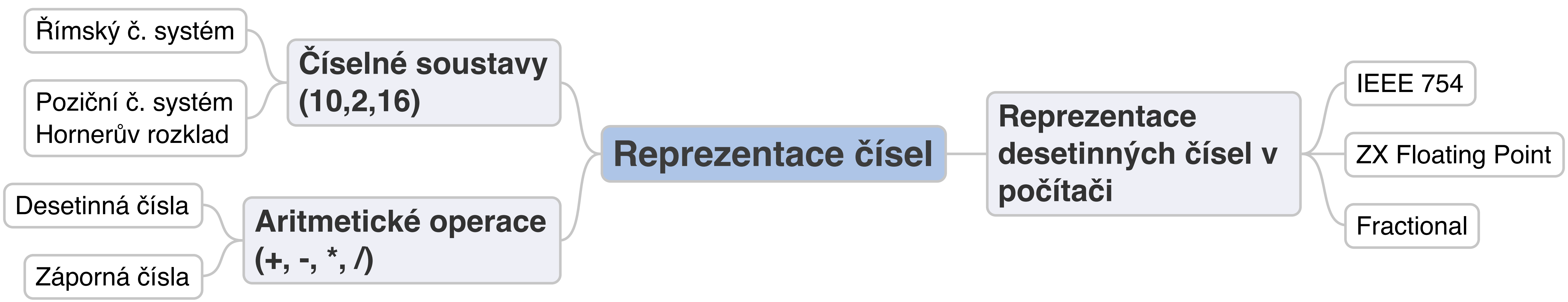
Od aritmetiky Říma k počítačům

Petr Petyovský, @poke128
#JHCon2012
(rev. 4)

Od římské aritmetiky k počítačům



Římský číselný systém

Nepoziční číselná soustava je způsob reprezentace čísel, ve kterém není hodnota číslice dána jejím umístěním v dané sekvenci číslic. [wiki]

5

100 , 50

10

1000 , 500

MMCM = 1000 + 1000 + (1000-100) = 2900

Arabský číselný systém

Poziční číselná soustava je způsob reprezentace čísel, ve kterém pozice číslice v dané sekvenci číslic definuje její hodnotu.

$$\mathbf{324} = \mathbf{3}{*}(100) + \mathbf{2}{*}(10) + \mathbf{4}{*}(1) =$$

$$\mathbf{3}{*}10^2 + \mathbf{2}{*}10^1 + \mathbf{4}{*}10^0$$

Hornerův rozklad je postup jak získat jednotlivé číslice zapsané pomocí pozičního systému. Vede na postupné dělení se zbytkem.

$$\mathbf{324} = \mathbf{3}{*}10^2 + \mathbf{2}{*}10^1 + \mathbf{4}{*}10^0 =$$

$$\{\ [(\mathbf{3}) * 10] + \mathbf{2}\ \} * 10 + \mathbf{4}$$

Reprezentace reálných čísel

Báze číselné soustavy (libovolné číslo) představuje základ soustavy, která je na další pozici umocněna. Pro řády soustavy reprezentující hodnoty menší než nula se využívá převrácená hodnota báze.

$$\mathbf{32.4} = \mathbf{3}^{*}(10) + \mathbf{2}^{*}(1) + \mathbf{4}^{*}(1/10) =$$

$$\mathbf{3}^{*}10^{1} + \mathbf{2}^{*}10^{0} + \mathbf{4}^{*}10^{-1}$$

Pravidla rozkladu platí pro všechny poziční soustavy o libovolném základu.

Příklady převodu do dvojkové soustavy

$$13_{10} = ?_2$$

$$121_{10} = ?_2$$

$$0.6875_{10} = ?_2$$

$$0.53_{10} = ?_2$$

Aritmetika ve dvojkové soustavě

Sčítání: 0+0=0; 1+0=1; 0+1=1; 1+1=0 (+přenos); 1+1+1=1 (+přenos)

Odečítání: 0-0=0; 1-0=1; 1-1=0; 0-1=1 (+výpůjčka); 0-1-1=0 (+výpůjčka)

Definice záporných čísel: nutno definovat pro výsledky operace odčítání (výsledek může být záporný). Existuje několik variant reprezentace:

- Absolutní hodnota + znaménko
- Posunutá nula
- První doplněk
- Druhý doplněk

Aritmetika ve dvojkové soustavě (pokrač...)

Operace násobení a dělení: Definovány jako rozlehlé sčítání resp. odečítání

```
  0010 1101
 *0001 0100
 ----------
          ?
```

100 0001 0110 : 11 1010 =

Reprezentace desetinných čísel dle IEEE 754

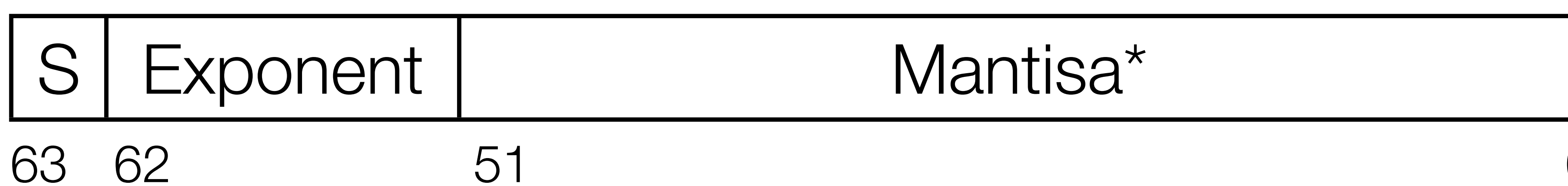
Norma definující zápis libovolného čísla pomocí dvou čísel:

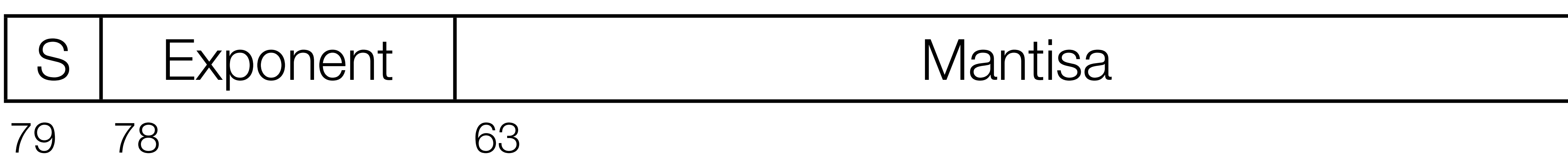
- Mantisa ve tvaru: desetinné číslo se znaménkem s omezeným počtem bitů
- Exponent: celé číslo se znaménkem s omezeným počtem bitů

$$Mantisa * 2^{Exponent}$$

Mantisa se udržuje vždy v binárnímu tvaru: $1.xxxxxxx_2$

Pozn: Mantisa* - Mantisa v úsporném tvaru.





Příklady desetinných čísel dle IEEE 754

Exponent reprezentace posunutá nula.
Offset $7F_{16}$ představuje 0.

31	30	22	0
S	Exp	Mantisa*	
1	10...0	101100...0	

$$= -1.1011_2 * 2^1 =$$

$$= -11.011_2 = -3.375_{10}$$

Speciální hodnoty/symboly dle IEEE 754

31 30 22 0

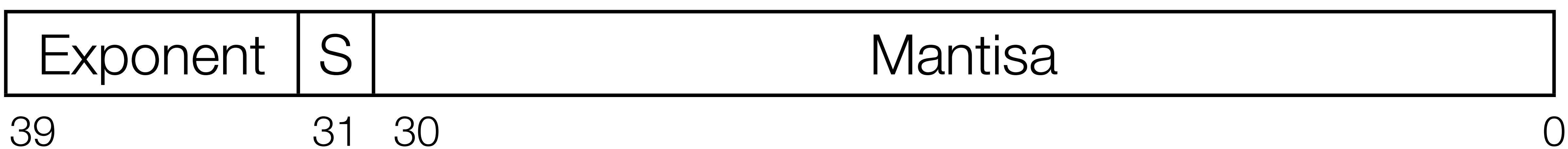
S	Exp	Mantisa*	
0	0...0	0...0	$= +0$
1	0...0	0...0	$= -0$
0	1...1	0...0	$= +\infty$
1	1...1	0...0	$= -\infty$
X	1...1	<100...01 - 101...11>	Quite Not a Number (qNaN) (nehlášené NaN)
X	1...1	110...00	Indefinite - Neurčitá hodnota
X	1...1	<110...01 - 111...11>	Signaled Not a Number (sNaN) (gen. vyjimku)

Více na: http://en.wikipedia.org/wiki/NaN

ZX Floating point formát

Princip prakticky shodný s IEEE 754 (znaménko, exponent, mantisa). Využívá 5-ti bajtů (40 bitů).

Existuje několik komprimovaných způsobů zápisu (menší počet bajtů, obdobně jako u IEEE 754).



Fractional formát

Fixed point formát (zlomkový formát): Výpočet jako celočíselný formát. Každé číslo reprezentované jako zlomek s pevný jmenovatelem.

$$\frac{frac8A}{256}+\frac{frac8B}{256}=\frac{(frac8A+frac8B)}{256}$$

$$\frac{frac8A}{256}-\frac{frac8B}{256}=\frac{(frac8A-frac8B)}{256}$$

$$\frac{frac8A}{256}\cdot\frac{frac8B}{256}=\frac{(frac8A\cdot frac8B)}{65536}=\frac{frac16}{65536}=\frac{round8(frac16)}{256}$$

$$\frac{frac8A}{256}:\frac{frac8B}{256}=\frac{frac8A}{256}\cdot\frac{256}{frac8B}=\frac{frac8A}{frac8B}$$

Závěrem aneb co nám chybí

ZX komunita nemá jednoduché assemblerové nástroje pro používání ROM kalkulátoru. tj. definice maker, definice operačních kódů matematických operací, apod...

Všichni demomakeři využívají fractional aritmetiku, přesto neexistuje veřejně dostupná knihovna pro obecné použití.

Další prostor pro rozšíření Z80bits knihovny...

Děkuji za pozornost...