

5. lekce

Převod z desítkové do dvojkové soustavy

Miroslav Jílek

05 – Číselné soustavy - princip, dvojková číselná soustava

Princip číselných soustav:

$$k_n \cdot z^n + \dots + k_2 \cdot z^2 + k_1 \cdot z^1 + k_0 \cdot z^0 + \frac{q_1}{z^1} + \frac{q_2}{z^2} + \frac{q_3}{z^3} + \dots + \frac{q_n}{z^n}$$

z – základ číselné soustavy

k – koeficient řádu celých čísel

q – koeficient řádu desetinné části čísla

V desítkové soustavě zapíšeme číslo 123,456

$$1 \cdot 10^2 + 2 \cdot 10^1 + 3 \cdot 10^0 + \frac{4}{10^1} + \frac{5}{10^2} + \frac{6}{10^3}$$

upravíme:

$$100 + 20 + 3 + 0,4 + 0,05 + 0,006$$

sečteme:

$$123,456$$

Převod z desítkové do dvojkové číselné soustavy

Základ číselné soustavy je číslo 2, Soustava používá dvě číslice – 0 a 1

Číslo 57,28 zapsané v desítkové soustavě převedeme do dvojkové soustavy:

Číslo rozdělíme na celočíselnou část (57) a desetinnou část (0,28)

Převod celočíselné části:

Dokud je číslo větší než 0 dělíme dvěma a zapisujeme zbytky po dělení v opačném pořadí, než jsme je vypočetli:

$$57 : 2 = 28 + 1$$

$$28 : 2 = 14 + 0$$

$$14 : 2 = 7 + 0$$

$$7 : 2 = 3 + 1$$

$$3 : 2 = 1 + 1$$

$$1 : 2 = 0 + 1$$

celočíselná část je **111001**

Převod desetinné části:

Dokud je není výsledek roven nule nebo dokud jsme nevytvořili požadovaný počet pozic násobíme desetinnou část dvěma. Pokud se velikost výsledku zvětší o řád, pak číslice tohoto nového řádu číslicí ve dvojkové soustavě a dále násobíme zbylé číslice:

$$28 * 2 = 056$$

$$56 * 2 = 112$$

$$12 * 2 = 024$$

$$24 * 2 = 048$$

$$48 * 2 = 096$$

$$96 * 2 = 192$$

$$92 * 2 = 184$$

$$84 * 2 = 168$$

desetinná část je **01000111**

Číslo 57,28 v desítkové soustavě je **111001.01000111** ve dvojkové soustavě.

Prověrka:

$$1*2^5+1*2^4+1*2^3+0*2^2+0*2^1+1*2^0+0*2^{-1}+1*2^{-2}+0*2^{-3}+0*2^{-4}+0*2^{-5}+1*2^{-6}+1*2^{-7}+1*2^{-8} =$$

$$32+16+8+0+0+1+0+1/4+0+0+0+1/64+1/128+1/256$$

$$57 + 64/256 + 4/256 + 2/256 + 1/256$$

$$57 + 71/256$$

$$57 + 0,27734375$$

$$57,27734375$$

K dosažení vyšší přesnosti bychom museli přidat další desetinné pozice!

Záporná čísla ve dvojkové soustavě

Možnosti vyjádření záporného čísla v počítači:

- a) přímý kód
- b) doplňkový kód
- c) aditivní kód

Pro vyjádření čísla musíme předem definovat, kolika bitové je dané číslo!

Přímý kód

Převédeme absolutní hodnotu čísla do dvojkové soustavy a před číslo doplníme znak znaménka (0 je kladné číslo, 1 je záporné číslo). Pro osmibitové číslo je jeden bit vyhrazený pro znaménko a sedm pro absolutní hodnotu čísla.

Např.: 8 bitové číslo kladné bude +0 až 127 a 8 bitové číslo záporné bude -127 až -0

Pozor: nula má dvě hodnoty: -0 a +0!

$10_{(10)}: 00001010_{(2)}$

$-10_{(10)}: 10001010_{(2)}$

Doplňkový kód

Převédeme absolutní hodnotu čísla do dvojkové soustavy. Pokud je číslo kladné, pak je toto výsledek. Pokud je záporné, pak všechny bity znegujeme a přičteme jedničku.

$$+19_{(10)} \quad 00010011_{(2)}$$

$$-19_{(10)} \quad 11101101_{(2)}$$

Záporné číslo se pozná tak, že má na prvním bitu jedničku!

Aditivní kód

Má aditivní konstantu. Aditivní konstanta je číslo, které se ke každému číslu přičítá. Pro osmibitové číslo se používá nejčastěji číslo 127 nebo 128. Je možné ale použít jakékoli číslo. Toto číslo musí být konstantní pro celý systém čísel, který v projektu používáme.

Pro osmibitové číslo a aditivní konstantu 127 budou čísla:

$$+19_{(10)} \text{ bude: } +19_{(10)} + 127_{(10)} = 146_{(10)} = 10010010_{(2)}$$

$$-19_{(10)} \text{ bude: } -19_{(10)} + 127_{(10)} = 108_{(10)} = 01101100_{(2)}$$

Zobrazení čísel v pohyblivé desetinné čárce

Číslo vyjádříme pomocí formule: $C = M * Z^E$

M – mantisa

Z – je základ čísla – u dvojkové soustavy je to 2

E – exponent

Např.: $53_{(10)}$ $00110101_{(2)}$ bude zapsáno jako $1.10101E101_{(2)}$

Za první jedničku vložíme tečku a za poslední jedničku vložíme znak E a číslo, které reprezentuje exponent.

$$1.10101E101_{(2)} \text{ bude } \left(1 + \frac{1}{2} + \frac{0}{4} + \frac{1}{8} + \frac{0}{16} + \frac{1}{32}\right) \cdot 2^5 = \frac{53}{32} * 32 = 53$$

Převeďte číslo $82_{(10)}$ do dvojkové soustavy v pohyblivé desetinné čárce:

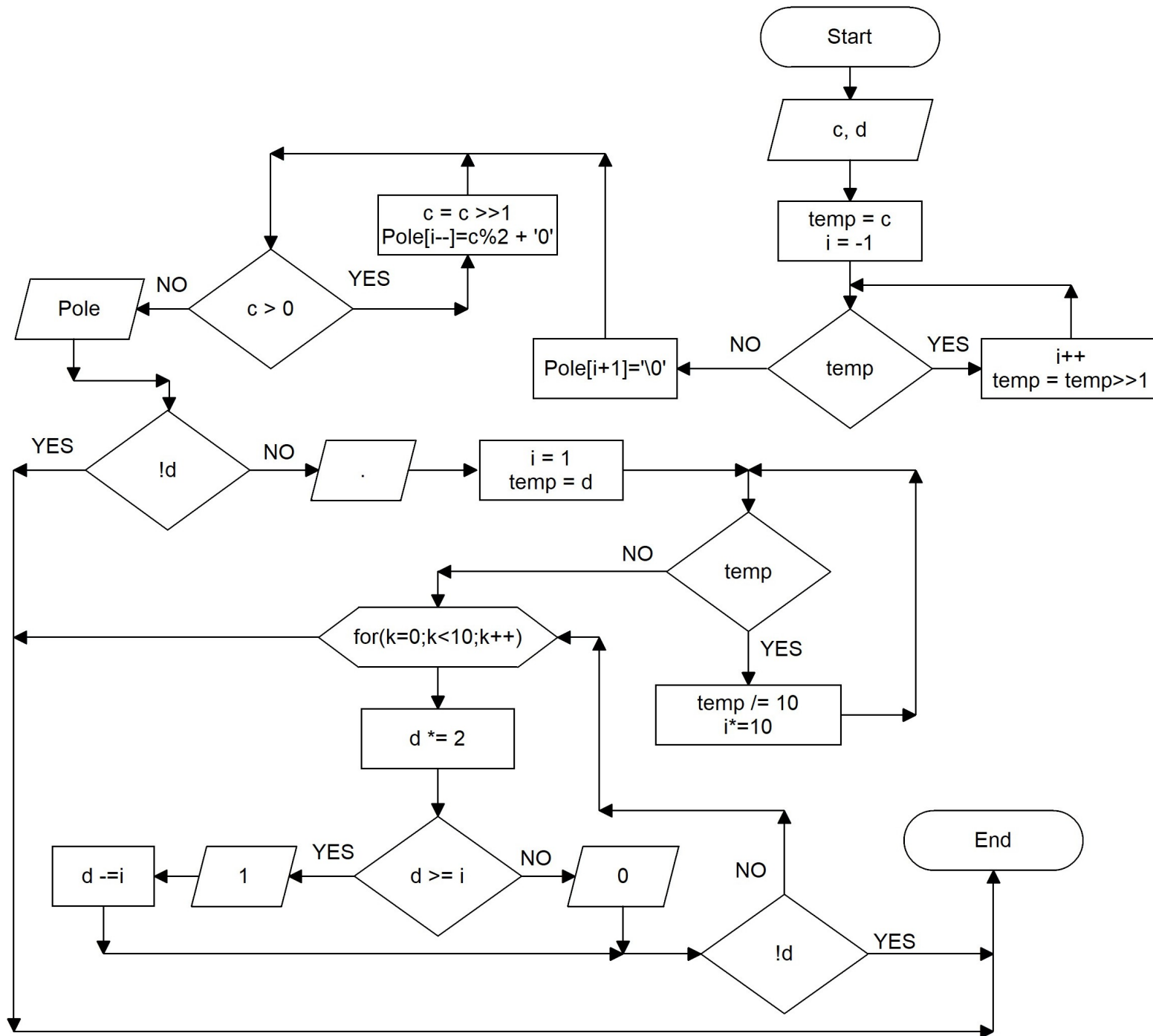
Převeďte číslo $82_{(10)}$ do dvojkové soustavy v pohyblivé desetinné čárce:

V dvojkové soustavě je číslo 82 : 1010010

V pohyblivé čárce : 1.01001E110

Zpětná kontrola:

$$\left(1 + \frac{0}{2} + \frac{1}{4} + \frac{0}{8} + \frac{0}{16} + \frac{1}{32}\right) \cdot 2^6 = \frac{32 + 0 + 8 + 0 + 0 + 1}{32} * 64 = \frac{41}{32} * 64 = 41 * 2 = 82$$



```

#include <stdio.h>

int main (void)
{
    //vystup na deset desetinnych pozic
    int temp,c,d,i=-1,k;
    char t, Pole[33];
    printf("Zadej desetinne cislo: ");
    if (scanf("%d",&c)!=1)
    {
        printf("Nespravny vstup!");
        return 1;
    }
    t=getchar(); //stejne jako scanf("%c",&t);
    switch(t)
    {
        case '!':
            if ((scanf("%d",&d)!=1)||((d<0)))
            {
                printf("Nespravny vstup!");
                return 1;
            }
            break;
        case '\n':
            d=0;
            break;
        default:
            printf("Nespravny vstup!");
            return 1;
    }
    temp=c;
    while(temp) //urceni poctu pozic celociselné casti
    {

```

```

        temp=temp>>1;
        i++;
    }
    Pole[i+1]='\0';
    while(c>0)
    {
        Pole[i--]=c%2+'0'; // pricteni ascii 0, prevod int na char
        c=c>>1;
    }
    printf("%s",Pole);
    if (!d) return 0;
    putchar('.'); //stejne jako printf(".");
    i=1;
    temp=d;
    while(temp) //pocet radu v desitkove soustave
    {
        temp/=10;
        i*=10;
    }
    for(k=0;k<10;k++) //vypocet na 10 pozic
    {
        d*=2;
        if (d>=i)
        {
            printf("1");
            d-=i;
        }
        else
            printf("0");
        if (!d) break;
    }
    return 0;
}

```