INTERPRETOVANÍ DATA

UMĚT NAPROGRAMOVAT PŘEVODY DO LIBOVOLNÝCH SOUSTAV

23. 11. 2024

5. Zobrazení dat v počítači

a) Pojem informace, jednotky pro měření informace

- informace = jakékoliv sdělení, kterým jsme schopni snížit množství existujících variant
  ↳ redundantní ⇒ nenese pro mě informaci, jsem přesvědčen/á (úplná jistota)0
- data = informace zpracovatelné počítačem
- bit (binary digit) = základní jednotka informace; nabývá dvou hodnot 0 nebo 1
- Byte (B) = 8 bitů
- měříme se pracovně i s bitem jako DESETINNÝM ČÍSLEM

Bit (K) = 10^3
Mega (M) = 10^6
giga (G) = 10^9
Tera (T) = 10^{12}

1 KiB = 1024 bytů

Jednotka	Hodnota v bitech	Použití
1 bit	1 bit	základní jednotka
1 byte	8 bitů	1 znak v ASCII kódování
1 kilobit	1000 bitů	přenosová rychlost (např. sítě)
1 kilobyte	1000 bytů	kapacita paměti, disku

* bitový pan dopsán v 1

V počítači uložený data JSOU v binární soustavě.

b) Číselné soustavy, převody mezi soustavami

- Binární (dvojková; základ 2): 0–1
- Trojková (základ 3): 0–2
- Čtyřková (základ 4): 0–3
- Pětková (základ 5): 0–4
- Šestková (základ 6): 0–5
- Sedmičková (základ 7): 0–6
- Osmičková (základ 8): 0–7
- Desítková (základ 10): 0–9
- Šestnáctková (základ 16): 0–9 a A–F

Číselná hodnota = vyjadřuje množství nebo pořadí (NEZÁVISÍ na reprezentaci či zápisu)

Číslice = reprezentace číselné hodnoty

Různé číslice

Různé číselné soustavy

Pozor na hodnotu i pozici číslice; nula i desítková číslice

Reprezentace číselné soustavy X Pozice číselné soustavy

0+0=0
0+1=1
1+1=10 (napíšeme 0 a přenášíme 1 do vyšších bitů)

Desítková → libovolná

Polynomický zápis
1037
$1*10^3+0*10^2+3*10^1+7*10^0$

128 64 32 16 8 4 2 1

Binární → desítková; Desítková → Binární

Čtyřková

[1 1] → 3, 2, 1, 0 ( )4

[1 1 1] → 7 ( )8

$(3102)_4$ → dvojková

1	1	0	1	0	0	1	0
3		1		0		2	

2^3

b. číslo: 1011
$1\cdot2^3+0\cdot2^2+1\cdot2^1+1\cdot2^0=11$

Binární → Šestnáctková

b. číslo: 11011110
1101 = D 1110 = E
DE

30	2
15	0
7	1
3	1
1	1
0	1

0 0 0 1 1 1 1 0

10	A	1	0	1	0										
11	B	1	0	1	1										
12	C	1	1	0	0										
13	D	1	1	0	1										
14	E	1	1	1	0										
15	F	1	1	1	1										

c) Přímý, inverzní, doplňkový kód (kódy ZÁPORNÝCH ČÍSEL)

Přímý kód (PK)

- nejvyšší bit je znaménkový (0 – kladné, 1 – záporné)
- nevýhody: operace jsou složitější kvůli nutnosti se znaménkem i dvě nuly

n = 8 bitů

7	6	5	4	3	2	1	0
znaménkový							n−1

$(00000101)_2 = (+5)_{10}$
$(10000101)_2 = (-5)_{10}$
$(00000000)_2 = (+0)_{10}$
$(10000000)_2 = (-0)_{10}$

DVĚ NULY!

Doplňkový kód (DK)

- nejvyšší bit je znaménkový
- kladná ⇒ nezáporné číslo a přičítáme jedničky
- odečítáme první přičítané záporné číslo
- nevýhody: jen jedna nula
  - složitější aritmetika

NEJSOU DVĚ NULY!

$(00000101)_2 = (+5)_{10}$
$(11111011)_2 = (-5)_{10}$
$(00000000)_2 = (0)_{10}$

11 → $(00001011)_{PK} = 1K = DK$

20 − 11
00010100
+ 11110101
00001001 = $(9)_{10}$

Inverzní kód (IK)

- nejvyšší bit je znaménkový
- kladné číslo je nezměněné kladné číslo
- přináší dvě nuly

n = 8
kdyby 1 přidáno ⇒ +1 na konci

10 − 6 = 10 + (−6)
00001010 (10)
+ 11111001 (−6)
11
+1
00000100

$(00000101)_2 = (+5)_{10}$
$(11111010)_2 = (-5)_{10}$

- nevýhoda dvě reprezentace:
  $(00000000)_2 = (+0)_{10}$
  $(11111111) = (-0)_{10}$
  } DVĚ NULY!

(0 → 1, 1 → 0)

- převrácení přičítání vzít ze PK
- znaménko odečítáme od přičítaného záporného čísla
- ale ke výsledku je třeba přičíst přenos z nejvyššího řádu

Příklady

$(10010001)_{PK} \rightarrow (-17)_{10}$
$(1110\ 1110)_{IK} \rightarrow (-17)_{10}$
$(1110\ 1111)_{DK} \rightarrow (-17)_{10}$

IK 26 − 18 = 26 + (−18) = 8
00011010
+ 11101101
00000111
+1
00001000

DK 26 − 18
00011010
00010010
00011010
11101110 ⇒ 10010 ⇒ 01101 + 1
01101000
1000
přenos z 5. bitu ⇒ 6. bit přičteme a zahodíme

26.11. 2:24

5. Zobrazení dat v počítači

Převody pro desetinná čísla

$(0,4)_{10} \rightarrow (0,0110\ 0110)_2$

↳ v dalších 8 cifrách

X	X*2	celá část
0,4	0,8	0
0,8	1,6	1
0,6	1,2	1
0,2	0,4	0
0,4	0,8	0
0,8	1,6	1
0,6	1,2	1
0,2	0,4	0

Zapiš desítkovou soustavou hodnotu $0,0110\,1101$ v dvojkové soustavě

1/2	1/4	1/8	1/16	1/32	1/64	1/128	1/256
0	1	1	0	0	1	1	0

$\frac{1}{4} + \frac{1}{8} + \frac{1}{64} + \frac{1}{128} = 0,3984375$

Aditivní kód – s posunutou nulou

- např. na 8 bitech → 0 = 127
- mocniny hodnoty jsou různé, rychlé hledání
- omezení množiny aritmetiky
  - +, −: druhý operand musí být stejný
  - *, /: nevhodné
- používá se pro uložení exponentu v plovoucí řádové čárce

$(0)_{10} \rightarrow (0+127)_2 = (0111\ 1111)_2$

$(+5)_{10} \rightarrow (5+127)_{10} = (1000\ 0100)_2$

$(-5)_{10} \rightarrow (-5+127)_{10} = (0111\ 1010)_2$

Pevná řádová čárka

- pro primitivní HW
- pevný počet bitů pro celou a pro desetinnou část
- nelze vyjádřit malé exponenty a malá přesnost
- přibližně pro 3 bity až 8 pro desetinnou část

$(5,5)_{10} \rightarrow (0010\,1,1000)_2$

2

$0,5 = \frac{1}{2} = 2^{-1}$

$0,5 = 0,1000_2$

Plovoucí řádová čárka

- Standard IEEE 754
  - typ float (4B = 32b)
  - typ double (8B = 64b)
    long double (16B = 128b)

n = znaménkový bit ($0 \rightarrow +1$; $1 \rightarrow -1$)

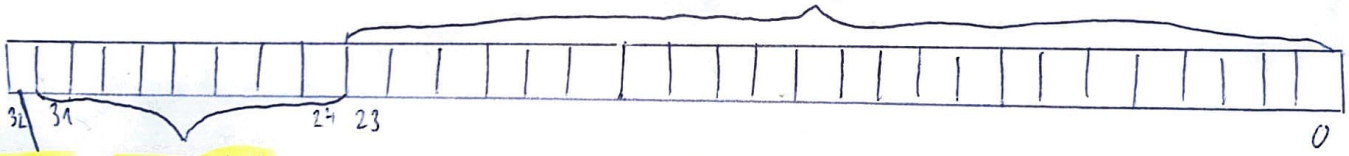
m = mantisa

e = exponent

$X = n \cdot m \cdot 2^e$

Mantisa (23b) = desetinný zlomek čísla

- jedna celá čárka
  ↳ posunutá za celou část



Znaménkový bit

Exponent (8b)

↳ 11b pro double

↳ v aditivním kódu +127 pro float +1023 pro double

Uložení hodnoty 5,5 podle IEEE 754

1) $n = 1$

2) Převod do dvojkové soustavy

$(5,5)_{10} = (101,1)_2$

3) Normalizace

- Proces desetinné čárky pro zpracování počítačem
- Jednička před desetinnou čárkou se ztratí (mantisa)
- Mantisa se doplní zprava nulami

$(5,5)_{10} = (101,1)_2 \rightarrow (1,011)_2 \rightarrow (0\,110\ 0000\ 0000\ 0000\ 0000\ 0000)_2$

4) Exponent uložení před mínus posunutý

↳ máte +2

- aditivní kód: $(+2+127)_{10} = (1000\ 0001)_2$

Nula +0, −0 – 2 nuly možná nesmí $e=0$; $m=0$, $n=1$ nebo 0

Nekonečno +INF, −INF – $n = 0$ nebo 1, $e =$ všechny 1, $m =$ všechny 0

Neplatná hodnota NAN (Not a Number) $n = 0$ nebo 1

↳ dělení nulou nebo

$e =$ všechny 0

$m =$ může být cokoliv kromě 0

↳ dělení nulou