Glava 2

Predstavljanje podataka

ZA PREDSTAVLJANJE podataka u memoriji računara koriste se različiti formati zavisno od toga o kakvim se podacima radi i koje će se operacije u procesu obrade podatatka nad njima izvršavati. Zajedničko za sve formate je da se podaci u memoriji računara registruju u obliku binarnih nizova za čije se generisanje u osnovi koriste dva opšta koncepta predstavljanja podataka: binarni brojni sistemi i binarni kodovi.

2.1 Brojni sistemi

Pod brojnim sistemom podrazumevamo skup pravila definisanih za predstavljanje brojnih vrednosti podataka. U praksi danas najširu primenu imaju pozicioni brojni sistemi u koje se svrstava i dekadni sistem brojeva koji koristimo u svakodnevnom radu sa brojevima. Zajedničko za pozicione brojne sisteme je da se za predstavljanje brojnih vrednosti koristi određeni skup cifara, pri čemu vrednost koju označava određena cifra zavisi od pozicije na kojoj se u zapisu brojeva nalazi. Na primer, u dekadnom brojnom sistemu koristimo cifre: 0,1,2,3,4,5,6,7,8 i 9 ali u broju 777.7 cifra 7 označava četiri različite vrednosti: u prvom slučaju je to sedam stotina (7×100), u drugom sedam desetic (7×10), u trećem sedam jedinica (7), a u četvrtom sedam desetih delova jedinice (7×0.1).

Ako jedna cifra ima uvek istu vrednost, bez obzira na kom se mestu u zapisu nalazi, onda je to nepozicioni brojni sistem. Tipičan primer je rimski brojni sistem sa ciframa: I,V,X,C,D,M.

U opštem slučaju zapisu broja u pozicionom brojnom sistemu sa osnovom N, gde je N broj različitih cifara brojnog sistema, oblika

$$(X)_N = x_n x_{n-1} \cdots x_0, x_{n-1} \cdots x_m, \tag{2.1}$$

odgovara brojna vrednost određena relacijom

$$(X) = \sum_{i=-m}^{n} x_i N^i$$
$$= x_n N^n + x_{n-1} N^{n-1} + \ldots + x_0 + x_{-1} N^{-1} + \ldots + x - mN^{-m} \quad (2.2)$$

gde su sa x_i označene cifre brojnog sistema

U računarskoj tehnici značajnu primenu imaju različiti pozicioni brojni sistemi među kojima su najznačajniji: binarni, oktalni i heksadecimalni. U svim ovim sistemima koriste se ista pravila za predstavljanje brojeva ali se razlikuju skupovi cifara pomoću kojih se izražavaju brojne vrednosti. U binarnom brojnom sistemu koriste se samo cifre 0 i 1, u oktalnom skup cifara $\{0, 1, 2, 3, 4, 5, 6, 7, 8\}$, a u heksadecimalnom skup: $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F\}$. Utabeli 2.1 dati su neki primeri dekadnih brojeva i odgovarajući ekvivalenti u binarnom, oktalnom i heksadecimalnom brojnom sistemu.

Tab. 2.1: Primeri brojeva u različitim brojnim sistemima

Dekadni	Binarni	Oktalni	Heksadecimalni
1	1	1	1
7	111	7	7
8	1000	10	8
9	1001	11	9
15	1111	17	*F*
16	10000	20	10
17	10001	21	11

Predstavljanje brojeva u heksadecimalnom brojnom sistemu se vrlo široko koristi u računarskoj tehnici, naročito kada treba štampati ili prikazati binarne sadržaje, jer su zapisi kompaktniji i lakše se prate.

Pošto se u običnom životu najčešće koristi decimalni brojni sistem, potrebno je pri unošenju numeričkih podataka u računar pretvoriti decimalne brojeve u binarne. Takođe je potrebno po završetku obrade podataka u računaru izlazne rezultate iz binarnog pretvoriti u decimalni oblik.

U slučaju konverzije decimalnih brojeva u binarni oblik posmatraćemo odvojeno konverziju celih i razlomljenih brojeva. Konverzija celog decimalnog broja N u binarni broj može se primeniti primenom sledećeg algoritma:

1. Ispitati da li je N paran broj,
2. a) Ako je N neparan, zapisati 1 u rezultat i formirati novu vrednost $N = N - 1$. Skok na 3.

   b) Ako je N paran, zapisat 0 u rezultat.
3. Naći novu vrednost deljenjem N iz koraka 2 sa 2.
4. a) Ako je $N > 1$, vratiti se na korak 1 i ponoviti postupak,
   b) ako je $N = 1$ upisati 1 u rezultat.

Pri upisivanju jedinica i nula u rezultat prvo se upisuje *bit najmanje težine* (engl. Least Significant Bit) koji se skraćeni se obeležava sa - LSB, a ostali bitovi se upisuju levo od njega. Poslednji upisani bit je *bit najveće težine* (engl. Most Significant bit) koji se skraćeno obeležva sa -MSB. Dobijeni broj je binarni ekvivalent decimalnog broja N. U sledećem primeru je detalno opisana konverzoja decimalnog broja u binarni.

Primer 2.1 *Konverzija decimalnog broja* 109_{10} *u binarni broj. Postupak konverzije se može prikazati tabelarno na sledeći način*

Broj	Aktivnost	Bit	
109	Neparan - oduzmi 1	1	LSB
108	Podeli sa 2		
54	Paran - podeli sa 2	0	
27	neparan - oduzmi 1	1	
26	Podeli sa 2		
13	Neparan - oduzmi 1	1	↓
12	Podeli sa 2		
6	Paran-podeli sa 2	0	
3	Neparan - oduzmi 1	1	
2	Podeli sa 2		
1	Kraj algoritma	1	MSB

Dakle, $109_{10} = 1101101_2$.

Pri konverziji pravog decimalnog razlomka koristi se sličan algoritam samo proces počinje određivanjem bita najveće težine, odnosno onog koji se nalazi neposredno iza tačke koja razdvaja celobrojni od razlomljenog dela broja. Algoritam za konverziju se sastoji samo od dva koraka:

1. Formirati novi broj N množenjem N sa 2.
2. a) Ako je $N > 1$, upisati 1 u rezultat i oduzeti 1 od N. Skok na korak 1.
   b) Ak je $N < 1$, upisati 0 u rezultat. Skok na korak 1

Postupak se završava kada se dobije $N = 0$.

U sledećem primeru je detalno opisana konverzoja razlomljenog decimalnog broja u binarni.