Computer Systems and -architecture

Data Representation

1 Ba INF 2017-2018

Stephen Pauwels
stephen.pauwels@uantwerpen.be

Time Schedule

Exercises are made individually. Put all your files in a tgz archive, as explained on the course's website, and submit your solution to the exercises on Blackboard. **Write down all intermediate results on how you obtained the results.**

- Deadline: **November 12, 23u55**

Exercises

1. Convert these positive numbers to base 10.
   (a) $(101100011)_2$
   Solution: $(355)_{10}$
   (b) $(A36E)_{16}$
   Solution: $(41838)_{10}$
   (c) $(111001010100)_2$
   Solution: $(3668)_{10}$
   (d) $(641)_8$
   Solution: $(417)_{10}$
   (e) $(666)_9$
   Solution: $(546)_{10}$
2. Convert to base 10.
   (a) $(10110001)_2$ (2's complement)
   Solution: **-79**
   (b) $(11)_2$ (2's complement)
   Solution: **-1**
   (c) $(0.3112)_4$
   Solution: $\frac{3}{4} + \frac{1}{16} + \frac{1}{64} + \frac{2}{256} = \frac{214}{256} =$ **0.8359**
   (d) $(0.123)_{13}$
   Solution: **0.090**
3. Convert to base 2.
   (a) $(2017)_{10}$
   Solution: $(11111100001)_2$

(b) $(666)_8$
Solution: $(110110110)_2$

(c) $(14AD)_{16}$
Solution: $(1010010101101)_2$

(d) $(5.32)_{10}$
Solution: $(101.0101000)_2$

(e) $(42)_{16}$
Solution: $(1000010)_2$

4. Convert to base 2. Represent the negative numbers with 8 bits in *signed magnitude*, *one's complement*, *two's complement* and *excess 128*.

   (a) $(-111)_{10}$
   Solution: $(11101111)_2$, $(10010000)_2$, $(10010001)_2$, $(00010001)_2$

   (b) $(-86)_{10}$
   Solution: $(11010110)_2$, $(10101001)_2$, $(10101010)_2$, $(00101010)_2$

   (c) $(-131)_{10}$
   Solution: **Not possible!!! Only possible range up to -127**

   (d) $(-1A)_{16}$
   Solution: $(10011010)_2$, $(11100101)_2$, $(11100110)_2$, $(01100110)_2$

5. For the following single-precision IEEE 754 bit patterns, show the numerical value as a base 2 significand with an exponent (e.g. $+1.11 \cdot 2^5$).

   (a) `0 10010101 00111011000000000000000`
   Solution: $+1.00111011 \cdot 2^{22}$

   (b) `1 01000001 10100000000000000000000`
   Solution: $-1.101 \cdot 2^{-62}$

   (c) `0 11111111 00000000000000000000000`
   Solution: $+\infty$

   (d) `0 00000000 00010111000000000000000`
   Solution: $1.0111 \cdot 2^{-131}$ **(denormalized form)**

   (e) `1 00010111 11100110000000000000000`
   Solution: $-1.1110011 \cdot 2^{-104}$

   (f) `0 10101011 01101000000000000000000`
   Solution: $+1.01101 \cdot 2^{44}$

   (g) `0 11111111 11010100010001010100010`
   Solution: **+NaN**

6. Represent these numbers in the *IEEE-754 (single precision)* format.

   (a) $(8982.5)_{10}$
   Solution: **0 10001100 00011000101101000000000**

   (b) $(2017)_{10}$
   Solution: **0 10001001 11111000010000000000000**

   (c) NaN
   Solution: **0 11111111 11100110110000010010000**

   (d) $(-42.666)_{10}$
   Solution: **1 10000100 01010101010100111111100**

(e) $+\infty$
Solution: **0 11111111 00000000000000000000000**

(f) $+0$
Solution: **0 00000000 00000000000000000000000**

(g) $(1.110101 * 2^{-134})_2$ (denormalized)
Solution: **0 00000000 00000011101010000000000**

(h) $(333.0)_{10}$
Solution: **0 10000111 01001101000000000000000**

7. Suppose we are using a 15 bit floating point, in a normalized, base 8 floating point format, with a sign bit, followed by a 5-bit exponent with a certain bias, followed by three base 8 digits.

   (a) Determine the bias we have to use for the exponent, assuming we do not want to change the range of exponents we would have reached when using a 5-bit 2's complement exponent.
   Solution: **Range goes from -16 to 15, so we use a bias 16 exponent**

   (b) Represent the number $(-142)_{10}$ in our new format (with the bias from the previous question) as a binary string.
   Solution: **1 10011 010 001 110 (in octal:** $.216 * 8^3$)

   (c) What is the largest possible error that can be made using this representation?
   Solution: **Notice that the maximal error made by this representation is equal to the largest possible gap between two consecutive numbers that can be represented by the floating point representation. Next just use the formulas given for Largest Gap (b^M x b^{-s}), with s the number of significant digits in the fraction and M the largest exponent.** $b = 8, M = 15, s = 3$ **so we have** 8^{15} **x** $8^{-3} = 8^{15-3} = 8^{12}$

   (d) What is the smallest gap using this representation?
   Solution: b^m **x** $b^{-s} = 8^{-16}$ **x** $8^{-3} = 8^{-16-3} = 8^{-19}$

8. Write a Python program that, using the module `files`, does the following:

   (a) Read the given file `input.txt` using the correct encoding.

   (b) Write the contents you just read back to file using the UTF-16 encoding scheme.

   (c) Convert all characters to their appropriate code points.

   (d) Convert the code points to their correct html code, make sure you can display a new line in a correct manner.

   The module `files` has the following functions:

   - `read_file(filename, encoding)`: this function takes a filename and reads it according to the given encoding and returns the resulting string.
   - `write_file(filename, contents, encoding)`: this functions writes the string `contents` to the file `filename` using the given encoding.
   - `write_html_file(filename, contents)`: writes the string `contents` to a html file.

   The module has the following encodings:

   - `ASCII`
   - `UTF_8`
   - `UTF_16`