BROJEVNI SISTEMI

Do sada smo primetili kod konvencionalnih pozicionih brojevnih sistema sa fiksnom osnovom povećavanjem osnove sistema smanjuje se dužina zapisa broja!

Zapis mešovitih brojeva

Bez obzira na osnovu u kojoj se zapisuju brojevi, uvek važi:

1. Mešoviti brojevi se zapisuju u uobičajenom obliku sa tačkom osnove (tzv. radix point) izmedju celobrojnog i razlomljenog dela.
2. Svaki mešovit broj se uvek zapisuje pomoću n cifara, pri čemu se sa $m \leq n$ cifara zapisuje razlomljeni deo, a sa $n-m$ cifara celobrojni deo broja. Ako je broj cifara u celobrojnom delu broja veći od $n-m$ javlja se greška pri zapisu. Ukoliko je broj cifara u razlomljenom delu < m preostale pozicije se popunjavaju nulama. Ukoliko je broj cifara u razlomljenom delu >m tada se zapis broja skraćuje na m cifara u razlomljenom delu.
3. Zapis u fiksnom zarezu: broj cifara u razlomljenom delu uvek fiksiran (i jednak m) bez obzira na veličinu broja.
4. Zapis u pokretnom zarezu: svaki mešovit broj zapisan u osnovi N može da se zapiše kao uredjen par (F,E) čiji su elementi frakcija (F) i eksponent (E) koji su predstavljeni kao brojevi u fiksnom zarezu. Vrednost broja je jednaka $F*N^E$.

PREVOĐENJE BROJEVA

Prevodjenje broja X iz sistema sa osnovom N u sistem sa osnovom M je postupak odredjivanja cifara u zapisu broja X u sistemu sa osnovom M.

Prevodjenje celih brojeva

Rekurentna formula za odredjivanje cifara je: $X_i/M = X_i+1+y_i/M$,

$X_0 = X$

pri čemu se aritmetičke operacije izvode u sistemu sa osnovom N.

Prevodjenje razlomljenog dela

Rekurentna formula za odredjivanje cifara je: $X_i * M = y_{-(i+1)} + X_{-(i+1)}$

$X_{-0} = X$

Aritmetičke operacije se izvode u sistemu sa osnovom N

Specijalni slučaj kodiranja

Ako važi $N = M^s, s > 1$, pri prevodjenju brojeva izmedju sistema sa osnovama N i M se koristi tvrdjenje:

Vrednost broja X u sistemu sa osnovom N zapisana u sistemu sa osnovom M je identična zapisu koji se dobija kodiranjem cifara broja X u sistemu sa osnovom M . Prevodjenje mešovitih brojeva se vrši tako što se posebno prevedu celobrojni i razlomljeni deo i od dobijenih prevoda formira željeni prevod.

ZAPIS NEOZNAČENIH CELIH BROJEVA

- U računaru se celi brojevi zapisuju u n-bitnoj reči u obliku binarnog broja $a_{n-1}a_{n-2}...a_1a_0$.

a_{n-1}	a_{n-2}	..	.....	..	a_1	a_0

Zapis neoznačenih brojeva

- broj čiji zapis ne sadrži znak se naziva neoznačen
- zapis neoznačenih celih brojeva je identičan njihovoj reprezentaciji u binarnom brojevnom sistemu
- za A neoznačen ceo (dekadni) broj zapisan u binarnom obliku važi $A \in [0, 2^n - 1]$

Binarni zapis	Dekadna vrednost
0000000000000000	0
0000000000000001	1
0000000000001111	15
0000000000100000	32
0001000000001111	4111
1000000000000000	32768
1111111111111111	65535
11111111111111111111111111111111	4294967295

Tabela 1: Primeri zapisa nekih neoznačenih brojeva

Zapis označenih celih brojeva

1. znak i apsolutna vrednost

- znak je krajnje levi bit, a ostali bitovi su apsolutna vrednost broja

- dekadna vrednost broja A u ovom zapisu je $A \in [-2^{n-1}+1, 2^{n-1}-1]$

- glavni nedostaci ovog načina zapisa su:

   1. pri izvodjenju računskih operacija za otkrivanje eventualnog prekoračenja neophodno je ispitivati znak i apsolutnu vrednost oba argumenta.

   2. nula se može zapisati na dva načina, npr. u 16-bitnoj reči: +0 = 0000000000000000

      -0 = 1000000000000000

      što dodatno otežava izvodjenje kako računskih operacija tako i operacije poredjenja sa nulom

2. zapis u nepotpunom komplimentu

- krajnje levi bit označava znak broja, dok se ostalih $n-1$ bitova zapisuje:

    1. za pozitivne brojeve kao apsolutna vrednost broja

    2. za negativne brojeve kada se u zapisu apsolutne vrednosti broja A (bez znaka broja) svaka cifra zameni njenim komplementom do najveće cifre brojnog sistema

- dekadna vrednost broja A u ovom zapisu je $A \in [-2^{n-1}+1, 2^{n-1}-1]$

- aritmetičke operacije se jednostavnije izvode u odnosu na prethodni zapis

- i u ovom zapisu se nula može predstaviti na dva načina: +0 = 0000000000000000

  -0 = 1111111111111111

3. zapis u obliku potpunog komplementa

- krajnje levi bit označava znak broja, a ostalih n-1 bitova su vrednost broja A

- vrednost broja A se zapisuje na sledeći način:
  - za pozitivne brojeve kao apsolutna vrednost tog broja
  - za negativne brojeve kao broj koji se dobija kada se na zapis broja A u nepotpunom komplementu doda jedinica na mesto najmanje težine, odnosno kao vrednost 2n-A.

- dekadna vrednost broja A u ovom zapisu je $A \in [-2^{n-1}, 2^{n-1}-1]$

- do sada najjednostavnije izvodjenje aritmetičkih operacija

- u ovom zapisu nula ima jedinstvern zapis kao +0

Primeri zapisa:

- n=16

$$-2^{15} \leq x \leq +2^{15}-1, \quad \text{odnosno}$$
$$-32768 \leq x \leq +32767$$

- n=32

$$-2^{31} \leq x \leq +2^{31}-1, \quad \text{odnosno}$$
$$-2147483648 \leq x \leq +2147483647$$

- n=64

$$-2^{63} \leq x \leq +2^{63}-1, \quad \text{odnosno}$$
$$-9223372036854775808 \leq x \leq +9223372036854775807$$

Prevodjenje izmedju zapisa celih brojeva u dekadnom sistemu i njihovog zapisa u potpunom komplementu – pojednostavljen način

1. Tabela sa vrednostima binarnih pozicija

7	6	5	4	3	2	1	0	binarna pozicija
-128	64	32	16	8	4	2	1	vrednost pozicije

2. Pri prevodjenju iz potpunog komplementa u dekadnu vrednost sabiraju se sve vrednosti koje odgovaraju pozicijama na kojima je 1.

-128	64	32	16	8	4	2	1	vrednost pozicije
1	0	1	1	0	1	0	1	binarni zapis broja
-128	0	+32	+16	0	+4	0	+1	$= -75_{10}$

3. Pri prevodjenje iz dekadnog zapisa u potpuni komplement brojevi se zapisuju kao zbir vrednosti pozicija u binarnom zapisu.

-128	64	32	16	8	4	2	1	vrednost pozicije
0	+64	0	0	+8	0	+2	+1	$= +75_{10}$
0	1	0	0	1	0	1	1	binarni zapis broja

4. zapis uz dodavanje uvećanja

- broj se predstavlja kao zbir njegovog potpunog komplementa i vrednosti k koja je poznata pod nazivom uvećanje ili višak.

Na primer, zapis brojeva $(+12)_{10}$ i $(-12)_{10}$ u 8 bita se pomoću uvećanja 128 formira tako što se sabere 128 sa originalnim brojem i dobijeni zbir se zapiše kao neoznačen ceo broj.

$(128)_{10}+(12)_{10}=(140)_{10} \longrightarrow (10001100)_2$

$(128)_{10}+(-12)_{10}=(116)_{10} \longrightarrow (01110100)_2$

- vrednost uvećanja nema numerički značaj, već je njena jedina funkcija pomeranje reprezentacije broja u potpunom komplementu

- vrednost uvećanja se obično bira tako da ima istu masku bitova kao i najmanji negativan broj. Tada su brojevi u ovom zapisu sortirani, ako se posmatraju kao neoznačeni celi brojevi.

Dekadna vrednost	Znak i apsolutna vrednost	Nepotpuni komplment	Potpuni komplement	Uvećanje 128
+127	01111111	01111111	01111111	11111111
+64	01000000	01000000	01000000	11000000
+32	00100000	00100000	00100000	10100000
+16	00010000	00010000	00010000	10010000
+15	00001111	00001111	00001111	10001111
+10	00001010	00001010	00001010	10001010
+9	00001001	00001001	00001001	10001001
+8	00001000	00001000	00001000	10001000
+7	00000111	00000111	00000111	10000111
+6	00000110	00000110	00000110	10000110
+5	00000101	00000101	00000101	10000101
+4	00000100	00000100	00000100	10000100
+3	00000011	00000011	00000011	10000011
+2	00000010	00000010	00000010	10000010
+1	00000001	00000001	00000001	10000001
+0	00000000	00000000	00000000	10000000

-0	10000000	11111111	---	---
-1	10000001	11111110	11111111	01111111
-2	10000010	11111101	11111110	01111110
-3	10000011	11111100	11111101	01111101
-4	10000100	11111011	11111100	01111100
-5	10000101	11111010	11111011	01111011
-6	10000110	11111001	11111010	01111010
-7	10000111	11111000	11111001	01111001
-8	10001000	11110111	11111000	01111000
-9	10001001	11110110	11110111	01110111
-10	10001010	11110101	11110110	01110110
-15	10001111	11110000	11110001	01110001
-16	10010000	11101111	11110000	01110000
-32	10100000	11011111	11100000	01100000
-64	11000000	10111111	11000000	01000000
-127	11111111	10000000	10000001	00000001
-128	---	---	10000000	00000000

Tabela 2: Zapis označenih celih brojeva u obliku znaka i apsolutne vrednosti, nepotpunog komplementa, potpunog komplementa i uvećanja od 128 u binarnoj reči dužine 8

Konverzija izmedju zapisa različitih dužina

- ceo broj zapisan u binarnoj reči dužine n treba upisati u binarnu reč dužine m.

- Ako je $m < n$ upisivanje nije moguće izvesti korektno zbog mogućeg gubitka značajnih cifara.
- Ako je $m = n$ konverzija ne postoji.
- Ako je $m > n$ tada način konverzije zavisi od načina zapisa celog broja.

1. Znak i apsolutna vrednost: bit za znak se pomeri na mesto najveće težine i ostala mesta se popune nulama.
2. Nepotpuni i potpuni komplement: upisuje se a_{n-1} na sve pozicije i u zapisu broja, gde važi $n \leq i < m$. Na primer, za $n = 8, m = 16$

Dekadna vrednost	8-bitna reč	16-bitna reč	Zapis
+5	00000101	0000000000000101	nepotpuni komp.
–5	11111010	1111111111111010	nepotpuni komp.
+9	00001001	0000000000001001	potpuni kompl.
–9	11110111	1111111111110111	potpuni kompl.

ARITMETIKA

1. Promena znaka

• Znak i apsolutna vrednost – komplementira se bit za znak broja

Dekadna vrednost	Binarni zapis
+9	00001001
+5	00000101
-5	10000101
-9	10001001

• Nepotpuni komplement – komplementiranje svake cifre u binarnom zapisu broja, uključujući i mesto za znak.

Dekadna vrednost	Binarni zapis
+9	00001001
+5	00000101
-5	11111010
-9	11110110

- U zapisu pomoću potpunog komplementa promena znaka broja se vrši u dva koraka:

1. U prvom se izvrši komplementiranje svake cifre do najveće cifre brojnog sistema (u binarnom sistemu do jedinice), uključujući i mesto za znak.
2. U drugom se dobijeni broj sabere sa jedinicom, pri čemu se sabiranje obavlja po pravilima za sabiranje neoznačenih brojeva

+9 = 00001001 potpuni komplement 00000000 = 0 10000000 = -128

11110110 1. korak 11111111 01111111

+00000001 2. korak +00000001 +00000001

-9 = 11110111 rezultat 1|00000000 = 0 10000000 = ?

pri čemu se prekoračenje ignoriše

Prekoračenje

- prekoračenje nastaje kada se aritmetičkim operacijama nad brojevima koji su zapisani sa n-1 cifrom dobije broj zapisan sa n cifara

$$
\begin{array}{l}
A = \boxed{a_n}\, a_{n-1}a_{n-2}\ldots a_1a_0 \\
B = \boxed{b_n}\, b_{n-1}b_{n-2}\ldots b_1b_0 \\
\hline
C = \boxed{c_n}\, c_{n-1}c_{n-2}\ldots c_1b_0
\end{array}
$$

- broj sa cifrom više naziva se modifikovani broj

- kako otkriti da li je došlo do prekoračenja?

- sabiranje neoznačenih brojeva: pojavljuje se prenos sa pozicije najveće težine

- znak i apsolutna vrednost: apsolutne vrednosti se sabiraju kao neoznačeni

- potpuni komplement: dobija se rezultat suprotnog znaka; ignorišemo ga

a) 14 + 10

14	=	00001110
10	=	00001010
24	=	00011000

b) 252 + 5

252	=	$\boxed{0}$ 11111100
5	=	$\boxed{0}$ 00000101
***	=	$\boxed{1}$ 00000001

a) +14 + 10

+14	=	0|0001110
+10	=	0|0001010
24	=	0|0011000

b) +127 + 3

+127	=	$\boxed{0}$ 0|1111111
+3	=	$\boxed{0}$ 0|0000011
***	=	$\boxed{1}$ 0|0000010

a) +14 + 10

A=+14	=	00001110
B=+10	=	00001010
C'	=	0|00011000
C=+24	=	00011000

b) +100 + 65

A=+100	=	01100100
B=+65	=	01000001
C'=	=	0|10100101
C=***	=	10100101

Prekoračenje jer se sabiranjem dva pozitivna dobija negativan broj.

c) +127 -10

A=+127	=	01111111
B=-10	=	11110110
C'=	=	1|01110101
C=+117	=	01110101

d) -100-(+65)=-100+(-65)

A=-100	=	10011100
B=-65	=	10111111
C'=	=	1|01011011
C=***	=	01011011

Prekoračenje jer se sabiranjem dva negativna dobija pozitivan broj.

LOGIČKE OSNOVE OBRADE PODATAKA

- oznaka u matematici ⊥ ⊤ $\neg$ $\wedge$ $\vee$

oznaka u računarstvu 0 1 – · +

Osnovni zakoni

$A \cdot B = B \cdot A$	$A + B = B + A$	Zakon komutacije
$(A \cdot B) \cdot C = A \cdot (B \cdot C)$	$(A + B) + C = A + (B + C)$	Zakon asocijacije
$A \cdot (B + C) = (A \cdot B) + (A \cdot C)$	$A + (B \cdot C) = (A + B) \cdot (A + C)$	Zakon distribucije
$1 \cdot A = A$	$0 + A = A$	Neutralni element
$A \cdot \overline{A} = 0$	$A + \overline{A} = 1$	Inverzni element

Identiteti i pravila

$A \cdot 0 = 0$	$A + 1 = 1$	
$A \cdot A = A$	$A + A = A$	
$\overline{\overline{A}} = A$		Dvostruka negacija
$(A \cdot B) \cdot C = A \cdot B \cdot C$	$(A + B) + C = A + B + C$	Brisanje zagrada
$\overline{A \cdot B} = \overline{A} + \overline{B}$	$\overline{A + B} = \overline{A} \cdot \overline{B}$	De Morganove teoreme

Svaka funkcija f: $(A_1, A_2, A_3,...A_n) \longrightarrow \{0,1\}$ naziva se logička funkcija, $A_1, A_2, A_3,...A_n$ – logičke promenljive

Argument	Vrednost		Naziv	Oznaka
A	0	1		
Funkcija				
f_{10}	0	0	nula funkcija	
f_{11}	0	1	identitet	
f_{12}	1	0	negacija	$\neg A$
f_{13}	1	1	jedinična funkcija	

Tabela 1: Logičke funkcije sa jednim argumentom

Argument	Vrednost				Naziv	Oznaka
A	0	0	1	1		
B	0	1	0	1		
Funkcija						
f_{20}	0	0	0	0	nula funkcija	
f_{21}	0	0	0	1	konjunkcija	$A \wedge B$
f_{22}	0	0	1	0	negacija implikacije od A ka B	$A \wedge \neg B = \neg(A \Rightarrow B)$
f_{23}	0	0	1	1	prva projekcija	
f_{24}	0	1	0	0	negacija implikacije od B ka A	$\neg A \wedge B = \neg(B \Rightarrow A)$
f_{25}	0	1	0	1	druga projekcija	
f_{26}	0	1	1	0	ekskluzivna disjunkcija	$(A \wedge \neg B) \vee (\neg A \wedge B)$
f_{27}	0	1	1	1	disjunkcija	$A \vee B$
f_{28}	1	0	0	0	Pirsova (Lukašievičeva) funk.	$A \downarrow B$
f_{29}	1	0	0	1	ekvivalencija	$(A \Leftrightarrow B)$
f_{2A}	1	0	1	0	negacija druge projekcije	
f_{2B}	1	0	1	1	implikacija od B na A	$B \Rightarrow A$
f_{2C}	1	1	0	0	negacija prve projekcije	
f_{2D}	1	1	0	1	implikacija od A na B	$A \Rightarrow B$
f_{2E}	1	1	1	0	Šeferova funkcija	$A \uparrow B$
f_{2F}	1	1	1	1	Jedinična funkcija	

Tabela 2: Logičke funkcije sa dva argumenta

Logički elementi

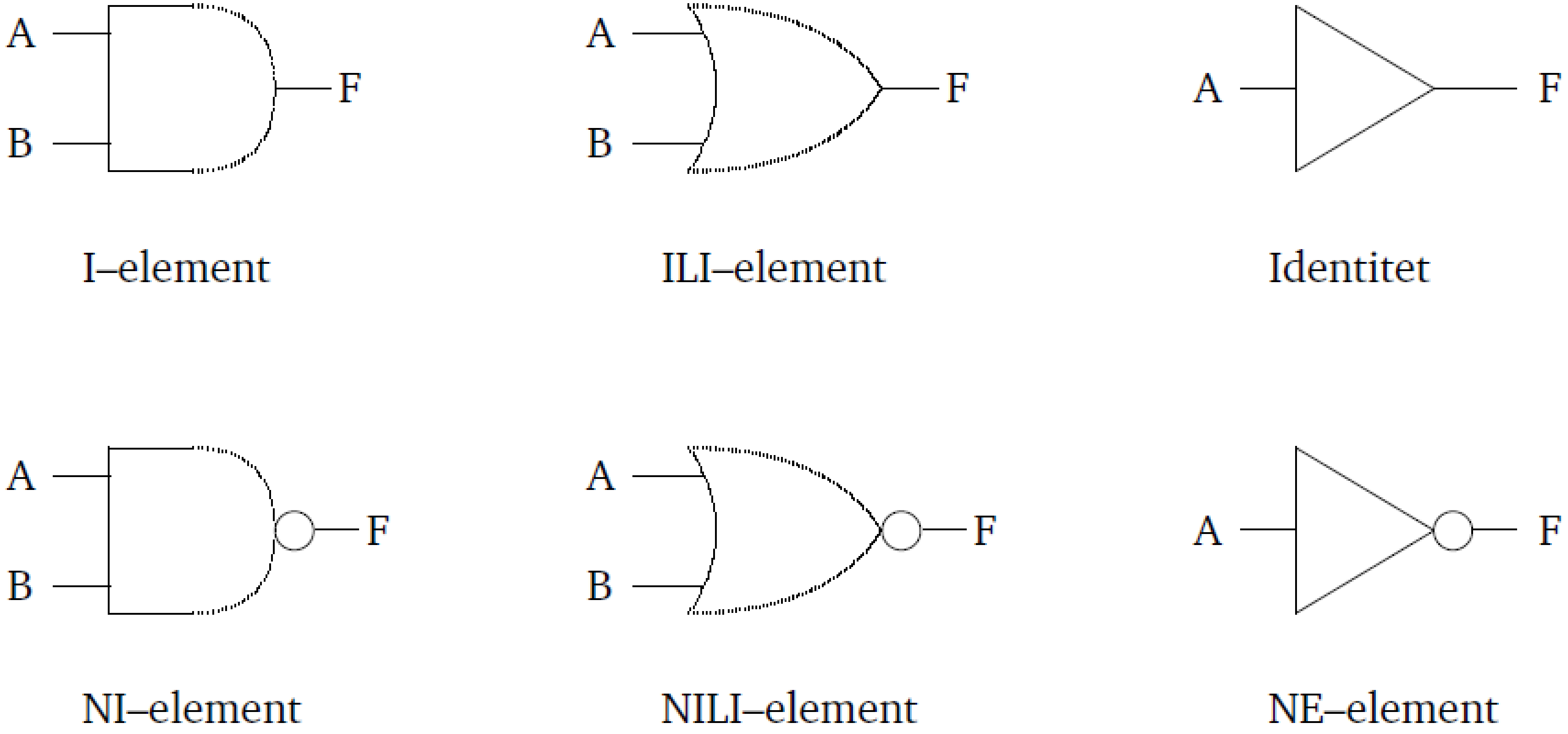
• su fizički objekti koji implementiraju neku od funkcija algebre logike

• predstavljaju osnovne komponente svih elektronskih kola računara

• argumenti funkcija odgovaraju ulaznim veličinama logičkih elemenata, a vrednosti funkcija njihovim izlaznim veličinama

Funkcija algebre logike	Naziv logičkog elementa	Naziv na engleskom
Negacija	NE–element	NOT–element
Konjunkcija	I–element	AND–element
Disjunkcija	ILI–element	OR–element
Šeferova funkcija	NI–element	NAND–element
Pirsova funkcija	NILI–element	NOR–element

Osnovne logičke funkcije i logički elementi



Osnovni logički elementi