecede (od a do z) od 97 do 122, a cifre (od 0 do 9) na mestima od 48 do 57.
- **ISO 8859-1 (Latin 1).** Pomoću ove kodne strane se može zapisati 256 različitih karaktera i svakom karakteru se dodeljuje osmobitna niska. Na prvih 128 mesta se poklapa sa ASCII kodom. Pomoću nje se zapisuju svi znakovi zapadnoevropskih latinica.

- **ISO 8859-2 (Latin 2).** Ova kodna strana ima slične karakteristike kao Latin 1, s tim što se u ovom slučaju mogu zapisati svi znakovi istočnoevropskih latinica, uključujući srpsku.
- **ISO 8859-5.** Ova kodna strana ima slične karakteristike kao Latin 1, s tim što se u ovom slučaju mogu zapisati svi znakovi istočnoevropskih ćirilica, uključujući srpsku.
- **Windows 1250 i Windows 1251.** Ove dve kodne strane su Majkrosoftovi standardi. Obe koriste 8 bitova za zapis svakog karaktera, pri čemu se prvih 128 mesta poklapa sa ASCII-jem. Pomoću Windows 1250 se mogu zapisati sva srpska latinična, a pomoću Windows 1251 sva srpska ćirilična slova.

Primeri kodnih strana (višebajtni strandard):

- **ISO 10646.** Ovo je četvorobajtni standard kod koga se prvih 2^{16} karaktera koriste kao osnovni višejezični skup karaktera, a ostali prostor je ostavljen kao proširenje za drevne jezike, celokupnu naučnu notaciju i slično.
- **Unicode UCS-2.** Ovaj standard svakog karakteru dodeljuje dvobajtni kôd. Prvih 256 karaktera se poklapa sa Latin 1 standardom, a na preostalim mestima su, između ostalih, obuhvaćeni karakteri današnjih jezika.
- **Unicode UTF-8.** UTF-8 algoritmom se svakom karakteru dodeljuje od 1 do 6 bajtova, s tim što se ASCII karakteri zapisuju pomoću jednog bajta.

Postoje dva osnovna modela predstavljanja boja – aditivni i subtraktivni. Aditivni model se primenjuje kada se boje grade dodavanjem komponenti svetlosti, kao što je slučaj kod monitora i projektora, a subtraktivni kada se boje grade odbijanjem svetlosti, npr. pri slikanju, štampanju i sl. RGB je aditivni model boja. Svaka boja se opisuje kroz tri vrednosti: deo crvene (R), deo zelene (G) i deo plave (B). Zastupljenost svake od ovih komponenti varira u intervalu od 0 do 255. Kod HSB modela boja, svaka boja se opisuje preko vrednosti sledeće tri komponente: ton (H), zasićenost (S) i osvetljenost (B). Ton se opisuje pomoću kruga od 360 stepeni, pri čemu $0° = 360°$ odgovara crvenoj, $120°$ zelenoj i $240°$ plavoj. Zasićenost se izražava u procentima, gde 0% predstavlja sivu, a 100% čistu jarku boju. Osvetljenost se takođe opisuje na skali od 0 do 100 procenata, pri čemu 0% predstavlja crnu, a 100% čistu svetlu boju.

1. Boja se u RGB modelu predstavlja u obliku $(200, 100, 175)$. Odrediti HSB model koji odgovara ovom RGB modelu.

Rešenje: Vidi se da je u datoj boji crvena komponenta najizraženija (200), a zatim slede plava (175) i zelena (100). Označimo $B_1 = 200$, $B_2 = 175$ i $B_3 = 100$. Otklon predstavlja odnos između dve najizraženije komponente i određuje se prema obrascu $H_0 = \frac{60(B_2-B_3)}{B_1-B_3}$. Ovde je $H_0 = \frac{60 \cdot 75}{100} = 45$. Ton je zbir (ili razlika) vrednosti tona za najizraženiju komponentu i vrednosti otklona prema drugoj najizraženijoj komponenti, pri čemu se u zavisnosti od položaja dve najizraženije komponente na krugu uzima zbir, odnosno razlika. Kako je $H(R) = 360$ i $H(B) = 240$, u ovom slučaju je $H = H(R) - H_0 = 360 - 45 = 315$. Zasićenost se određuje prema obrascu $S = \frac{B_1-B_3}{B_1}$, pa je $S = 0.5 = 50\%$. Osvetljenost je data sa $B = \frac{B_1}{256}$, pa je u ovom slučaju $B = 0.78125 = 78.125\%$. Dakle, dobija se model $(315, 50\%, 78.125\%)$.

2. Boja se u HSB modelu predstavlja u obliku $(320, 50\%, 70.3125\%)$. Odrediti RGB model koji odgovara ovom HSB modelu.

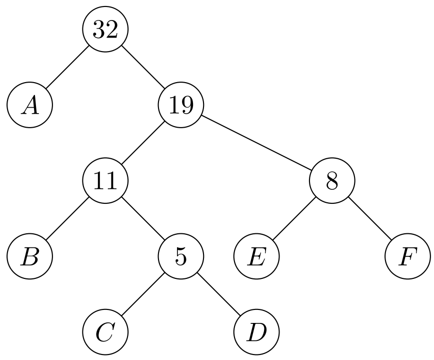
Rešenje: Iz primera se vidi da je $H = 320$, $S = 0.5$ i $B = 0.703125$. Kako je $B = \frac{B_1}{256}$, sledi $B_1 = 256B = 256 \cdot 0.703125 = 180$. Iz $S = \frac{B_1-B_3}{B_1}$ se dobija $B_1 - B_3 = SB_1 \Rightarrow B_3 = B_1(1 - S) = 180 \cdot 0.5 = 90$. Kako je $H = 320$, najbliže komponente su crvena i plava, a otklon se računa krećući se od $H(R)$ do $H(B)$ i iznosi $H_0 = H(R) - H = 360 - 320 = 40$. Najzad, iz $H_0 = \frac{60(B_2-B_3)}{B_1-B_3}$ sledi da je $B_2 = \frac{H_0(B_2-B_3)}{60} + B_3 = \frac{40 \cdot 90}{60} + 90 = 150$. Dakle, dobija se RGB model $(180, 90, 150)$.

5 Hafmanovo kodiranje

Za kôd kažemo da ima prefiksno svojstvo ako ni za jedan znak ne važi da je njegov kôd prefiks koda nekog drugog znaka, a da je promenljive dužine ako različitim znakovima odgovaraju kodovi različitih dužina (u suprotnom kažemo da je fiksne dužine). Na primer, ako se kôd sastoji od dva kodirana simbola (A i B) i ako je kod karaktera A jednak 11, a kod karaktera B jednak 1110, u pitanju je kod promenljive dužine koji nije prefiksni. Jedan od primera prefiksnih kodova promenljive dužine je Hafmanov kôd.

1. Dat je tekst u kome se 13 puta pojavljuje slovo A, 6 puta slovo B, 3 puta slovo C, 2 puta slovo D, 4 puta slovo E i 4 puta slovo F. Odrediti Hafmanove kodove za date karaktere.

Rešenje: Najpre se pronađu dva karaktera koja se najređe javljaju (u prvom koraku su to karakteri C i D) i zatim se konstruiše binarno drvo sa korenom čija je vrednost jednaka zbiru pojavljivanja C i D u tekstu, a čvorovi C i D su potomoci, pri čemu se pamte njihova pojavljivanja. Ukoliko postoji više karaktera koji se podjednako javljaju, dovoljno je izabrati bilo koje. Zatim se novokreirani čvor posmatra kao novi karakter sa frekvencijom 5 zajedno sa preostalim karakterima (A, B, E i F) i, u zavisnosti od toga koja dva karaktera imaju najmanju frekvenciju, formira se novo drvo koje se može (a i ne mora u opštem slučaju) nadovezivati na početno. Postupak se ponavlja sve dok svaki od karaktera u tekstu ne bude procesuiran. Konačno se dobija binarno drvo čiji su listovi polazni karakteri iz teksta, a vrednost korena je jednaka zbiru pojavljivanja svakog od karaktera. Drvo sada izgleda ovako:



Na kraju se na svaku levu granu dopiše cifra 0, a na svaku desnu cifra 1. Konačno se za svaki karakter iščita niz cifri na putu od korena do njega samog, što predstavlja i njegov Hafmanov kôd. Tako su kodovi sa slike: $A \to 1$, $B \to 011$, $C \to 0101$, $D \to 0100$, $E \to 001$ i $F \to 000$. (Primetimo da ukupna memorija koju zauzima ovaj tekst ukoliko je predstavljan Hafmanovim kodom iznosi 75 bita, a ako bismo za kodiranje karaktera koristili binarne brojeve od 000 do 101, potrošili bismo 96 bitova, što je značajna ušteda.)

6 Binarno kodirani dekadni brojevi

Postoji više zapisa pomoću kojih se može svaka dekadna cifra kodirati nekim binarnim brojem. Kako je broj dekadnih cifara 10, za odgovarajuće kodiranje se koriste 4 binarne cifre. Kod zapisa 8421 se svaka dekadna cifra pretvori u binarni broj, a zatim se zapis eventualno dopuni vodećim nulama. Kod zapisa višak 3 se na dekadnu cifru najpre doda vrednost 3, a zatim se ponovi prethodni postupak. Odgovarajući zapisi za sve dekadne cifre u kodu 8421 i višak 3 su prikazane u sledećoj tabeli:

Cifra	0	1	2	3	4	5	6	7	8	9
8421	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001
Višak 3	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100

Sabiranje se u zapisu 8421 vrši tako što se najpre dva broja zapišu u datom kodu, nakon čega se vrši sabiranje cifru po cifru u binarnom sistemu i pamte se prenosi za svaku od 4 grupe binarnih cifara (koje se odnose na vrednost date dekadne cifre). U drugoj fazi se vrši korekcija rezultata sabiranje svake četvorke. Ako je data četvorka veća ili jednaka binarnoj vrednosti 1010 ili je prenos na narednu četvorku jednak 1, vrši se dodavanje broja 0110. Inače se dodaje 0000 (nula). Pritom se pamti poslednji prenos. Ukoliko je poslednji prenos prve faze ili poslednji prenos druge faze jednak 1, došlo je do prekoračenja. Rezultat je broj koji se dobije nakon druge faze sabiranja. Oduzimanje se vrši analogno sabiranju, jedina razlika je što se u svakoj fazi umesto sabiranja vrši oduzimanje dva binarna broja, pri čemu se pamte odgovarajuće pozajmice. U drugoj fazi se vrši korekcija, tj. oduzima vrednost 0110, samo ukoliko je pozajmica iz naredne četvorke jednaka 1, a inače se oduzima vrednost 0000.

1. Izvršiti sledeće operacije u zapisu 8421 ako su brojevi zapisani na 5 mesta: (a) $-23492 + (-5189)$, (b) $2634 - 52629$.

Rešenje: Sabiranja i oduzimanja je uvek potrebno svoditi na sabiranje dva pozitivna broja ili oduzimanje manjeg broja od većeg. Konkretno, u primeru pod (a) je $-23492 + (-5189) = -(23492 + 5189)$, a u primeru pod (b) $2634 - 52629 = -(52629 - 2634)$. Sada se odgovarajuće sabiranje (odnosno oduzimanje) vrši na sledeći način (u levom delu slike je prikazano sabiranje iz (a), a u desnom oduzimanje iz (b)):

	0010	0011	0100	1001	0010
	0000	0101	0001	1000	1001
0	0	0	1	0	0
	0010	1000	0110	0001	1011
0					
	0000	0000	0000	0110	0110
	0010	1000	0110	1000	0001
	2	8	6	8	1

	0101	0010	0110	0010	1001
	0000	0010	0110	0011	0100
0	1	1	1	0	0
	0100	1111	1111	1111	0101
	0000	0110	0110	0110	0000
	0100	1001	1001	1001	0101
	4	9	9	9	5

Dakle, rešenja su: (a) -28681 i (b) -49995. Kako su poslednji prenosi prve i druge faze jednaki 0, sledi da u primeru pod (a) ne dolazi do prekoračenja.

Sabiranje se u zapisu višak 3 vrši tako što se najpre dva broja zapišu u datom kodu, nakon čega se vrši sabiranje cifru po cifru u binarnom sistemu i pamte se prenosi za svaku od 4 grupe binarnih cifara (koje se odnose na vrednost date dekadne cifre). U drugoj fazi se vrši korekcija rezultata sabiranje svake četvorke. Ako je prenos na narednu četvorku jednak 1, vrši se dodavanje vrednosti 0011, a inače (ukoliko je naredni prenos jednak 0) se dodaje 1101. Prekoračenje se kod sabiranja javlja ukoliko je poslednji prenos u prvoj fazi sabiranja jednak 1. U drugoj fazi sabiranja se ne vrši prenos na narednu četvorku. Oduzimanje se vrši tako što se umanjilac komplementira u osnovi 10, a zatim se svodi na sabiranje. Pritom se prekoračenje ne javlja.

2. Izvršiti sledeće operacije u zapisu višak 3 ako su brojevi zapisani na 5 (pod (a)) odnosno 4 (pod (b)) mesta: (a) $28367 + 2847$, (b) $8586 - 5836$.

Rešenje: Rešenje je prikazano na donjoj slici. U levom deku slike je prikazano sabiranje, a u desnom oduzimanje. Do prekoračenja kod sabiranja ne dolazi, budući da je poslednji prenos jednak 1. Kod oduzimanja, najpre se broj -5836 napiše u potpunom komplementu u osnovi 10 – 94164. Sada se vrši sabiranje brojeva 94164 i 08586, koji su u potpunom komplementu. Početne cifre 9 i 0 se odnose na znak broja, ali se pri algoritmu posmatraju ravnopravno. Nakon izvršene operacije oduzimanja se broj iz potpunog komplementa prebacuje u dekadnu vrednost i iznosi 2750.

	0101	1011	0110	1001	1010
	0011	0101	1011	0111	1010
0	1	1	1	1	0
	1001	0001	0010	0001	0100
	1101	0011	0011	0011	0011
	0110	0100	0101	0100	0111
	3	1	2	1	4

	1100	0111	0100	1001	0111
	0011	1011	1000	1011	1001
1	1	0	1	1	0
	0000	0010	1101	0101	0000
	0011	0011	1101	0011	0011
	0011	0101	1010	1000	0011
	0	2	7	5	0

7 Otkrivanje i korekcija grešaka

Pri prenosu podataka često dolazi do promene pojedinih bitova podataka zbog smetnji na prenosnom putu i različitih tipova šumova na lokacijama odašiljanja i prijema. Smetnje se dešavaju bez obzira na udaljenost uređaja, tj. kako pri prenosu podataka između dva računara tako i između komponenti istog računara. Postoje dva osnovna pristupa rešavanju ovog problema – kontrola grešaka unatrag i kontrola grešaka unapred. Kod kontrole grešaka unatrag uz podatke se šalju dodatne informacije koje služe da se ustanovi da li postoje greške, ali ne i da se one otklone. Neispravno preneseni podaci se ponovo šalju. Kod kontrole grešaka unapred se uz podatke šalju dodatne informacije koje služe kako da se ustanovi da greške postoje, tako i da se odredi njihova lokacija. Neispravno preneseni podaci se automatski koriguju. Najčešći korišćen metod koji vrši kontrolu grešaka unatrag je ciklična provera redundanci (CRC), a za kontrolu grešaka unapred Hamingovog SEC kodovi.

1. Koristeći polinom generator $G(x) = x^4 + x^3 + 1$ odrediti oblik za slanje poruke 11100110.

Rešenje: Najpre je potrebno izračunati binarni ekvivalent za odgovarajući polinom generator. Binarni ekvivalent se dobija tako što se redom, jedna do druge, dopišu koeficijenti uz odgovarajuće stepene promenljive u polinomu (ti koeficijenti su u polinomu generatoru jedino 0 ili 1). Kako je $G(x) = x^4 + x^3 + 1 = 1 \cdot x^4 + 1 \cdot x^3 + 0 \cdot x^2 + 0 \cdot x^1 + 1 \cdot x^0$, sledi da je odgovarajući binarni ekvivalent 11001. Na poruku koja se šalje je potrebno dodati onoliko nula koliki je stepen polinoma generatora (u ovom slučaju 4), a zatim izvršiti deljenje tog broja i binarnog ekvivalenta. U svakom koraku se vrši oduzimanje, dok se ne dobije ostatak koji je manji od binarnog ekvivalenta. Oduzimanje se vrši bez pozajmica, drugim rečima između svake dve cifre pri oduzimanju se novodobijena cifra formira eskluzivnom disjunkcijom. Vodeće nule se ignorišu. Dakle, deljenje se vrši na sledeći način:

```
111001100000 : 11001
11001
  10111
  11001
   11100
   11001
     10100
     11001
      11010
      11001
         110
```

Poruka koja se šalje je poruka iz zadatka na koju je nadovezan ostatak. Ostatak je potrebno dopuniti vodećim nulama, tako da bude zapisan na onoliko mesta koliki je stepen polinoma. Dakle, ostatak postaje 0110, a poruka koja se šalje je 111001100110.

2. Utvrditi da li je poruka 1100101101 uspešno primljena i, ukoliko jeste, odrediti njen polazni oblik. Korišćen je polinom generator $G(x) = x^2 + 1$.

Rešenje: Najpre je binarni ekvivalent polinoma generatora jednak 101. Potrebno je izvršiti deljenje primljene poruke i binarnog ekvivalenta polinoma generatora i ustanoviti da li je ostatak jednak ili različit od nule. Ako je jednak nuli, poruka je uspešno primljena, a ako je različit, nije. Postupak deljenja se, analogno deljenju u prethodnom zadatku, odvija na sledeći način:

```
1100101101 : 101
101
 110
 101
  111
  101
   100
   101
     111
     101
      100
      101
        11
```

U ovom slučaju je ostatak 11, pa poruka nije uspešno primljena. Da poruka jeste uspešno primljena, bilo bi potrebno odrediti njen polazni oblik. U opštem slučaju, ako je stepen polinoma generatora n, polazni oblik poruke se dobija odbacivanjem njenih poslednjih n bitova.

3. Koristeći Hamingove SEC kodove, izvršiti korekciju greške u poruci 101001100110.

Rešenje: Najpre je potrebno prvih osam bitova poruke označiti sa $m_8, m_7, \ldots, m_1$ (koji se odnose na sadržaj), a preostala četiri sa c_4, c_3, c_2, c_1 (koji predstavljaju kontrolne bitove):

m_8	m_7	m_6	m_5	m_4	m_3	m_2	m_1	c_4	c_3	c_2	c_1
1	0	1	0	0	1	1	0	0	1	1	0

Nakon toga se formira tablica Hamingovih kodova. U prvoj koloni su dekadni brojevi od 12 do 1, u drugoj koloni odgovarajući binarni brojevi zapisani pomoću 4 cifre, a u trećoj se svakom broju (od 12 do 1) dodeljuje neki od bitova $m_8, m_7, \ldots, m_1, c_4, c_3, c_2, c_1$. Bitovi c_i se dodeljuju onim brojevima koji su stepeni dvojke (brojevima 1, 2, 4 i 8), a na preostala mesta se popune bitovi $m_8, m_7, \ldots, m_1$ na način prikazan u sledećoj tabeli:

12	1100	m_8
11	1011	m_7
10	1010	m_6
9	1001	m_5
8	1000	c_4
7	0111	m_4
6	0110	m_3
5	0101	m_2
4	0100	c_3
3	0011	m_1
2	0010	c_2
1	0001	c_1

Zatim se računaju kontrolni bitovi c_4', c_3', c_2', c_1' preko $m_8, m_7, \ldots, m_1$ tako što se redom u četvrtoj, trećoj, drugoj i prvoj koloni svih binarnih zapisa iz tabele u odgovarajuću formulu ubace oni brojevi m_i čija je vrednost 1 i izvrši se njihova ekskluzivna disjunkcija (vrednosti za m_i se menjaju iz poruke u zadatku). Drugim rečima, važi:

$$\begin{aligned}
c_4' &= m_5 \oplus m_6 \oplus m_7 \oplus m_8 = 0 \oplus 1 \oplus 0 \oplus 1 = 0\\
c_3' &= m_2 \oplus m_3 \oplus m_4 \oplus m_8 = 1 \oplus 1 \oplus 0 \oplus 1 = 1\\
c_2' &= m_1 \oplus m_3 \oplus m_4 \oplus m_6 \oplus m_7 = 0 \oplus 1 \oplus 0 \oplus 1 \oplus 0 = 0\\
c_1' &= m_1 \oplus m_2 \oplus m_4 \oplus m_5 \oplus m_7 = 0 \oplus 1 \oplus 0 \oplus 0 \oplus 0 = 1
\end{aligned}$$

Konačno, izračunava se vrednost $c_4c_3c_2c_1 \oplus c_4'c_3'c_2'c_1' = 0110 \oplus 0101 = 0011$. Binarnom broju 0011 odgovara dekadna vrednost 3. U tabeli, u redu gde je broj 3 se nalazi bit m_1. To znači da je došlo do grešeke baš na tom bitu i da je tu vrednost bita potrebno komplementirati. Zbog toga je ispravna poruka 10100111. Ako je u pravcu odgovarajućeg broja u tabeli neka vrednost c_i ili ako je rezultat poslednje ekskluzivne disjunkcije broj koji je veći od 12 ili manji od 1, ne dolazi do greške.

8 DPD kodiranje

DPD kodiranje je jedan od načina zapisa brojeva u IEEE 754 zapisu sa dekadnom osnovom. Pri kodiranju i dekodiranju, koje će detaljnije biti objašnjeno u zadacima, koriste se sledeće dve tabele (leva je za kodiranje, a desna za dekodiranje):

aei	pqr stu v wxy
000	bcd fgh 0 jkm
010	bcd jkh 1 01m
100	jkd fgh 1 10m
001	bcd fgh 1 00m
110	jkd 00h 1 11m
101	fgd 01h 1 11m
011	bcd 10h 1 11m
111	00d 11h 1 11m

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

1. Zapisati sledeće brojeve u pokretnom zarezu pomoću dekadne osnove koristeći DPD kodiranje: (a) 123.4, (b) $-982.5294 \cdot 10^{42}$.

Rešenje: (a) Najpre je potrebno broj 123.4 napisati u obliku $x \cdot 10^y$, pri čemu je x ceo broj zapisan na 7 mesta (sa eventualnim vodećim nulama), a y odgovarajući stepen. Ukoliko frakcija broja sadrži više od 7 cifara, potrebno je izvršiti zaokruživanje. Dakle, $123.4 = 0001234 \cdot 10^{-1}$. Frakciju 0001234 je potrebno podeliti na tri grupe, od kojih prva ima jednu, a preostale dve po tri cifre: 0, 001 i 234. Sada je potrebno tri cifre 001 pretvoriti u 10 bitova koristeći tabelu za kodiranje. Najpre prevedimo svaku od cifara 0, 0 i 1 u 8421 zapis i dobijenih 12 cifara označimo redom sa $abcd$, $efgh$ i $ijkm$. Dobija se $0 \rightarrow abcd = 0000$, $0 \rightarrow efgh = 0000$ i $1 \rightarrow ijkm = 0001$. Sada je potrebno u zavisnosti od cifara $abcdefghijkm$ generisati novih 10 bitova $pqrstuvwxy$. Prema bitovima aei je potrebno pozicionirati se na odgovarajuće mesto u tabeli za kodiranje. Kako je $aei = 000$, potrebno je posmatrati prvi red tabele. Tu važi da je $pqr = bcd$, $stu = fgh$, $v = 0$ i $wxy = jkm$, pa se zamenom dobija $pqrstuvwxy = 0000000001$. Na sličan način se prevode i poslednje tri cifre frakcije 234. Važi da je $2 \rightarrow abcd = 0010$, $3 \rightarrow efgh = 0011$ i $4 \rightarrow ijkm = 0100$. Kako je i ovde $aei = 000$, na osnovu prvog reda tabele za kodiranje se dobija $pqrstuvwxy = 0100110100$. Eksponent -1 je potrebno zapisati u binarnom sistemu na 8 mesta sa uvećanjem 101. Dakle, zapis eksponenta je oblika $-1 + 101 = 100 = (01100100)_2$. Prva cifra frakcije je 0 i ona se kodira u obliku 000. U opštem slučaju, ukoliko je prva cifra frakcije 0 – 7, ona se kodira sa odgovarajuća tri bita (binarna vrednost za datu dekadnu cifru), a ukoliko je 8 ili 9, kodira se kao 0 za 8 i 1 za 9. Kombinacija je deo od 5 bitova koje se sastoji od početka eksponenta (prva dva bita) i početka frakcije. U ovom slučaju kombinacija je oblika 01000 (prva dva bita 01 su početni bitovi eksponenta, a preostala tri 000 su bitovi cifre frakcije). Za njom u zapisu sledi nastavak eksponenta od 6 cifara, a za njim preostalih 20 cifara frakcije. Na početku koda je bit za znak – 0 ako je broj pozitivan, a 1 ako je negativan. Dakle zapis (od 32 bita) broja 123.4 koristeći DPD kodiranje je oblika 0 01000 100100 0000000001 0100110100. (b) Zadatak se radi analogno slučaju pod (a). Najpre je $-982.5294 \cdot 10^{42} = -9|825|294 \cdot 10^{38}$. Za zapis dela frakcije 825, koristeći činjenicu da je $8 \rightarrow abcd = 1000$, $2 \rightarrow efgh = 0010$ i $5 \rightarrow ijkm = 0101$ i tabelu za kodiranje, dobija se $pqrstuvwxy = 1000101101$, a za zapis 294, uzimajući u obzir da je $2 \rightarrow abcd = 0010$, $9 \rightarrow efgh = 1001$ i $4 \rightarrow ijkm = 0100$, sledi $pqrstuvwxy = 0101011010$. Zapis eksponenta sa uvećanjem 101 u binarnom sistemu na 8 mesta je $38 + 101 = 139 = (10001011)_2$. Prva cifra frakcije je 9, pa se ona kodira sa 1. U ovom slučaju, budući da kombinacija ima 5 cifara, a u ovom slučaju početak eksponenta 10 i početak frakcije 1 zauzimaju ukupno tri cifre, na njen početak se dodaju dve jedinice, 11, pa je ona oblika 11101. Uzimajući i još u obzir da je broj negativan, njegov zapis je 1 11011 001011 1000101101 0101011010.

2. Odrediti dekadni broj koji je predstavljen u pokretnom zarezu zapisanim u IEEE 754 zapisu pomoću dekadne osnove (DPD kodiranje): 0 01111 011110 0000000000 0000110101.

Rešenje: Najpre je potrebno uočiti da li kombinacija počinje sa 11 ili nekim drugim parom cifara. U prvom slučaju bi trebalo te dve cifre ignorisati i uzeti u obzir da su 3. i 4. cifra kombinacije prve dve cifre eksponenta, a 5. prva cifra frakcije. U ovom slučaju su prve dve cifre kombinacije zapravo prve dve cifre eksponenta, a preostale tri prva cifra frakcije. Dakle, eksponent je (budući da se on uvek zapisuje sa uvećanjem 101) oblika $(01011110)_2 - 101 = 94 - 101 = -7$. Preostale tri cifre kombinacije 111 određuju prvu cifru frakcije 7. Prvih 10 cifara nastavka frakcije $pqrstuvwxy = 0000000000$ kodira naredne tri cifre frakcije dekadnog broja. Kako je u ovom slučaju $v = 0$, pozicioniranjem u prvi red tabele za dekodiranje može se zaključiti da je $abcd = 0pqr$, $efgh = 0stu$ i $ijkm = 0wxy$, odnosno $abcd = (0000)_2 = 0$, $efgh = (0000)_2 = 0$ i $ijkm = (0000)_2 = 0$. Slično, na osnovu preostalg koda frakcije 0000110101, pozicioniranjem na odgovarajući red u tabeli za dekodiranje se dobija $abcd = (0000)_2 = 0$, $efgh = (0011)_2 = 3$ i $ijkm = (0101)_2 = 5$. Kako je još bit za znak 0, broj je pozitivan, pa je on oblika $7000035 \cdot 10^{-7} = 0.7000035$.

Specijalne vrednosti koje se mogu pojaviti u broju su 0, ∞ i NaN vrednost. Za nulu važi da su joj sve cifre frakcije 0, a eksponent može biti proizvoljan broj. Dakle, kod zapisa nule poslednje tri cifre kombinacije su 000 (što predstavlja prvu cifru frakcije), a frakcija se sastoji od 20 nulâ (što predstavlja poslednjih 6 cifara frakcije), a ostali bitovi mogu biti proizvoljni, s tim što, ukoliko je bit za znak 0 tada je u pitanju pozitivna nula, a ukoliko je bit za znak 1 u pitanju je negativna nula. Za ∞ važi da je kombinacija oblika 11110, a u zavisnosti bita za znak, radi se o pozitivnoj ili negativnoj beskonačnosti. Pri aritmetičkim operacijama važi npr. da je $\infty + \infty = \infty$, $5/0 = \infty$, $-5/0 = -\infty$, itd. NaN vrednosti su specijalne vrednosti koje se javljaju u nekim izuzetnim situacijama. Postoji signalni NaN (sNan) i tihi NaN (qNaN). Prvi se javlja pri operacijama kao što su nedozvoljena konverzija i sl., a drugi pri aritmetičkim operacijama, kao npr. $\infty - \infty$, $0/0$, $0 \cdot \infty$, sNaN + 3, itd. Ako je argument neke operacije sNaN, rezulat je qNaN. I sNaN i qNaN u kombinaciji sadrže sve jedinice (11111), dok prva cifra nastavka eksponenta za sNaN počinje jedinicom, a za qNaN nulom.

9 IEEE 754 zapis sa binarnom osnovom

Broj se u IEEE 754 standardu zapisuje u obliku $x \cdot 2^y$, gde se x naziva frakcija, a y eksponent. Broj 2 je osnova. Pomoću ovog standarda se mogu zapisati realni brojevi (sa odgovarajućom preciznošću) i neke specijalne vrednosti među kojima su 0, ∞, NaN vrednosti, itd. Najčešći su zapisi u jednostrukoj i dvostrukoj. U jednostrukoj tačnosti broj se zapisuje pomoću 32 bita, od kojih 1 bit zauzima znak, 8 bitova eksponent, a preostala 23 bita frakcija. U dvostrukoj tačnosti 1 bit zauzima znak, 11 bitova eksponent, a 52 frakcija (ukupno 64 bita).

1. Zapisati sledeće brojeve prema IEEE 754 standardu u jednostrukoj tačnosti: (a) 13.25, (b) −111.625.

Rešenje: (a) Najpre se broj 13.25 prevodi u binarnu osnovu: $13.25 = (1101.01)_2$. Dobijeni rezultat u binarnoj osnovi je potrebno zapisati u obliku $1.f \cdot 2^e$, gde su f cifre frakcije, a e odgovarajući eksponent, pa dati zapis postaje $(1.10101)_2 \cdot 2^3$. Eksponent se u IEEE 754 standardu piše sa uvećanjem 127 u binarnom sistemu na 8 mesta i u ovom primeru iznosi $3 + 127 = 130 = (10000010)_2$. Prvih pet cifara frakcije su one cifre koje se u broju 1.10101 nalaze iza decimalne tačke (zapis je 10101), a preostalih 18 cifara treba dopuniti nulama. Broj je pozitivan, pa je bit za znak 0. Dakle, zapis broja je 0 10000010 10101000000000000000000. (b) Kako je $-111.625 = -(1101111.101)_2 = -(1.101111101) \cdot 2^6$, zapis eksponenta je $6 + 127 = 133 = (10000101)_2$. Broj je negativan, a cifre frakcije koje se zapisuju su iza decimalne tačke, pa je zapis 1

10000101 10111110100000000000000.

2. Pročitati zapis 1 10000110 01001000000000000000000 u IEEE 754 standardu u jednostrukoj tačnosti.

Rešenje: Kako je bit za znak 1, broj je negativan. Eksponent je dat sa uvećanjem 127 i iznosi $(10000110)_2 - 127 = 134 - 127 = 7$. Jednostavno se uočava i da je frakcija broja 1.01001. Dakle, dekadna vrednost broja je $(-1.01001)_2 \cdot 2^7 = (-10100100)_2 = -164$.

U prethodna dva zadatka prikazani su primeri zapisivanja tzv. normalizovanih brojeva. Postoje i denormalizovani brojevi; to su oni brojevi koji su po apsolutnoj vrednosti veoma bliski nuli. Kod njih se eksponent zapisuje pomoću 8 nulâ, a ako je zapis frakcije f, ona se iščitava u obliku $0.f$. Eksponent uvek iznosi -126. Na primer, na osnovu zapisa 0 00000000 00010000000000000000000, zaključujemo da se radi o denormalizovanom broju. Na osnovu početka zapisa frakcije 0001 (ostatak su nule, pa nisu od značaja), zaključujemo da je frakcija oblika 0.0001. Broj je pozitivan. Vrednost datog denormalizovanog broja je $(0.0001)_2 \cdot 2^{-126} = 2^{-4} \cdot 2^{-126} = 2^{-130}$.

Pored normalizovanih i denormalizovanih brojeva, IEEE 754 standard u jednostrukoj tačnosti podržava zapis za nulu, beskonačno i NaN vrednosti. Nula se piše pomoću svih nula u zapisu frakcije i eksponenta, s tim što bit za znak može uzeti vrednost 0 ili 1, pa se tada radi o pozitivnoj ili negativnoj nuli. Beskonačno se piše za svim jedinicama u eksponentu i svim nulama u frakciji. I ovde, u zavisnosti od bita za znak broja se može govoriti o $+\infty$, odnosno $-\infty$. Kod zapisa sa sNaN i qNaN eksponent uzima vrednost 11111111. Prvi bit frakcije za sNaN je 0, a ostatak frakcije je različit od nule (nisu sve preostale cifre jednake nuli). Prvi bit frakcije za qNaN je 1.

3. Zapisati sledeće brojeve prema IEEE 754 standardu u dvostrukoj tačnosti: (a) 48.125, (b) -1780.53125.

Rešenje: (a) Broj 48.125 je potrebno napisati u obliku $x \cdot 2^y$, gde je x binarna frakcija oblika $1.f$, a y odgovarajući eksponent. Tako je $48.125 = (110000.001)_2 = (1.10000001)_2 \cdot 2^5$. Eksponent se piše sa uvećanjem 1023 u binarnom sistemu na 8 mesta i iznosi $5 + 1023 = 1028 = (10000000100)_2$. Frakcije počinje sa 10000001 (bitovi iza decimalne tačke), a ostatak od 44 mesta (ukupno ima 52 cifre frakcije) je potrebno dopuniti nulama. Kako je još broj pozitivan, bit za znak je 0, pa je zapis 0 10000000100 10000001 $\underbrace{0...0}_{\text{44 nule}}$. (b) Kako je $-1780.53125 = -(11011110100.10001)_2 = -(1.101111010010001)_2 \cdot 2^{10}$, zapis eksponenta je $10 + 1023 = 1033 = (10000001001)_2$. Broj je još i negativan, pa je traženi zapis 1 10000001001 101111010010001 $\underbrace{0...0}_{\text{37 nulâ}}$.

4. Pročitati zapis 1 10000011001 1011 $\underbrace{0...0}_{\text{48 nulâ}}$ u IEEE 754 standardu u dvostrukoj tačnosti.

Rešenje: Bit za znak je 1, pa je broj negativan. Eksponent je dat sa uvećanjem 1023, pa iznosi $(10000011001)_2 - 1023 = 1049 - 1023 = 26$. Binarna vrednost frakcije je 1.1011 (ostale nule nisu od značaja). Dakle, traženi broj je $-(1.1011)2 \cdot 2^26 = -(11011)_2 \cdot 2^22 = -27 \cdot 2^22$.

I u IEEE 754 zapisu sa dvostrukom tačnošću postoje denormalizovani brojevi. Kod njih se eksponent zapisuje pomoću 11 nulâ, a ako je zapis frakcije f, ona se iščitava u obliku $0.f$. Eksponent uvek iznosi -1022. Takođe, podržan je zapis za nulu, beskonačno i NaN vrednosti. ula se piše pomoću svih nula u zapisu frakcije i eksponenta, s tim što bit za znak može uzeti vrednost 0 ili 1, pa se tada radi o pozitivnoj ili negativnoj nuli. Beskonačno se piše za svim jedinicama u eksponentu i svim nulama u frakciji. I ovde, u zavisnosti od bita za znak broja se može govoriti o $+\infty$, odnosno $-\infty$. Kod zapisa sa sNaN i qNaN eksponent uzima vrednost 11111111111. Prvi bit frakcije za sNaN je 0, a ostatak frakcije je različit od nule (nisu sve preostale cifre jednake nuli). Prvi bit frakcije za qNaN je 1.

10 Minimizacija logičkih funkcija

1. Izvršiti minimizaciju sledeće funkcije metodom algebarskih transformacija:

$$f(a,b,c,d) = ab'd' + ab'c + ab'd' + cd + a'b'c + c'd + abb'cd.$$

Rešenje: Funkcija f se minimizuje koristeći komutativnost i asocijativnost and i or funkcija, kao i distributivnost množenja (and funkcije) u odnosu na sabiranje (or funkcija). Dodatno, često se koriste i činjenice da je $a + a' = 1$ i $aa' = 0$. Funkcija f se minimizuje na sledeći način:

$$f(a,b,c,d) = ad'(b' + b) + b'c(a + a') + d(c' + c) + acd(bb') = ad' + b'c + d.$$

2. Funkcija $f = f(A, B, C, D)$ uzima vrednost 1 ako je dekadna vrednost broja $ABCD$ jednaka 0, 2, 3 ili 5–9, vrednost 0 ako je $ABCD$ jednako 1 ili 4, a inače je nedefinisana. Izvršiti njenu minimizaciju metodom Karnoovih mapa.

Rešenje: Odgovarajuće vrednosti za funkciju f ćemo napisati u matricu kao na donjoj. Pritom treba voditi računa da se u prvom redu, odnosno prvoj koloni svi susedi (pri čemu su susedi i prvi i poslednji par) kombinacije 00, 01, 11 i 10 za AB, odnosno BC razlikuju za tačno jedan bit. Vrednosti za koje funkcija f nije definisan su označene sa n. Zatim se vrši ucrtavanje što manjeg broja pravougaonika tako da svaki od njih ima što veću površinu i dužinu stranice oblika 2^n. Njima sve jedinice moraju biti obuhvaćene, a nijedan ne sme sadržati nule. Pritom su susedni elementi u matrici sa slike susedni, ali su susedni i oni iz prve i poslednje kolone, odnosno reda. Svaki od ovih pravougaonika će odgovarati jednom sabirku tražene formule. Na primer, za pravougaonik koji zahvata polja $(3, 4) \times (1, 2, 3, 4)$, odgovarajuća formula je C, budući da je jedino bit C ostao nepromenjen pri kretanju kroz pravougaonik. Ukoliko bi tada C imao vrednost 0, formula bi sadržala C' a ne C, a ukoliko bi npr. i element B bio nepromenjen, umesto C sabirak bi bio oblika BC. Tako su odgovarajuće formule za preostala tri pravougaonika oblika A, BD i $B'D'$. Zato je traženi oblik funkcije f dat sa $f = A+C+BD+B'D'$.

AB \ CD	00	01	11	10
00	1	0	n	1
01	0	1	n	1
10	1	1	n	n
10	1	1	n	n

11 Kombinatorna i sekvencijalna logička kola

1. Polusabirač je logicko kolo koje na ulazu dobija dve binarne cifre, A i B, a kao rezultat vraća njihovu sumu S i eventualni prenos P. Nacrtati odgovarajuće logičko kolo za polusabirač.

Rešenje: Konstruišimo najpre tabelu čije su kolone dva argumenta, A i B, njihova suma S i prenos do koga može doći, u oznaci P. Iz tabele se jasno vidi da se suma elemenata A i B bez prenosa može shvatiti i kao njihova ekskluzivna disjunkcija. Dakle, $S = A \oplus B$. Sa druge strane, jasno je i da je $P = AB$.

A	B	S	P
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

2. Sabirač je logičko kolo koje na ulazu dobija tri binarne cifre, A, B i C, a kao rezultat vraća njihovu sumu S i eventualni prenos P. Nacrtati odgovarajuće logičko kolo za sabirač.

Rešenje: Analogno prethodnom, najpre konstruišimo odgovarajuću tabelu sa kolonama A, B, C, S i P (tabela 3). Iz tabele se vidi da S uzima vrednost 1 ako i samo ako trojka ABC sadrži neparan broj jedinica. Drugim rečima, $S = A \oplus B \oplus C$. Da bismo pronašli formulu za P, posmatrajmo slučajeve kada je $P = 1$, tj. kada ima prenosa. Ukoliko je $A = B = 1$, kakvo god da je C, doći će do prenosa. Ako to nije slučaj, do prenosa će tada doći jedino ako tačno jedan od bitova A i B uzima vrednost 1, a da pritom C takođe uzima vrednost 1. Kako to znači i da su A i B različiti, u ovom slučaju ćemo nad njima primeniti ekskluzivnu disjunkciju. Dakle, $P = AB + (A \oplus B)C$.

A	B	C	S	P
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

3. Nacrtati logičko kolo za 2–4 dekoder.

Rešenje: Dekoder je logičko kolo koje se sastoji od n ulaza i 2^n izlaza. U svakom trenutku, u zavisnosti od ulaza, aktivan je tačno jedan izlaz. U našem slučaju, gde je $n = 2$, se na osnovu binarnog broja $s_1 s_0$ dobjenog spajanjem ulaza s_1 i s_0 aktivira jedan od četiri izlaza (prvi izlaz označava broj $0 = 00$, drugi $1 = 01$, treći $2 = 10$ i četvrti $3 = 11$). Ako sa d_0, d_1, d_2 i d_3 označimo odgovarajuće izlaze, biće $d_0 = s_1' s_0'$, $d_1 = s_1' s_0$, $d_2 = s_1 s_0'$ i $d_3 = s_1 s_0$.

4. Nacrtati logičko kolo za 4–1 multiplekser.

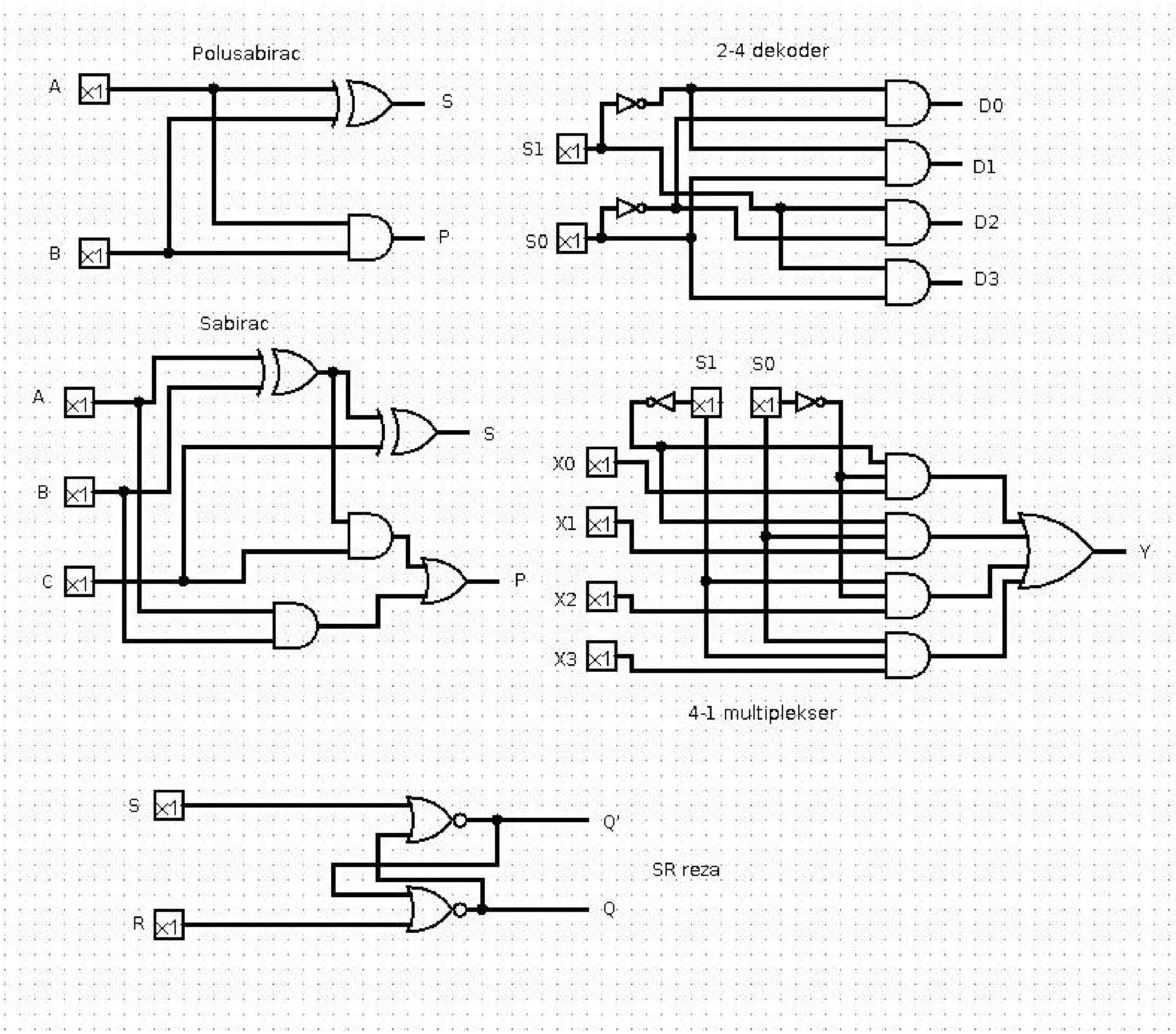
Rešenje: Multiplekser je logičko kolo koje ima 2^n ulaza, n selektorskih ulaza i jedan izlaz. Ako u binarnom zapisu selektorski ulazi predstavljaju binaran broj i, na izlazu se nalazi vrednost i-tog ulaza. Naš multiplekser ima četiri ulaza, dva selektorska ulaza i jedan izlaz. Neka su x_0, x_1, x_2, x_3 ulazi i s_0, s_1 selektorski ulazi. Ako je $s_1 s_0 = 00$ (odnosno 01, 10 i 11), vrednost izlaza je jednaka nultom (odnosno prvom, drugom i trećem) ulazu, pa je formula multipleksera $x_0 s_1' s_0' + x_1 s_1' s_0 + x_2 s_1 s_0' + x_3 s_1 s_0$.

5. Nacrtati SR rezu.

Rešenje: Sva prethodna kola su bila kombinatorna. Sekvencijalna mreža u opštem slučaju ne zavise jedino od trenutnog ulaza kao kombinatorna, već i od prethodnog stanja kola. SR reza ima dva ulaza, S i R i dva izlaza, Q i Q'. Za $S = 0$ i $R = 1$ izlaz će biti $Q = 0$ i $Q' = 1$, a za $S = 1$ i $R = 0$ izlaz će biti $Q = 1$ i $Q' = 0$. Stanje $S = 0$, $R = 0$ čuva izlaz onakav kakav je bio, dok se stanje $R = 1$, $S = 1$ smatra nedefinisanim.

Na donjoj slici se nalaze crteži za sledeća logička kola:

- polusabirač sa ulazima A i B i izlazima, sumom S i prenosom P;
- sabirač sa ulazima A, B i C i izlazima, sumom S i prenosom P;
- 2-4 dekoder sa dva ulaza i četiri izlaza;
- 4-1 multiplekser sa četiri ulaza, dva selektorska ulaza i jednim izlazom;
- SR reza sa ulazima S i R i izlatima Q' i Q'.


Polusabirac
A
B
S
P
2-4 dekoder
S1
S0
D0
D1
D2
D3
Sabirac
A
B
C
S
P
S1
S0
X0
X1
X2
X3
Y
4-1 multiplekser
S
R
Q'
Q
SR reza