



ŠOLSKI CENTER NOVA GORICA

TEHNIŠKA GIMNAZIJA IN ZDRAVSTVENA ŠOLA

SEMINARSKA NALOGA PRI PREDMETU LABORATORIJSKE VAJE

KODIRANJE ŠTEVIL

PROGRAM: Tehniška gimnazija

NOVA GORICA, april 2015

IZVLEČEK

Seminarska naloga razloži različne številske sisteme, kaj je kodiranje, pretvarjanje in povezave med različnimi številskimi sistemi. Opisana je predstavitev podatkov v računalniku, predstavljen problem zapisa negativnih števil in razložen zapis floating point – plavajoča vejica.

Ključne besede:

Številski sistemi, šestnajstiški, desetiški, osmiški, dvojiški sestav, kodiranje, kodiranje števil, pretvorbe med sistemi, predstavitev informacij v računalniku, dvojiški komplement, plavajoča vejica.

KAZALO VSEBINE

KAZALO SLIK

Slika 1: Heksadecimalno (šestnajstiško) število.....	4
Slika 2: Algoritem za pretvorbo iz desetiškega v binarni kod.....	6
Slika 3: Algoritem pretvorbe iz desetiškega sistema v šestnajstiški.....	7
Slika 4: Postopek pretvorbe iz dvojiškega v desetiški sistem - zapis polinoma.....	8

KAZALO TABEL

Tabela 1: Dvojiški in osmiški zapis.....	9
Tabela 2: Dvojiški in šestnajstiški zapis.....	10

1. UVOD

V seminarski nalogi bom razložil pojem kodiranje. Kodiranje števil bom obdelal z matematičnega, pa tudi računalniškega vidika. Predstavil bom različne številske sisteme, binarnega, osmiškega, desetiškega in šestnajstiškega. Opisal bom različne algoritme za pretvorbe med številskimi sistemi ter tesno povezanost nekaterih sistemov med sabo. Najbolj se bom posvetil dvojiškemu številskemu sistemu, saj je najpomembnejši za razumevanje delovanja računalnika oziroma procesorja (CPE), pri kateremu bom poleg osnovnega postopka pretvarjanja opisal tudi postopke dela z negativnimi in decimalnimi števili.

2. ŠTEVILSKI SISTEMI

Številski sistem ali sestav je sistem, v katerem so urejena števila. V rabi je pozicijski sistem ali sistem z mestnimi vrednostmi, v katerem je vsako število izraženo v obliki polinoma. Na primer, s 54321 je predstavljeno število:

$$5 \cdot 10^4 + 4 \cdot 10^3 + 3 \cdot 10^2 + 2 \cdot 10^1 + 1 \cdot 10^0.$$

Kot vidimo, je v zapisu zelo pomembno, na katerem mestu stoji posamezna številka, saj iste številke v drugačnem vrstnem redu predstavljajo neko drugo število. Tako, na primer, predstavlja zapis 12345 število $1 \cdot 10^4 + 2 \cdot 10^3 + 3 \cdot 10^2 + 4 \cdot 10^1 + 5 \cdot 10^0$, ki seveda ni enako številu $5 \cdot 10^4 + 4 \cdot 10^3 + 3 \cdot 10^2 + 2 \cdot 10^1 + 1 \cdot 10^0$.

Na splošno ima zapis realnega števila v desetiškem sistemu obliko

$$\dots d_3 d_2 d_1 d_0 . d_{-1} d_{-2} \dots$$

pri čemer so d_i številke iz množice $\{0, 1, 2, \dots, 9\}$. Vrednost posameznega mesta v zapisu pa določajo potence števila 10, pri čemer velikost potenc od leve proti desni pada. Pika v zapisu nam pove, kje se začnejo mesta, ki pripadajo potencom z negativnimi eksponenti. Zgornji zapis tako predstavlja število

$$\dots + d_3 \cdot 10^3 + d_2 \cdot 10^2 + d_1 \cdot 10^1 + d_0 \cdot 10^0 + d_{-1} \cdot 10^{-1} + d_{-2} \cdot 10^{-2} + \dots$$

Seveda lahko število 10 nadomestimo s poljubnim naravnim številom $b > 1$. Tako dobimo številski sistem z osnovo b . Na primer, ko je $b = 2$, vzamemo za množico števk $\{0, 1\}$ in govorimo o dvojiškem številskem sistemu; pri $b = 8$ in množici števk $\{0, 1, 2, \dots, 7\}$ pa o osmiškem številskem sistemu. Predstavimo število 100 v nekaterih od teh sistemov:

$$100_{(10)} = 64_{(16)} = 84_{(12)} = 144_{(8)} = 400_{(5)} = 1100100_{(2)}$$

V katerem sistemu je število predstavljeno smo označili tako, da smo osnovo sistema zapisali podpisano v oklepaju kot indeks. Sledi še en primer števila, zapisanega v desetiškem in šestnajstiškem sistemu:

$$63_{(10)} = 3F_{(16)}$$

Ta zgled je zanimiv, ker ima število v šestnajstiškem sestavu v svojem zapisu črko F. Ker ljudje uporabljamo desetiški sistem, imamo tudi 10 znakov za zapis števil (0 – 9). Za šestnajstiški zapis števila pa zato vpeljemo še dodatne znake in sicer iz začetka abecede.

Tako dobimo množico števk {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F}, pri čemer črka A predstavlja desetiško zapisano število 10, črka F število 15, za ostale pa lahko iz te zveze sklepate sami. Primer šestnajstiško zapisanega števila in njegov pomen:



Slika 1: Heksadecimalno (šestnajstiško) število
(Vir: <http://goo.gl/cZcEBR>)

Če nas torej zanima desetiška vrednost tega števila, tole naredimo tako:

$$14_{(16)} = 4 \cdot 16^0 + 1 \cdot 16^1 = 4 + 16 = 20_{(10)}$$

Več o različnih pretvorbah pa v naslednjih poglavjih.

3. KODIRANJE

Kodiranje lahko imenujemo vsako pretvorbo podatkov iz ene oblike v drugo. Najpogosteje kodiramo v neko zaporedje simbolov. Cilji kodiranja:

- shranjevanje, prenašanje in obdelava podatkov
- razumljivost predstavitve (cifre za števila, črke za besede, prometni znaki...)
- varovanje, šifriranje vsebine

V naravi je večina fizikalnih sistemov analogne narave, npr.: čas, temperatura, zračni tlak... To pomeni, da se njihove vrednosti spreminjajo zvezno, če bi narisali recimo graf temperature v odvisnosti od časa, bi dobili sklenjeno krivuljo oziroma funkcijo. Računalnik ni zmožen operirati z analognimi podatki, zato moramo nezvezne za shranjevanje in obdelavo pretvoriti v diskretne (digitalne), torej v takšne, ki se ne spreminjajo zvezno. Pravimo, da jih digitaliziramo. V nadaljevanju se bom posvetil predvsem kodiranju števil, ki že sama po sebi diskretne strukture.

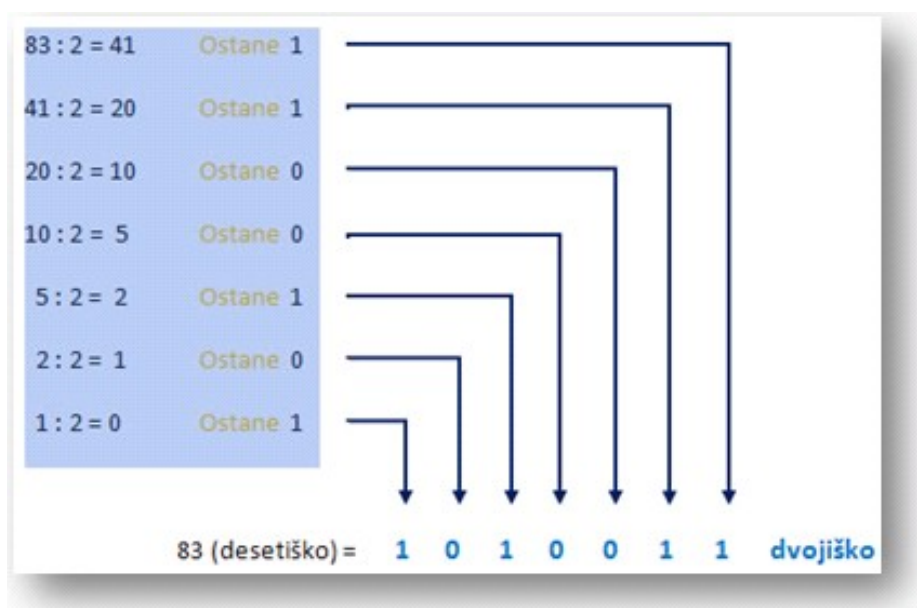
3.1. KODIRANJE ŠTEVIL

Računalnik zna operirati samo z binarnimi podatki, s podatki v dvojiškem številskem sistemu, ki imajo samo 2 signala; 1 in 0, kar predstavlja vklopljeno stikalce, skozenj teče tok (true) ali pa izklopljeno, tok ne teče (false). Torej je treba dobljene diskretne podatke pretvoriti še v binarni kod - zaporedje ničel in enic. To seveda počne računalnik sam, naredi s podatki kar mu je naročeno in jih nato pretvori nazaj v človeku prijaznejšo obliko in prikaže recimo na zaslonu. Pa pogledjmo si nekatere postopke za pretvarjanje med številskimi sistemi:

3.1.1. Pretvorba desetiškega števila v dvojiško

Algoritem je sledeč:

- izberemo poljubno število v desetiškem sistemu
- delimo ga s številom 2, zapišemo si rezultat in ostanek (tudi če je 0)
- z deljenjem nadaljujemo, dokler ne dobimo za rezultat vrednosti 0, takrat zapišemo še zadnji ostanek pri deljenju
- zapis števila v dvojiškem sistemu dobimo tako, da zapišemo niz enic in ničel v obratnem vrstnem redu od tega, kakor je nastajal



Slika 2: Algoritem za pretvorbo iz desetiškega v binarni kod
(Vir: <http://goo.gl/9GacmH>)

3.1.2. Pretvorba desetiškega števila v osmiško

Algoritem za pretvorbo je enak tistemu, za pretvorbo v dvojiški sistem, le da delimo z 8 namesto z 2:

- izberemo poljubno število v desetiškem sistemu
- delimo ga z 8, zapišemo si rezultat in ostanek (tudi če je 0)
- postopek nadaljujemo, dokler ne dobimo za rezultat vrednosti 0, takrat zapišemo še zadnji ostanek

- ostanke pri deljenju zapišemo v obratnem vrstnem redu od tega, kakor smo jih dobili, ta niz števil od 0 do 7 predstavlja izbrano število v osmiškem sistemu

3.1.3. Pretvorba desetiškega števila v šestnajstiško

Tudi ta algoritem deluje po istem principu:

- izberemo poljubno število v desetiškem sistemu
- delimo ga z 16, zapišemo si rezultat in ostanek (tudi če je 0)
- nadaljujemo z deljenjem rezultata in beleženjem ostanka
- ustavimo se, ko dobimo za rezultat 0, zapišemo še zadnji ostanek
- ostanke zapišemo v obratnem vrstnem redu od nastajanja, tako dobimo šestnajstiški zapis izbranega števila



Slika 3: Algoritem pretvorbe iz desetiškega sistema v šestnajstiški
(Vir: <http://goo.gl/GtxOBB>)

Pri zapisovanju števil v šestnajstiškem sistemu pa naletimo na posebnost, saj za zapisovanje v tej obliki potrebujemo več od 10 znakov, kot smo jih vajeni iz desetiškega sistema. Za množico znakov obstaja dogovor, ki vpeljuje črke iz začetka abecede za oznake števil večjih od deset, tako dobimo množico znakov za sestavljanje števil:

$$\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F\}$$

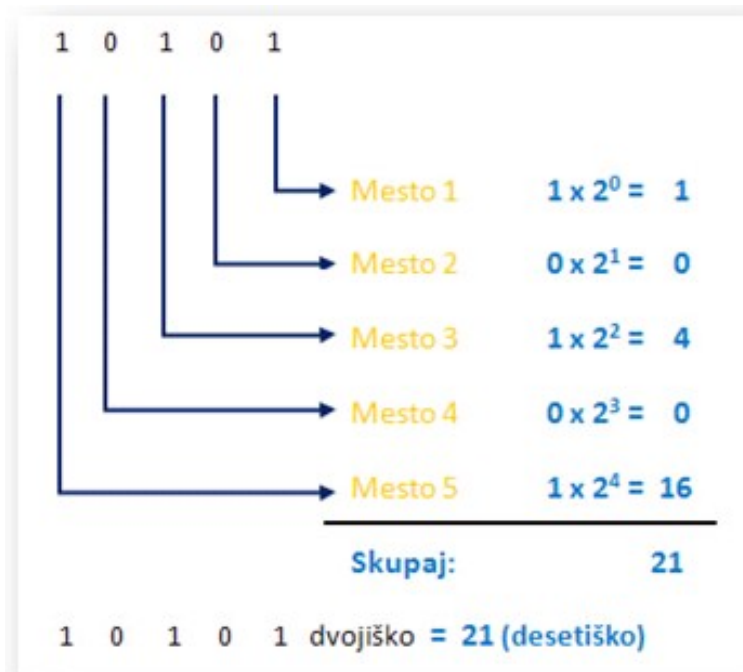
pri čemer:

$$A_{(16)} = 10_{(10)}, B_{(16)} = 11_{(10)}, C_{(16)} = 12_{(10)}, D_{(16)} = 13_{(10)}, E_{(16)} = 14_{(10)}, F_{(16)} = 15_{(10)}.$$

3.2. PRETVORBE IZ DRUGIH SISTEMOV V DESETIŠKI SESTAV

3.2.1. Pretvorba iz dvojiškega sistema

Število iz binarnega zapisa pretvorimo v desetiško tako, da zapišemo polinom pri katerem 1 ali 0 iz n-tega mesta števila, množimo z (n-1) potenco števila 2. Poglejmo si primer:



Slika 4: Postopek pretvorbe iz dvojiškega v desetiški sistem - zapis polinoma
(Vir: <http://goo.gl/Egmg6z>)

3.2.2. Pretvorba iz osmiškega sistema

Postopek je enak prejšnjemu, zapišemo polinom, pri katerem število 0 – 7 iz n-tega mesta, množimo z (n-1) potenco števila 8. Vzemimo za zgled število 421:

$$421_{(8)} = 4 \cdot 8^2 + 2 \cdot 8^1 + 1 \cdot 8^0 = 256 + 16 + 1 = 273_{(10)}$$

3.2.3. Pretvorba iz šestnajstiškega sistema

Postopek se od prejšnjih dveh ne razlikuje in je seveda enak pri pretvorbi števila iz drugih podobnih številskih sistemov (trojiški, šestiški, dvanajstiški...) v desetiškega. Pri pretvorbah iz sistemov višjih od desetiškega, torej tudi pri šestnajstiškem, pa moramo biti pozorni še na pravilen zapis znakov oziroma črk. Vzemimo za zgled število 2CF4:

$$2CF4_{(16)} = 2 \cdot 16^3 + 12 \cdot 16^2 + 15 \cdot 16^1 + 4 \cdot 16^0 = 8192 + 3072 + 240 + 4 = 11508_{(10)}$$

3.3. PRETVORBE IN ZVEZE MED DRUGIMI ŠTEVILSKIMI SISTEMI

Med drugimi številskimi sistemi lahko pretvarjamo tako, da najprej pretvorimo v desetiški sistem z zapisom polinoma nato pa z deljenjem v drug izbran sistem. Obstajajo pa tudi zelo lepe direktne pretvorbe med nekaterimi sistemi in sicer, ko lahko en izmed sistemov zapišemo kot neko potenco drugega z enako osnovo. Iz dvojiškega sistema lahko direktno pretvorimo v osmiški in šestnajstiški sistem, ker je:

$$2^3 = 8 \text{ in } 2^4 = 16$$

in seveda lahko pretvorimo direktno tudi nazaj iz osmiškega in šestnajstiškega v dvojiškega. Iz recimo osmiškega pa lahko v šestnajstiški pretvorimo po dveh poteh, prva je, da najprej pretvorimo v desetiškega in iz tam v šestnajstiškega, druga pa, da namesto v desetiškega preoblikujemo zapis v dvojiškega in nato v šestnajstiškega.

3.3.1. Iz dvojiškega v osmiški

Za to pretvorbo, vzamemo po 3 cifre oziroma bite skupaj, ker je $2^3 = 8$, te pa zapišemo z cifro iz množice za osmiški sistem $\{0, 1 \dots, 7\}$. Vzemimo za primer število $11010_{(2)}$. Najprej ga moramo zapisati na takšno število mest, da bo to večkratnik števila 3. To storimo tako, da mu od spredaj dodamo ničle. Poglejmo zgled:

$$011'010_{(2)} = 32_{(8)}$$

Znak ' je dodan samo za lažje ločevanje med trojicami bitov. Število $011_{(2)}$, vemo da predstavlja $3_{(8)}$ in število $010_{(2)}$ predstavlja $2_{(8)}$. Poglejmo si naslednjo tabelo, ki vsebuje dvojiški in osmiški zapis števil, potrebnih za to pretvarjanje:

Tabela 1: Dvojiški in osmiški zapis

3.3.2. Iz osmiškega v

Postopek je obraten postopku osmiški sistem. Vsak znak v pomočjo tabele iz prejšnjega enic in ničel.

dvojiško	osmiško
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

dvojiški

za pretvorbo iz dvojiškega v osmiškem zapisu pretvorimo s poglavja z zaporednje treh

3.3.3. Iz dvojiškega v

šestnajstiški

Postopek je enak prejšnjemu, le da vzamemo po 4 bite skupaj, ker je $2^4 = 16$, ter jih zapišemo s šestnajstiškim simbolom iz množice {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F}. Za zgled vzemimo število 1100101110, seveda mu moramo spredaj dodati dve ničli, tako da bo mest večkratnik števila 4. Primer:

$$0011'0010'1110_{(2)} = 32E_{(16)}$$

Znak ' je uporabljen za lažje ločevanje med četverkami bitov. Poglejmo si še tabelo vseh potrebnih števil za pretvarjanje:

Tabela 2: Dvojiški in šestnajstiški zapis

3.3.4. Iz šestnajstiškega v

Postopek je tudi tukaj iz dvojiškega sistema v šestnajstiškega sistema s pretvorimo v zaporedje štirih

3.3.5. Pretvorba med sistemom

Pri tej pretvorbi imamo 2 pomočjo desetiškega sistema, izbrano število pretvorimo izmed dveh zgoraj navedenih. pomočjo dvojiškega števila s pomočjo tabel in ničel, te grupiramo v zelene katerega izmed dveh sistemov pretvarjamo, in nato skupine bitov s pomočjo druge tabele pretvorimo v simbole uporabljene za sistem, v katerega pretvarjamo. Pretvorbe s pomočjo desetiškega in dvojiškega sistema veljata v obe smeri, torej iz osmiškega v desetiškega in iz desetiškega v osmiškega.

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

dvojiški

inverzen postopku pretvorbe šestnajstiški. Vsak znak iz pomočjo tabele iz točke 3.3.4. enic in ničel.

osmiškim in šestnajstiškim

možnost. To lahko počnemo s tako da za vmesni korak vanj, nato pa v drug sistem Druga možnost pa je s številskega sistema, izbrano pretvorimo v zaporedje enic skupine, odvisne od tega, v

4. PREDSTAVITEV INFORMACIJ V RAČUNALNIKU

Podatek je poljubna predstavitev dejstva, koncepta ali ukaza na formaliziran način. Biti mora primeren za komunikacijo, interpretacijo ali obdelavo s strani človeka in stroja (računalnika). V računalnik so informacije shranjene bitih, to so najmanjše enote informacije. Beseda bit je izpeljana iz angleške besedne zveze **binary digit** - dvojiška številka. Z njo lahko predstavimo 2 stanji, 1 ali 0, torej true ali false, torej ali na stikalcu je signal (tok) ali ga ni. Kaj pa več stanj? Recimo, štiri strani, gor, dol, levo in desno lahko predstavimo z 2 biti, to pomeni $2^2 = 4$ stanj. Vidimo da so možna 4 stanja: 00, 01, 10, 11. Torej večje število stanj predstavimo z zaporedjem več bitov. Naslednja, večja enota informacije ob bita je bajt, ki vsebuje 8 bitov, z njim lahko predstavimo 2^8 stanj, torej 256. Z n biti lahko predstavimo 2^n stanj. Iz bajta sledi kilobajt (kB), megabajt (MB), gigabajt (GB), terabajt (TB).

Kako so v računalniku predstavljena pozitivna cela števila (ne le števila, temveč vse informacije so tako predstavljene), v binarnem sestavu, ter kako med tem in drugimi številskimi sistemi pretvarjamo, je opisano v poglavju 3.1..

4.1. PROBLEM NEGATIVNIH IN DECIMALNIH ŠTEVIL

Dokler smo imeli opravka s celimi pozitivnimi števili ni bilo problema. Računalnik oziroma procesor pa sam po sebi ne zna računati ne z negativnimi in ne z decimalnimi števili, to ga je treba šele naučiti. Seveda je vse spet samo stvar dogovora predstavitve podatkov in pomena različnih bitov informacij na različnih mestih, tako, da bodo znali računalniki z njimi operirati, nam pa vračati vrednosti v oblikah katere razumemo oziroma s katerimi najlažje delamo.

4.1.1. Negativna cela števila

Negativna števila se da predstaviti na več načinov. Včasih se je uporabljala metoda, da je prvi bit v nizu pomenil predznak števila. Nato so začeli za negativna števila uporabljati inverze pozitivnih. Vendar obe te dve metodi imata slabosti. Sedaj se uporablja boljša metoda, imenovana dvojiški komplement. Do dvojiškega komplementa pridemo tako, najprej naredimo izbranemu dvojiškemu številu inverz, nato pa mu prištejemo 1. Dobljen rezultat pomeni negativno vrednost prvotnega števila. Narediti inverz pomeni zamenjati v vsakem bitu 0 z 1 in 1 z 0. Dvojiški komplement je aritmetično enak odštevanju od 2^n , z njim lahko z enim bajtom, torej z osmimi bitmi predstavimo 256 števil, vendar to pomeni od -128 do +127. Interval dometa za določeno število bitov je sledeč: $[-2^{n-1}, 2^{n-1}-1]$.

4.1.2. Realna števila

Množica realnih števil vključuje tudi tista, ki jih pišemo z decimalno vejico oziroma v računalništvu se namesto nje piše piko. V računalniku so decimalna števila predstavljena s pomočjo metode floating point - plavajoča vejica. Plavajoča vejica je za računalnik prilagojena različica znanstvene notacije, s katero se v tehniki in znanosti rešuje omejitve obsega števil, ki jih lahko predstavimo s fiksno vejico. Vsako realno število lahko zapišemo na več načinov: $1,23 \cdot 10^0 = 12,3 \cdot 10^{-1} = 0,0123 \cdot 10^2$. Standardni zapis s plavajočo vejico je:

$$0,m \cdot r^e$$

- m – mantisa (to so številke xy...z, pri čemer x ni enak 0)
- r – baza
- e – eksponent

Vloga eksponenta je, da nam pove kje je decimalna vejica. Z njegovim spreminjanjem se vejica pomika (plava) vzdolž leve in desne, od tod tudi izvira ime. V računalniku je baza r konstantna tako, da je število enolično predstavljeno le s parom (m, e), ki sta predznačeni števili v fiksni vejici. Natančnost števila predstavljenega v plavajoči vejici je odvisna od števila bitov namenjenih za mantiso. Največje in najmanjše število pa je odvisno od števila bitov namenjenih za eksponent. Še vedno pa lahko z n bitmi predstavimo le 2^n števil, razlika pa je v tem, da lahko predstavimo ulomke, torej realna števila. Iracionalnih števil se v

plavajoči vejici ne da predstaviti. Za enoličen zapis števil poznamo tri pravila, najpogosteje uporabljeno je, da je vejica postavljena pred prvo mesto mantise, torej mantisa se mora začeti s številom, ki ni enak nič, in pred njim je dogovorjeno da stoji vejica. Tako rejenemu številu pravimo normalizirano število.

Pri plavajoči vejici, zaradi normaliziranja obstaja še nekaj posebnosti in sicer, ker mora biti prvi bit mantise v binarnem zapisu enak 1, ga v predstavitvi lahko izpustimo, pri zapisu je pisan v oklepajih. Pri računanju pa ga seveda upoštevamo. Takšnemu zapisu pravimo implicitna predstavitev normalnega bita. Z izpustitvijo bita, ki je vedno 1 pridobimo na dragocenih bitih.

Še nekaj posebnosti je pri predstavitvi nekaterih števil:

- ničli (+0 in -0): eksponent in mantisa imata vse bite 0
- neskončnost ($+\infty$ in $-\infty$): eksponent ima vse bite 1 in mantisa vse bite 0
- neveljavna števila (NaN): eksponent ima vse bite 1 in mantisa je različna od 0

Primer zapisa števila v plavajoči vejici:

0 1000 0110 (1) 000 1000 0101 0000 0000 0000

To je 32-bitni zapis, prvi bit predstavlja predznak, naslednjih 8 bitov je eksponent, naslednjih 24 (23 brez števila v oklepaju – izpuščen bit, ki je vedno enak ena, ko je število normalizirano) števil pa predstavlja mantiso. Najpogostejša formata sta:

- enojna natančnost: 32-bitni zapis
- dvojna natančnost: 64-bitni zapis

pri čemer je 64-biten zapis bolj natančen, ni pa nujno da dvakrat bolj.

5. ZAKLJUČEK

Seminarska naloga mi je bila všeč, saj sem si z izdelavo le-te ponovno priklical v spomin različne stvari o kodiranju, od številskih sistemov, pretvarjanja med sistemi, do predstavitve negativnih in realnih števil v računalniku. Naloga je po mojem mnenju lahko zanimiva za vse, ki jih zanimajo teoretične osnove delovanja računalnika v povezavi z matematiko. Koristna je pa tudi za vse ostale, ki imajo kakor koli kaj opravka z računalnikom. Zdaj, ko poznamo številske sisteme in kako so podatki predstavljeni v računalniku, je čas še za naslednjo zanimivost, povedal bom namreč pogosto goljufijo, ki jo uporabljajo trgovci in proizvajalci. Zaradi različnih nedoslednosti se včasih namesto dvojiških uporabljajo desetiške predpone za zapisovanje za recimo koliko prostora za shranjevanje ima določen trdi disk ali spominska kartica... Vzemimo za zgled zapis 1GB. To se pravilno zapiše v dvojiškem sistemu in pomeni $2^{30} = 1'073'741'824$ bajtov torej 8x toliko bitov. V trgovinah pa bo točno ta oznaka velikokrat pomenila $10^9 = 1'000'000'000$ bajtov. Torej zaradi napačne oznake pa ima naš izdelek, ki naj bi imel 1GB prostora, 7,4% manj prostora, kot je označeno. Vprašanje je, ali je to vedno načrtno. Ne vemo. Ponekod sigurno je. Ob tem pa se mi poraja še eno vprašanje, zakaj se nikoli množično ne zmotijo nam torej kupcem v prid.

6. VIRI IN LITERATURA

1. Številski sistemi. [URL: http://sl.wikipedia.org/wiki/%C5%A0tevilski_sistem]
2. Številski sistemi in pretvorbe.
[URL: <https://sites.google.com/site/programiranjepic/tevilski-sistemi-in-pretvorbe>]
3. Osnove računalništva, predstavitev števil. [URL: <http://goo.gl/O5T3ie>]
4. Kodiranje števil, črk in posebnih znakov. [URL: <http://goo.gl/T7ypFF>]
5. Centralno procesna enota.
[URL: <http://eucilnica.scng.si/mod/resource/view.php?id=6220>]
6. Plavajoča vejica. [URL: http://sl.wikipedia.org/wiki/Plavajo%C4%8Da_vejica]