Computer Systems and -architecture

Data Representation

1 Ba INF 2015-2016

Bart Meyers
bart.meyers@uantwerpen.be

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 5, 23u55**

Exercises

1. Convert these positive numbers to base 10.
   (a) $(10110111100)_2$
   (b) $(3A6E)_{16}$
   (c) $(111001)_2$
   (d) $(164)_8$
2. Convert to base 10.
   (a) $(11101011)_2$ (2's complement)
   (b) $(11111)_2$ (2's complement)
   (c) $(.213)_4$
3. Convert to base 2.
   (a) $(666)_{10}$
   (b) $(316)_{10}$
   (c) $(AD14)_{16}$
   (d) $(7.75)_{10}$
   (e) $(14AD)_{16}$
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) $(-115)_{10}$
   (b) $(-137)_{10}$
   (c) $(-20)_{10}$

(d) $(-13)_{\mathbf{16}}$

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 10001110 00111010000000000000000`

   (b) `1 10000000 10000000000000000000000`

   (c) `0 11111111 00000000000000000000000`

   (d) `0 00000000 01101000000000000000000`

   (e) `0 11111111 11010000000000000000000`

   (f) `1 00000111 11100110000000000000000`

   (g) `0 00001011 01101000000000000000000`

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

   (a) $(2078.25)_{10}$

   (b) $(2015)_{10}$

   (c) $(1.618)_{10}$

   (d) $-\infty$

   (e) $+0$

   (f) $(1.11 * 2^{-129})_2$ (denormalized)

7. Suppose we are using a 22 bit floating point, in a normalized, base 16 floating point format, with a sign bit, followed by a 5-bit exponent with a certain bias, followed by four base 16 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.

   (b) Represent the number $-0.155 * 10^2$ in our new format (with the bias from the previous question) as a binary string.

   (c) What is the largest possible error that can be made using this representation?

   (d) What is the smallest gap using this representation?