XE36SKD

Seminar

Fixed-point and Floating-point Numbers

Tomáš Brabec, Miroslav Skrbek, Josef Hlaváč, Jiří Buček

Seminar 3

- Representation of Negative Numbers
  - Sign-magnitude code
  - Biased code (Excess-N, “additive” code)
- Fixed-point numbers
- Floating-point numbers

References

- http://en.wikipedia.org/wiki/Signed_number_representations
- http://en.wikipedia.org/wiki/Fixed-point_arithmetic
- http://en.wikipedia.org/wiki/Floating_point

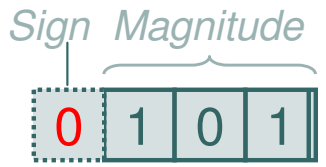
Sign-magnitude code

- Most significant digit denotes the **sign**, the rest is the **magnitude** (absolute value)
- Sign represented by a digit: +…0, -…1
- Formal definition: $\mathcal{P}(A) = \begin{cases} A & \text{for } 0 \le A < z^n \\ Z/z - A & \text{for } -z^n < A \le 0 \end{cases}$

Ex. Images of +5 and –5 ($z = 2$, $Z = 16$).

$\mathcal{P}(5) = 5_{10} = 101_2$ … $+101_2$ $\xrightarrow{\mathcal{P}}$ 0 1 0 1

$\mathcal{P}(-5) = (16/2)_{10}-(-5_{10}) = 13_{10} = 1101_2$… -101_2 $\xrightarrow{\mathcal{P}}$ 1 1 0 1



Biased code

- Also Excess-*N* (Excess-*K* here), "Additive" code
- Formal definition: $\mathcal{A}_K(A) = A + K \quad \text{for } -K \le A < Z - K$
- Often we choose: $K = ½\, Z$ $K=$ 

Ex. Images of +5 and –5 ($z = 2$, $Z = 16$, $K = 8$).

$\mathcal{A}(5) = 5_{10} + 8_{10} = 13_{10} = 1101_2 \quad \ldots +101_2$ $\mathcal{P}$ 

$\mathcal{A}(-5) = (-5_{10}) + 8_{10} = 3_{10} = 0011_2 \quad \ldots -101_2$ $\mathcal{P}$ 

Exercise: Encode in Sign-magnitude and Biased code.

- Convert the numbers to binary and write them down according to given parameters.

1. $(-9)_{10} = ?_2 \quad n = 5,\ m = 0$
2. $(17)_{10} = ?_2 \quad n = 7,\ m = 0$
3. $(-6C)_{16} = ?_2 \quad n = 8,\ m = 0$
4. $(-0)_{16} = ?_2 \quad n = 4,\ m = 0$

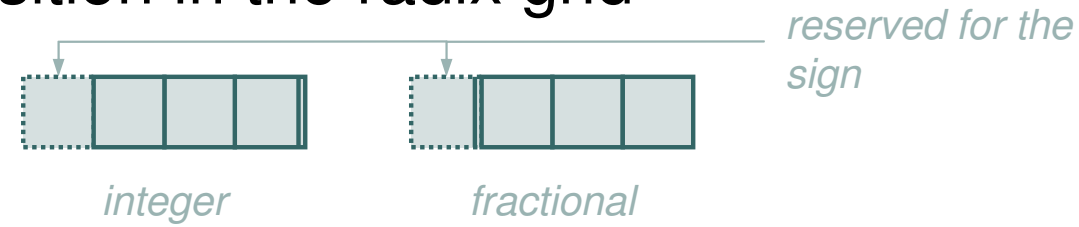
Exercise: Encode in Sign-magnitude and Biased code.

- Convert the numbers to binary and write them down according to given parameters.

1. $(-9)_{10} = ?_2$ … $\mathcal{P} = 10\ 1001_2$ $\mathcal{A} = 01\ 0111_2$
2. $(17)_{10} = ?_2$ … $\mathcal{P} = 0001\ 0001_2$ $\mathcal{A} = 1001\ 0001_2$
3. $(-6C)_{16} = ?_2$ … $\mathcal{P} = 1\ 0110\ 1100_2$ $\mathcal{A} = 0\ 1001\ 0100_2$
4. $(-0)_{16} = ?_2$ … $\mathcal{P} = 1\ 0000_2$ $\mathcal{A} = 1\ 0000_2$

Fixed-point numbers

- also ***fix-point***
- Radix point – fixed position in the radix grid
  - Most widely used:



- Significantly limited range

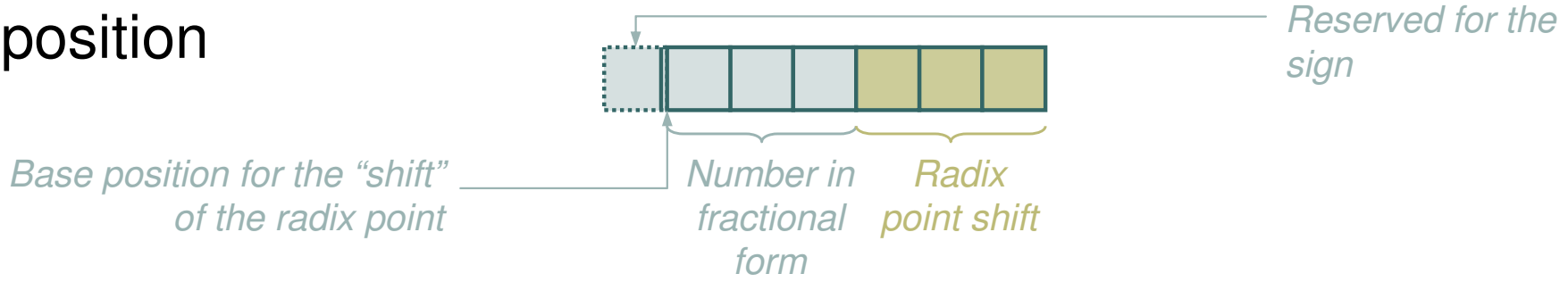
Ex. z = 2, length of r.g. l = 32 (i.e. 32-bit number)

Max. integer number … $A < 2^{32} < 5 \cdot 10^{9}$

Min. fractional number … $A > 2^{-32} > 2 \cdot 10^{-10}$

Floating-point I

- Similar to scientific notation
- Radix point – determined by a shift from the base position



- Introducing the "shift" e increases the representable range significantly: $A_2 = M \cdot 2^e$

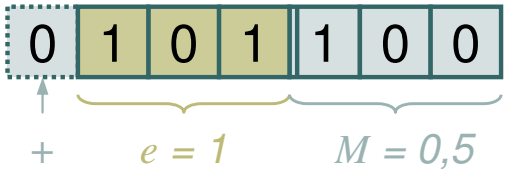
Most significant part of the number (fractional)

Coefficient resulting from the "shift" of the radix point

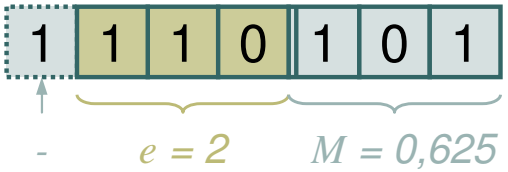
Floating-point II

- Radix grid has 2 parts (sub-grids):
  - **mantissa** (M) – "value" of the number, often fractional
  - **exponent** (e) – position of the radix point, integer
- M and e use **codes** to represent signed numbers
- Most often used format for floating-point numbers:

Ex. $(0{,}5 \cdot 2^1)_{10}$



Př. $(-5 \cdot 2^{-1})_{10}$



Floating-point III

- Value of a number $\Rightarrow$ many representations

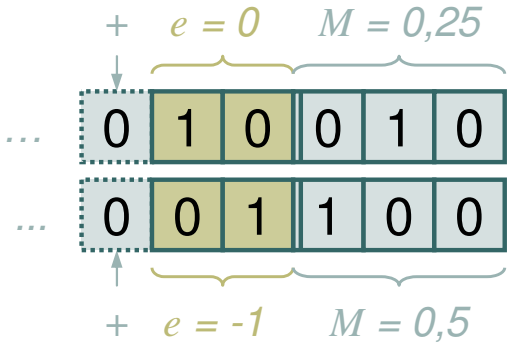
  Ex. $A = 20 = 5 \cdot 2^2 = 1{,}25 \cdot 2^4 = 0{,}625 \cdot 2^5 = 0{,}078125 \cdot 2^8$

- **Normalized form**
  - **Mantissa is shifted to the left as far as possible**
  - Unique representation
  - Simplifies arithmetic operations

  Ex. $A = 2^{-2} = 0{,}01_2 \cdot 2^0 = 0{,}1_2 \cdot 2^{-1}$

Not normalized

Normalized



Exercise: Write in normalized form.

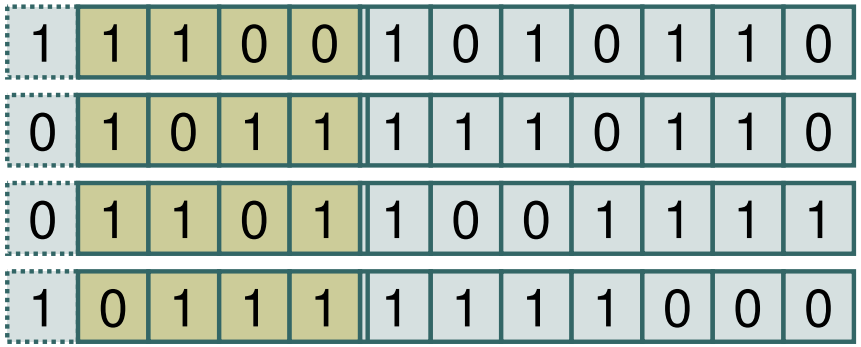
- Assume total length l = 12, the exponent sub-grid length is l_e = 4. Exponent uses biased code, mantissa uses sign-magnitude code.

1. $-(1010{,}11_2)$
2. $7{,}375_{10}$
3. $13{,}C_{16}$
4. $-(46{,}875 \cdot 10^{-2})_{10}$

Exercise: Write in normalized form.

- Assume total length l = 12, the exponent sub-grid length is l_e = 4. Exponent uses biased code, mantissa uses sign-magnitude code.

1. $-(1010{,}11_2)$
2. $7{,}375_{10}$
3. $13{,}C_{16}$
4. $-(46{,}875 \cdot 10^{-2})_{10}$



Arithmetic in floating point

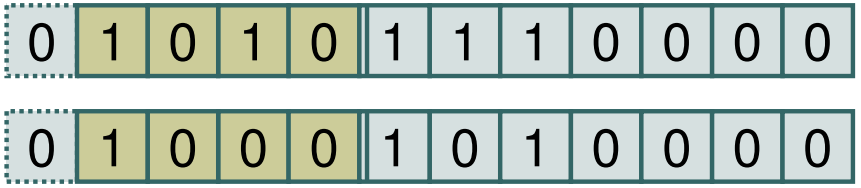
- Arithmetic operations:
  - **add/subtract:** Align exponents, add/subtract mantissas.
  - **multiply:** Add exponents, multiply mantissas.
  - **divide:** Subtract exponents, divide mantissas.
  - **compare:** Align exponents, compare mantissas.
  - **shift:** Enlarge/reduce exponent, or shift mantissa.
- Normalized operands do not guarantee normalized result!
  $\Rightarrow$ **normalize**
  - i.e. **convert the result to the normalized form**
  - must be performed after every operation

Example: Addition in floating point

- Write the numbers $3{,}5_{10}$ and $0{,}625_{10}$ in normalized form and then add them.

$3{,}5_{10} = 11{,}1_2 = 0{,}111_2 \cdot 2^2$

$0{,}625_{10} = 0{,}101_2 \cdot 2^0$



$$3{,}5_{10} + 0{,}625_{10} = 0{,}111_2 \cdot 2^2 + 0{,}101_2 \cdot 2^0 =$$
$$= (0{,}111_2 + 0{,}00101_2) \cdot 2^2 =$$
$$= (1{,}00001_2) \cdot 2^2 = 0{,}100001_2 \cdot 2^3$$

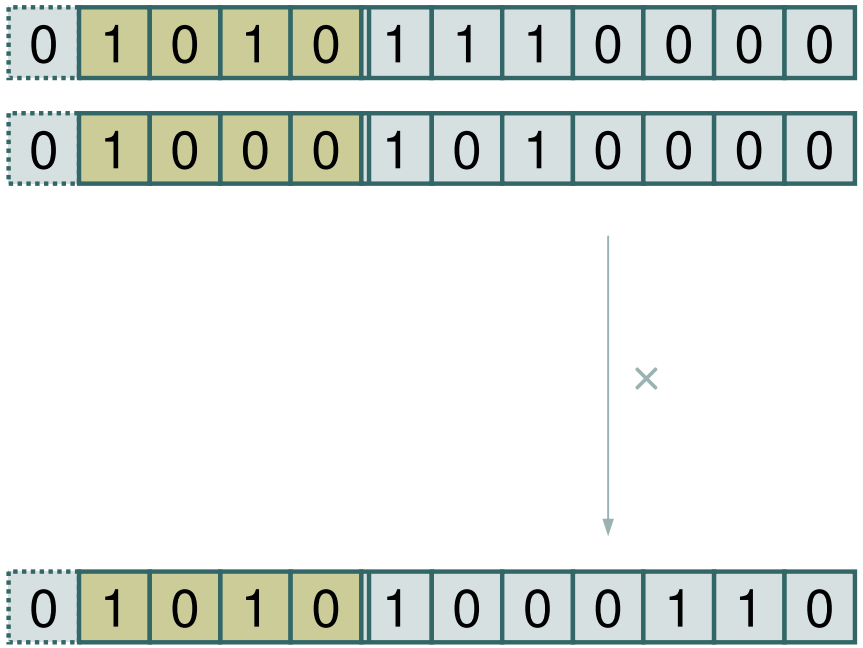
Example: Multiplication in floating point

- Write $3{,}5_{10}$ and $0{,}625_{10}$ in normalized form and multiply them.

$3{,}5_{10} = 0{,}111_2 \cdot 2^2$

$0{,}625_{10} = 0{,}101_2 \cdot 2^0$

$$
\begin{array}{rl}
 & 0{,}111_2 \\
\times & 0{,}101_2 \\
\hline
 & 0{,}0111_2 \\
+ & 0{,}00000_2 \\
+ & 0{,}000111_2 \\
\hline
 & 0{,}100011_2
\end{array}
$$



Exercise: Compute in floating point

- Assume total length l = 12, the exponent sub-grid length is l_e = 4. Exponent uses biased code, mantissa uses sign-magnitude code.

1. $10{,}375_{10} \times 0{,}125_{10}$
2. $13{,}625_{10} + 1{,}375_{10}$
3. $4{,}C_{16} - 3_{16}$
4. $(-0{,}40625_{10} \cdot 2^{-3}) \times (-0{,}28125_{10})$

Exercise: Compute in floating point