Kodimi i informacioneve. Paraqitja e tyre ne memorie

2.1 Sistemet e numërimit

Kemi thene qe informacioni brenda kompjuterit paraqitet si nje varg njeshash dhe zerosh. Ky lloj kodimi i informacionit quhet kodim binar. Në këtë kapitull do të trajtojmë sistemet e numërimit në përgjithësi dhe në mënyrë të veçantë sistemin binar sipas të cilit realizohet kodimi binar. Gjithashtu do të ndalemi në paraqitjen në memorie të numrave si dhe në kodimin e karaktereve alfanumerike dhe instruksioneve.

Në përgjithësi një sistem numërimi përcaktohet nga :

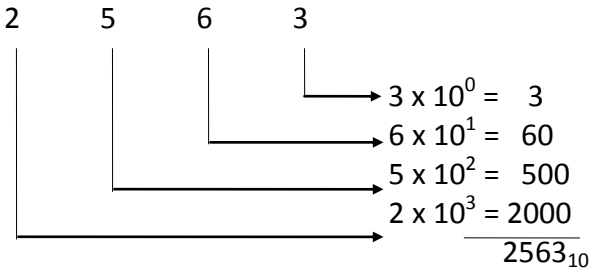
- një numër i plotë pozitiv b që quhet baza e sistemit (b>0);
- një bashkësi shifrash për paraqitjen e numrave, ku çdo element c i saj plotëson kushtin : $0 \leq c < b$;
- veprimet aritmetike : mbledhja, zbritja, shumëzimi, pjestimi.

Meqënëse sistemi dhjetor është më i njohuri për ne, po e fillojmë diskutimin me këtë sistem. Sistemi dhjetor quhet kështu pasi ka si bazë numrin 10, pra b = 10. Bashkësia e shifrave me të cilat paraqiten të gjithë numrat në këtë sistem është:

{ 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 }

Çdo shifër e një numri të plotë në sistemin dhjetor duke filluar nga e djathta në të majtë paraqet njëshet (10^0), qindëshet (10^2), mijëshet (10^3), etj.

Për shembull, numri 256310 (indeksi tregon sistemin) mund të analizohet si më poshtë:



Në përgjithësi për një numër a me k-shifra do të kishim:

$$a = C_{k-1}b^{k-1} + C_{k-2}b^{k-2} + C_{k-3}b^{k-3} + ... + C_1b^1 + C_0b^0 \qquad (*)$$

ku b-baza e sistemit.

Me marrëveshje paraqitje të numrit *a* në sistemin me bazë *b* do të quajmë koeficientët e zbërthimit të mësipërm të renditur njëri pas tjetrit. Pra:

$$a = C_{k-1}C_{k-2}C_{k-3}...C_1C_0$$

Për numrin 2563_{10} (k = 4) do të kishim:

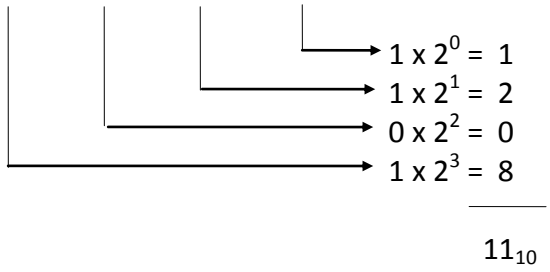
$C_0 = 3$ $C_1 = 6$ $C_2 = 5$ $C_3 = 2$

Sistemi binar quhet kështu pasi baza e tij është numri 2 (b = 2), kurse bashkësia e shifrave: {0, 1}.

Pra, çdo numër në sistemin binar paraqitet vetëm me anë të dy shifrave 0 dhe 1. Është kjo arsyeja që përdoret pikërisht ky sistem për kodimin e informacionit qe qarkullon në kompjuter (kodimi binar).

Duke ju referuar zberthimit (*) ku b = 2, çdo shifër e një numri në sistemin binar duke filluar nga e majta, paraqet njeshe (2^0), dyshe (2^1), katërshe (2^2), teteshe (2^3) etj. P.sh. numri 1011_2 mund të analizohet si me poshtë:

1 0 1 1



Siç duket numri 11 në sistemin dhjetor paraqitet në sistemin binar si 1011 pra:

$$11_{10} = 1011_2$$

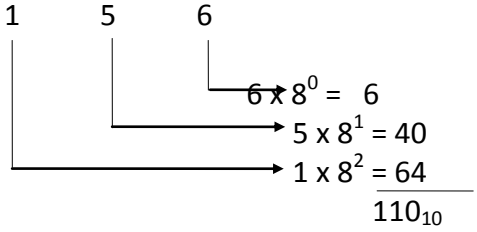
Theksojmë që baza e sistemit duhet të paraqitet në formën e një indeksi për të mënjanuar konfuzionet. Kështu p.sh. 1011 në sistemin binar është krejt e ndryshme me 1011 (njëmijë e njëmbëdhjetë) në sistemin dhjetor.

Veprimet arithmetike në sistemin binar kryhen njësoj si në sistemin dhjetor (arithmetika e zakonshme), veçse rolin e dhjetës këtu e luan dyshi, si p.sh.:

1010_2	11001_2	101_2	$110_2 \mid 10_2$
$+ \quad 110_2$	111_2	$\times 11_2$	$10 \mid 11_2$
10000_2	10010_2	101	10
		101	10
		1111_2	= =

Sistemi oktal quhet kështu pasi baza e tij është numri 8 (b = 8). Numrat në ketë sistem paraqiten me ane të shifrave: { 0, 1, 2, 3, 4, 5, 6, 7 }

Duke ju referuar zbërthimit (*) për numrin 1568 do të kishim:



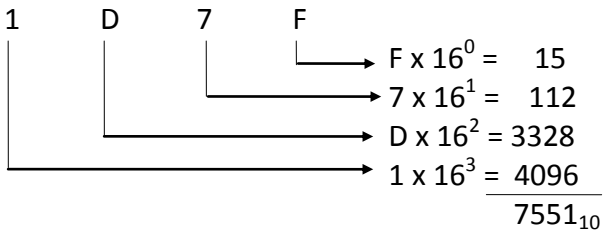
Pra 110 në sistemin dhjetor paraqitet si 156 në sistemin oktal:

$$110_{10} = 156_8$$

Sistemi hekzadecimal quhet kështu pasi baza e tij është numri 16 (b=16). Bashkësia e shifrave në ketë sistem do te ishte: { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15 }
Për të mos patur ngatërresa p.sh. ndërmjet shifres 12 dhe dy shifrave 1 e 2 me marrëveshje shifrat 10, 11, ...,15 paraqiten me germa A, B, C, D, E, F, pra bashkësia e shifrave në këtë sistem është:

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

Duke ju referuar zbërthimit (*) ku b = 16 për numrin $1D7F_{16}$ do të kishim:



Pra 7551 në sistemin dhjetor do të ishte 1D7F në sistemin hekzagonal:

$$7551_{10} = 1D7F_{16}$$

Ne përshkruam disa nga sistemet e numërimit të cilat lidhen ngushtë me kodimin e informacionit që trajtohet nga kompjuteri, por kjo nuk do të thotë qe nuk mund të ndertohen dhe analizohen edhe sisteme të tjere duke u mbeshtetur në përcaktimin e bërë në krye të paragrafit p.sh. sistemet me bazë 5, 7, 9, etj.

2.2 Konvertimi i numrave nga njëri sistem në tjetrin

a) Nga sistemi dhjetor në një sistem me bazë b.
Për të gjetur paraqitjen e nje numri në sistemin me bazë b duhet të përcaktojmë koeficientët C0, C1, C2,... përpara fuqive të bazës b, pra duhet të gjejmë sa b0, b1, b2,.. ka ky numër.
Veprohet në këtë mënyrë:
Pjestohet numri me bazen b. Mbetja ruhet ndërsa herësi pjestohet përsëri me bazën b. Ky veprim përsëritet derisa herësi të bëhet zero. Mbetjet e ruajtura renditen nga fundi në krye.
Për ilustrim le të konvertojmë numrin 23_{10} në sistemin binar.

23:2=11 (1)
11:2= 5 (1)
5:2= 2 (1)
2:2= 1 (0)
1:2= 0 (1)
Pra : $23_{10} = 10111_2$

Duke vepruar njësoj do të gjenim:
$117_{10} = 165_8$ dhe $1456_{10} = 5B0_{16}$

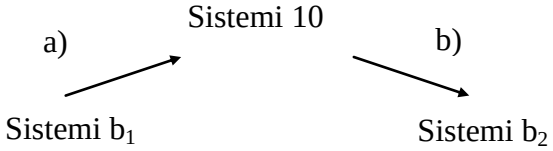
b) Nga sistemi me baze b në sistemin dhjetor.
Ne këtë rast mjafton të zberthehet numri dhe të kryhen veprimet siç tregohet edhe në shembujt e paragrafit 2.1
Keshtu:

$$101_2 = 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 5_{10}$$

ose:

$$1F_{16} = 1 \times 16^1 + 15 \times 16^0 = 31_{10}$$

c) Nga një sistem me bazë b1 në një sistem me bazë b2 .
Në përgjithesi ky kalim realizohet nëpërmjet sistemit dhjetor duke kombinuar konvertimet a) dhe b) sipas skemës:



Do të ndalemi në mënyrë të veçantë në konvertimet nga sistemi binar në oktal dhe hekzadecimal si edhe anasjelltas të cilat realizohen direkt pa patur nevojë të kalohet në sistemin dhjetor.
Konvertimi i numrave oktalë në binarë dhe anasjelltas mund të realizohet lehtësisht duke zëvendësuar çdo shifër oktale me grupin korrespondues prej tre shifrash binarë (kjo korrespondencë rrjedh nga fakti që bazat kanë lidhje fuqie : $8 = 2^3$).

Kështu: $7_8 = 111_2$ ose $6_8 = 110_2$
Le të ilustrojme këtë metodë me një shembull. Për numrin 1348, le të gjejmë paraqitjen e tij në sistemin binar.

1	3	4
↓	↓	↓
001	011	100

Pra: $134_8 = 001011100_2$

Konvertimi i anasjelltë, nga sistemi binar në oktal realizohet duke e ndarë numrin binar në grupe prej tre shifrash duke filluar nga e djathta në të majtë. Pastaj zëvendësojmë secilin prej tyre me shifrën korresponduese oktale siç tregohet në shembullin e mëposhtëm:

101110 111

↓	↓	↓
5	6	7

Pra: $101110111_2 = 567_8$

Konvertimi i numrave hekzadecimalë në binarë dhe anasjelltas.
Veprohet në të njejtën mënyrë si në rastin e mësipërm vetëm se katër shifra të sistemit binar i korrespondojnë nje shifre të sistemit hekzadecimal (16 = 24). Shembujt e mëposhtëm ilustrojne këtë mënyrë konvertimi:

6	B	2
↓	↓	↓
0110	1011	0010

Pra: $6B2_{16} = 011010110010_2$
Po kështu :

1101 1001 1111 1000

↓	↓	↓	↓
D	9	F	8

Pra: $1101100111111000_2 = D9F8_{16}$
Lexuesi mund të provojë rezultatet duke përdorur konvertimin nepërmjet sistemit dhjetor.

2.3 Kodimi i informacionit. Paraqitja e tij në memorie.

Përpunimi numerik me anë të kompjuterit imponon zgjidhjen e një sërë problemeve që nuk ndeshen në aritmetikën e zakonshme.I pari nga këta është paraqitja e shifrave në memorie.
Kujtojmë që memoria qëndrore është e përbërë nga qeliza elementare që janë sisteme me dy gjëndje të qëndrueshme, të cilat interpretohen njëra si 0 dhe tjetra si 1. Prandaj është e natyrshme që për të regjistruar një numër në memorie të përdoret paraqitja e tij në sistemin binar.

2.3.1 Paraqitja e numrave të plotë

Supozojme se kemi një vlerëmbajtës me një numër të caktuar bitesh, p.sh.5. Duke shoqëruar çdo pozicion të vlerëmbajtësit me një fuqi të caktuar të dyshit, do të përftonim numra binarë, si në figurën më poshtë.

$$\begin{array}{ccccc} 0 & 0 & 1 & 0 & 1 \\ 2^4 & 2^3 & 2^2 & 2^1 & 2^0 \end{array} \longleftrightarrow 1*2^2 + 0*2^1 + 1*2^0 = 5_{10}$$

Shihet se ruhet pozicionimi që karakterizon sistemet numerike. Në kompjuter numrat e një tipi të caktuar (të plotë, realë) regjistrohen në një numër të caktuar bitesh.Ne për thjeshtësi po shqyrtojmë rastin e vlerëmbajtësit të mësipërm me 5 bite. Numrat që mund të paraqiten në të janë nga 0 deri në 25- 1 = 31.

Për të realizuar përpunimin numerik të nevojshëm, këtij vlerëmbajtësi i shtohet një qark për të kryer veprimin e mbledhjes dhe një tjetër të zbritjes e ne për thjeshtësi supozojmë që këto qarqe kryejnë vetëm operacionet elementare të shtimi dhe pakësimit me 1, pra respektivisht (+1) dhe (-1). Shumat dhe diferencat merren nga përsëritja e këtyre operacioneve elementare.Në vlerëmbajtësin tonë, duke shtur me 1 numrin 3110 = 111112 merret 0 dhe jo 32 që nuk mund të paraqitet në vlerëmbajtësin 5 bitesh. Në mënyrë analoge, po qe se zvoglojmë me1 numrin 0, nuk merret një numër negativ por numri 3110.

Në analogji me aritmetikën e zakonshme, ku jepet një interpretim gjeometrik i numrave duke përdorur boshtet koordinative,në këtë rast për t'i dhënë një interpretim grafik sjelljes së vlerëmbajtësit tonë është më e përshtatshme të përdorim një rreth si në figurën 2-1.

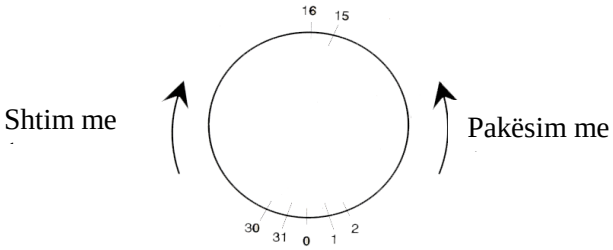


Figura 2-1. Interpretimi grafik i paraqitjes së numrave në vlerëmbajtësin me 5 bite

Operacionit të shtimit i korrespondon rrotullimi në sens antiorar, kurse atij të pakësimit rrotullimi në sens orar.

Paraqitja e numrave me shenjë është një tjetër vështirësi gjatë përpunimeve me kompjuter. Në aritmetikën e zakonshme kuptimi i shenjës është ai i "drejtimit" në bosht, ndërsa kuptimi i largësisë shoqërohet me modulin e numrit.Në kompjuter jemi të detyruar të paraqesim me bite si modulin e një numri, ashtu dhe shenjën e tij.

Metodat kryesore të paraqitjes së numrave negativë janë tre:

1. paraqitja me modul e shenjë
2. paraqitja me komplement të 2
3. paraqitja me komplement të 1.

Për lehtësinë e paraqitjes e të përpunimit që ofron, më tepër përdoret metoda e komplementimit me 2 tek e cila po ndalemi dhe ne shkurtimisht.

2.3.2 Komplementi i një numri

Supozojmë se jepet numri X me n shifra të pjesës së plotë në sistemin me bazë b.Komplementi i këtij numri në sistemin e dhenë përcaktohet si:

$$b^n - X$$

P.sh. po qe se

$X = 64, b = 10, n = 2$

atëherë komplementi në bazën 10 do të jetë

$10^2 - 64 = 36$

N.q.s

$X = 1630, b = 10, n = 4$

Komplementi me 10 do jetë:

$$\begin{array}{r} 10000 \\ -\ 1630 \\ =\ 8370 \end{array}$$

Që këtej del rregulla praktike se komplementi me 10 i një numri gjendet duke analizuar shifrat nisur nga e djathta e tij: zerot deri tek shifra e parë me vlerë transportohen njëlloj, shifrës së parë me vlerë i korrespondon komplementi i saj me 10, ndërsa të tjerave koplementi respektiv me 9.

Në mënyrë analoge , në se

$X = 01011, b = 2, n = 5$

komplementi me 2 do të jetë:

$2^5 - X = 100000 - 01011 = 10101$

Edhe në këtë shembullin tjetër mund të shihet se n.q.s

$X = 1011000, b = 2, n = 7$

atëherë komplementi me 2 gjëndet nga llogaritja e mëposhtme:

$$\begin{array}{r} 10000000 \\ -\ 1011000 \\ =\ 0101000 \end{array}$$

Pra, për komplementimin me 2, rregulla e mësipërme do merrte formën: filluar nga e djathta mbeten të pandryshuara të gjitha zerot deri në bitin e parë me vlerën 1, mbetet i pandryshuar ky bit (pra 1) dhe komplementohen të gjithë bitet e tjera, pra:

$$1 \longrightarrow 0 \text{ dhe } 0 \longrightarrow 1$$

2.3.3 Paraqitja e numrave si komplementë të 2

Në aritmetikën e zakonshme vlera –X, në se X është numër i plotë pa shënjë, mund të interpretohet si sa hapa elementarë mungojnë për të arritur 0 ose mund të lexohet si: 0 – X. Kjo vlen dhe për numrat e vlerëmbajtësit tonë prej 5 bitesh të përmendur më parë dhe në përgjithësi kur është i fiksuar numri i shifrave me të cilat paraqiten numrat. Në rastin tonë numrat paraqiten si modul me 2n, ku n është i biteve në dispozicion.

Kështu që 0 – X dhe 2n – X japin të njëjtin rezultat meqë 0 dhe 2n koinçidojnë. Por 2n – X është njëkohësisht dhe komplementi me 2 i numrit X.Duke përgjithësuar mund të themi se në paraqitjen si komplement i 2, numrat pozitivë paraqiten nga moduli i tyre dhe kanë bitin më të majtë (që përfaqson dhe shënjën) të barabartë me 0. Numrat negativë paraqiten nga komplementi me 2 i numrit korrespondues pozitiv, përfshirë dhe shënjën.Për këtë arsye, numrat negativë e kanë bitin e shënjës të barabartë me 1.

Pra:

$+3 \longrightarrow \quad 00011_2$

$-3 \longrightarrow \quad 11101_2$ (komplementi me 2 i numrit +3)

Shihet që

-ekziston një paraqitje e vetme për numrin 0 (00000), në fakt komplementi me 2 i zeros është gjithashtu zero.
- në 5 bite paraqiten numrat nga 16 (100002) deri në +15 (011112).
- numri më i vogël i paraqitshëm (-16 në 5 bite) nuk ka korrespondues pozitiv, pasi për të paraqitur +16 do të duheshin 6 bite:1 bit për shënjën dhe 5 bite për modulin.

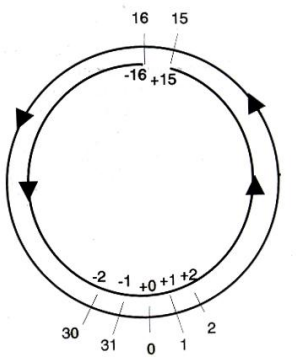


Figura 2-2. Rrethi i jashtëm: numrat e regjistruar në vlerëmbajtës,rrethi i brendshëm: numrat e paraqitur.

Shihet se nuk ka as ndërprerje në paraqitjen e numrave dhe as përsëritje. Duke lëvizur në sens orar nga –16 në +15 kalojmë nëpër vlera rritëse si të numrave të paraqitur ashtu dhe të atyre të regjistruar.Ndërprerja gjatë kalimit nga –1 (paraqitur me 31) në 0 (paraqitur me 0) kapërcehet duke kujtuar që llogaritet moduli 32.
Duke marrë në konsideratë makinën tonë elementare me 5 bite të lidhur me 2 qarqe elektronike për shtimin dhe paksimin me 1 të vlerave të saj, tani në se numrat do të jenë të paraqitur si komplemente të 2, për të kryer shumën mjafton të komandohet shumë herë shtimi dhe paksimi në rastin e llogaritjes së diferencave, pavarësisht nga vlera e operandëve.
Bazuar në gjykimet e mësipërme, gjatësia e fjalës së memories së një kompjuteri përcakton edhe intervalin e numrave të plotë që mund të regjistrohen në të.
Kështu, për një fjalë me gjatësi 16 bite numri më i madh pozitiv që mund të paraqitet është:

$$0111111111111111_2 = 2^{15} - 1 = 32767_{10}$$

dhe numri më i vogël negativ është:

$$1000000000000000_2 = -2^{15} = -32768_{10}$$

Në një kompjuter me fjalë 32 bite ky interval do të ishte:

$$(-2^{31}, 2^{31} - 1) \quad \text{pra} \quad (-2147483648, 2147483647)$$

Përpjekja për të regjistruar numra të plotë jashtë këtij intervali do të shkaktonte derdhje ose fenomenin e njohur si overflow.

2.3.4 Paraqitja e numrave realë me pjesë dhjetore

Në shkrimin e zakonshëm, pika dhjetore është një simbol me funksionin e vetëm që të tregojë se shifrat e numrit në të majtë të saj kanë fuqi pozitive të bazës si b0, b1, etj , ndërsa shifrat në të djathtë fuqi negative b-1, b-2,etj.Në një makinë numerike nuk është e nevojshme që pika dhjetore të paraqitet si veprohet për shënjën, po mjafton ta supozojmë në një pozicion të dhënë.Për lehtësi, në

makinat numerike pika dhjetore supozohet të jetë në bitin që pason atë shënjës.Numrat e plotë hyjnë në këtë paraqitje si rast i veçantë, duke i menduar si numra me pjesë dhjetore zero.
Atëherë, nisur nga përcaktimi i numrave me shënjë dhe duke i pjestuar të gjithë me 2n , numrat mund të paraqiten në formën:

Y = ±y0 * 20 + Y* - numrat pozitivë si komplement i 2

dhe numrat negativë si

$$Y^* = \sum_{i=1}^{n} \bar{y}_i * 2^{-i} + 2^{-n} \quad \text{ku} \qquad \bar{y}_i = (1 - y_i)$$

Vlera Y* < 1, ndërsa y0 mund të marrë vlerat 0 dhe 1 dhe ky mundet ose jo të përfaqsojë dhe shënjën.
Motivet për zgjedhjen e një pozicioni të tillë të pikës dhjetore janë si të karakterit teorik, ashtu dhe praktik.Në fakt, në këtë rast ndodh që:
duke kryer shumëzimin e 2 numrave, pozicioni i pikës dhjetore dhe qarqet që kryejnë këtë veprim nuk kanë pse preokupohen për një ripozicionim të rezultatit.
Po ashtu, gjatë shumëzimeve, meqë X* * Y* <1, nuk ndodh asnjëherë overflow (rrjedhja). Mund të merret një gjendje underflow, pra një gabim trunkimi kur rezultati nuk mund të paraqitet me numrin e disponueshëm të biteve.
Ndërkaq, sistemet e përpunimit mini dhe mikrokompjutera operojnë vetëm mbi numra të plotë sikur pika dhjetore të ishte vendosur djathtas bitit më pak të vlershëm. Këtu nuk hyjnë sistemet që adaptojnë një paraqitje të veçantë të numrave të quajtur me presje notuese (floating point), për të cilët do të flasim më poshtë.

2.3.5 Veprime aritmetike me numra të paraqitur si komplement i 2

Le të jenë X e Y dy numra të dhënë.Shuma e tyre në paraqitjen si komplement i 2 merret duke bërë shumën aritmetike të të gjitha biteve, përfshirë dhe bitin e shenjës: transporti i mundshëm i 1 përtej bitit të shenjës shkarkohet.Fenomeni *overflow* verifikohet në rastin kur X e Y janë me shënjë të njëjtë.
Për zbritjen e këtyre dy numrave mblidhet i mbledhshmi i parë me komplementin me 2 të të mbledhshmit të dytë.
Referuar paraqitjes së numrave në rreth (figura 2-3), shihet se:

a) operacionit të shumës i korrespondon lëvizja nisur nga i mbledhshmi i parë në sens antiorar në një sasi të barabërtë me modulin e të mbledhshmit të dytë nëse ky i fundit është pozitiv dhe në sens orar nëse ky është negativ;
b) çdo herë që kalohet rreth e qark zeros, gjenerohet një zhvendosje 1 që shkarkohet dhe nuk ndryshon paraqitjen;
c) çdo herë që pas kryerjes së një operacioni kalohet nga +15 në –16 dhe anasjelltas ndodh një overflow dhe paraqitja është e gabuar.

Të njëjtat konsiderata vlejnë dhe për zbritjen e numrave.

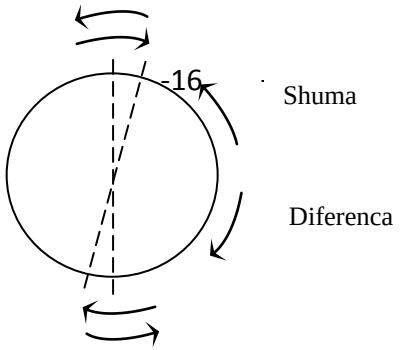


Figura 2-3. Mbledhja dhe zbritja e numrave në paraqitjen si komplement i 2.

Konsideratat e mësipërme shihen dhe në shembujt e mëposhtëm.

01001	(+ 9_{10})	01001	(+ 9_{10})
00100	(+ 4_{10})	11100	(- 4_{10}) si komplement i 2
01101	(+13_{10})	(1) 00101	(+ 5_{10})

↓ shkarkohet

10111	(- 9_{10})	si komplement i 2	01001	(+ 9_{10})	
11100	(- 4_{10})	si komplement i 2	01000	(+ 8_{10})	
(1) 10011	(-13_{10})	si komplement i 2	10001	Overflow	Rezultat i gabuar

↓ shkarkohet

Teknikat për të kryer shumëzimin e dy numrave me shenjë janë shumë komplekse. Për një trajtim të thelluar të këtij argumenti mund t'u drejtoheni teksteve të specializuara.

2.3.6 Paraqitja e numrave reale me presje notuese

Në shumë aplikime, veçanërisht në ato shkencore, merr rëndësi më të madhe madhësia e numrit se sa preçizioni në paraqitjen e tij. Ndërkaq makina të afta të manipulojnë njëlloj numrat me madhësi shumë të ndryshme nga njëra tjetra do të ishin shumë të kushtueshme, Për regjistrimin e numrave të tillë përdoret paraqitja e tyre e quajtur "me presje notuese" (floating point).
Sipas kësaj paraqitjeje numrat jepen në trajtë të standartizuar si:

$$N_{10} = \pm X.YYY * b^{WW}$$

ku pjesa e plotë X shpreh sasinë, numri i shifrave të pjesës dhjetore YYY përcakton saktësinë me të cilën është paraqitur sasia, b është baza e sistemit të numërimit, ndërsa eksponenti WW na informon mbi madhësinë e të dhenës.
Duke përdorur kështu 6 karaktere (X,YYY,WW) dhe duke u kufizuar vetëm në numrat pozitivë, mund të paraqesim numrat nga 0 (0.000*100) deri në 10100 (9.999*1099).
Në paraqitjen e zakonshme me presje fikse, me të njëjtin numër bitesh mund të paraqesim vetëm numrat e përfshirë në intervalin (0,106 –1). Përdorimi i koproçesorëve matematikë ka favorizuar pranimin e standartit në paraqitjen e numrave të tillë, sipas të cilit në sistemin binar numri paraqitet si
N2 = ± 1.XXXXX 2α
ku XXXXX është pjesa fraksionare, ndërsa α eksponenti. P.sh. numri 13.2510 transformohet në paraqitjen e normalizuar të modelit të mësipërm si:

+1101.012 = +1.101012*23

Nga kjo formë kalohet pastaj në paraqitjen standarte në të cilën përdoren grupe me 32 bite të ndarë: 1 bit për shenjën, 8 bite për eksponentin dhe 23 bite për pjesën fraksionare.

Shumica e numrave realë nuk kanë një paraqitje binare të fundme, p.sh.

$0.7_{10} = 0.10110011001..._2$

ku blloku 0110 përsëritet pafundësisht. Nëse do të regjistroheshin vetëm 10 bitet e para, pjesa tjetër do të trunkohej dhe numri 0.7 do të paraqitej në memorie si:

0.1011011002_2

që i korrespondon numrit 0.6992187510. Pra, në këtë rast nuk kemi ekzaktësisht vlerën 0.7. Ky quhet dhe gabim i rrumbullakimit (roundoff error), i cili mund të reduktohet duke përdorur fjalë me gjatësi më madhe, por jo të zhduket fare.

2.3.7 Paraqitja e karaktereve alfanumerike në memorie

Kompjuteri regjistron dhe trajton jo vetëm të dhëna numerike por edhe logjike (e vërtetë, e gënjeshtërt), karaktere alfanumerike si dhe tipa të tjerë të informacioneve jo numerike.
Paraqitja e vlerave logjike në memorie është e lehtë. Një vlerë logjike " e gënjeshtërt" (false) kodohet si 0 ndërsa " e vërtetë " (true) si 1 dhe këto bite regjistrohen.
Sqarojme që karakter alfanumerik quhet çdo karakter që mund të shtypet nga tastjera e kompjuterit pra : gërmat e vogla dhe të mëdha të alfabetit, shifrat 0....9, shenjat e pikësimit, simbole të veçantë si: $, @, &, etj.
Për regjistrimin e tyre në memorie përdoren skema kodimi ku një karakteri i shoqërohet një kod numerik (një numër).

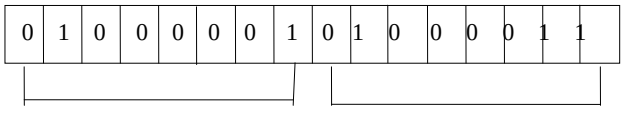
Nga skemat më të përhapura të kodimit janë :

- Skema ASCII (American Standart Code for Information Interchange).
- Skema EBCDIC (Extended Binary Coded Decimal Interchange Code).

Në tabelën e mëposhtëme jepen këto kode për pesë gërmat e para të mëdha të alfabetit.

	ASCII		EBCDIC	
Karakter	Decimal	Binar	Decimal	Binar
A	65	01000001	193	11000001
B	66	01000010	194	11000010
C	67	01000011	195	11000011
D	68	01000100	196	11000100
E	69	01000101	197	11000101
.....				

Karakteret regjistrohen në memorie duke përdorur këto kode binare. Një karakter regjistrohet në 8 bite ose 1 bajt.
Kur fjala është 16 bite aty mund të regjistrohen dy karaktere. P.sh. stringu (vargu) i karaktereve AC mund të regjistrohet në nje fjale 16 bite si me poshtë:



Stringjet e karaktereve që janë më të gjatë së një fjalë memorie zakonisht regjistrohen në dy ose më shumë fjalë të njëpasnjëshme.Deri tani kemi parë si mund të regjistrohen tipa të ndryshme të dhënash në memorien e kompjuterit. Për trajtimin e të dhënave ndërtohen programe, instruksionet e të
cilave gjithashtu duhen regjistruar në memorie.
Për të ilustruar këtë le të marrim një shembull të thjeshtë. Supozojmë që tre vlerat:

$8 = 10000_2$ $24 = 11000_2$ $58 = 111010_2$

janë regjistruar në qelizat në adrese 4, 5, 6. Kërkohet të realizohet veprimi:

8 x 24 + 58

dhe rezultati të regjistrohet në qelizën 7 (për lehtësi adresat e qelizave janë paraqitur në sistemin dhjetor).

Për të kryer këtë llogaritje duhet të ekzekutohen instruksionet e mëposhtme:

- merret përmbajtja e qelizës 4 dhe ngarkohet ajo në regjistrin akumulator të Nj.A.L.
- merret përmbajtja e qelizës 5 dhe llogaritet produkti i saj me vlerën në akumulator.
- merret përmbajtja e qelizës 6 dhe mblidhet me vlerën e akumulatorit.
- regjistrohet përmbajtja e akumulatorit në qelizën 7.

Për të regjistruar këto instruksione në memorien e kompjuterit, ato duhet të paraqiten në formë binare.

Adresat nuk paraqesin problem pasi ato konvertohen lehtësisht në adresa binare:

$4 = 100_2$ $5 = 101_2$ $6 = 110_2$ $7 = 111_2$

Operacionet: ngarkimi, shumëzimi, mbledhja, zbritja, etj. paraqiten me kode numerike që quhen *opcode* p.sh.:

Ngarkimi (LOAD) $= 16 = 10000_2$
Regjistrimi (STORE) $= 17 = 10001_2$
Mbledhja (ADD) $= 35 = 100011_2$
Shumëzimi (MULTIPLAY) $= 36 = 100100_2$

Duke përdorur një pjesë të nje fjale memorie për të regjistruar opcodin dhe pjesën tjetër për adresën e operandit, mund të paraqesim sekuencën e instruksioneve tona në gjuhë makinë:

1. 0001000000000100
2. 0010010000000101
3. 0010001100000110
4. 0001000100000111

Këto instruksione mund të ngarkohen në 4 fjalë të njëpasnjëshme memorie. Kur programi ekzekutohet njësia e kontrollit kap secilin nga këto instruksione, e dekodon atë dhe sipas kërkesës së operacionit përdor Nj.A.L n.q.s. duhet. Programet për kompjuterat e parë kane qënë shkruar në gjuhë makinë. Më vonë u be e mundur të shkruheshin programe në gjuhën assembler në të cilen përdoren mnemonika (emra) në vend të opcodeve numerik dhe emra variablash në vend të adresës numerike.