

Computer Architecture

*Homework assignment of section*

*Representing and Manipulating Information*

Deadline: 3 March, 2023

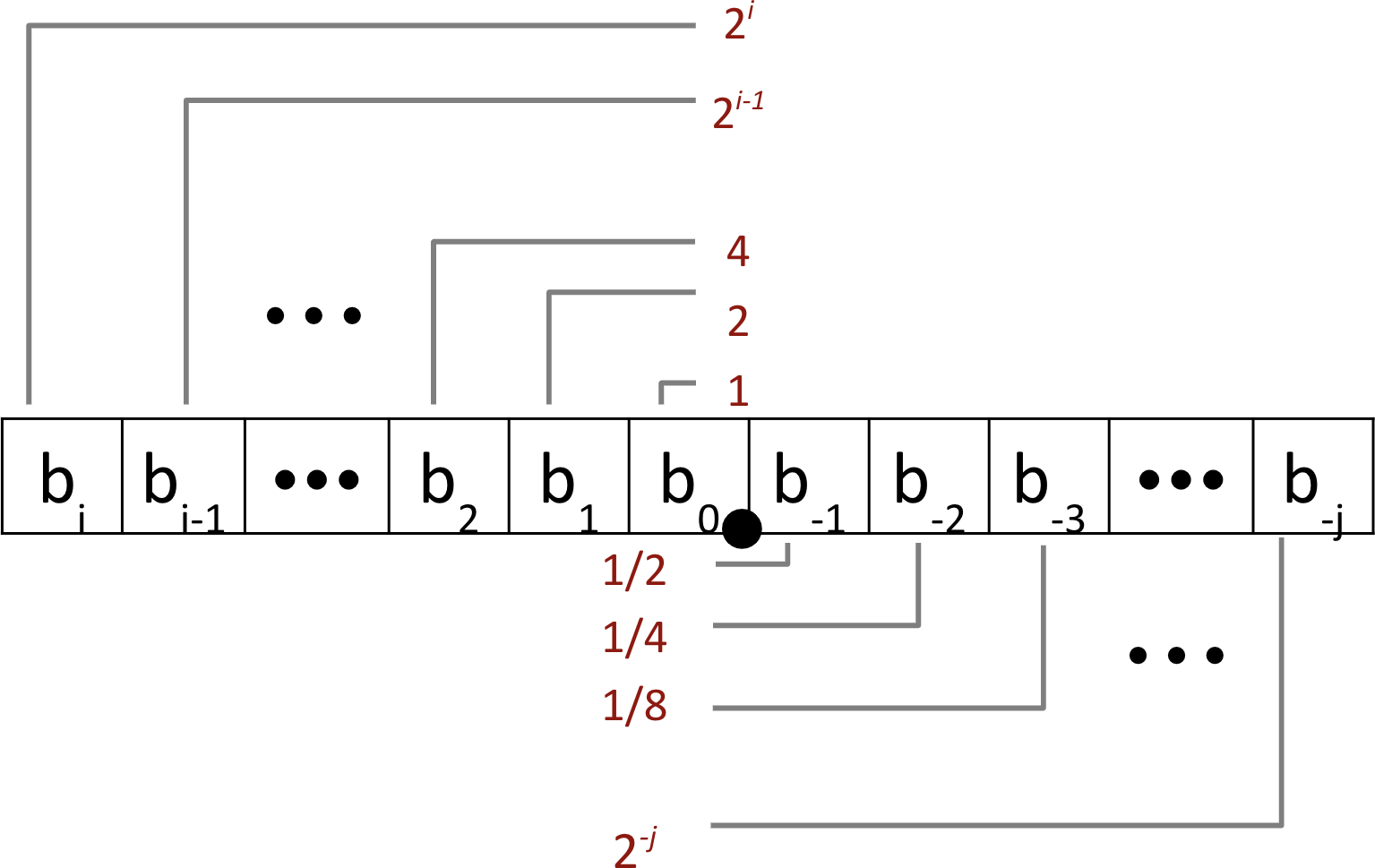
Upload one solution per group to Brightspace!

# TOPICS COVERED

* Fractional binary numbers
* Floating point (FP) numbers
* FP rounding
* FP types in C

1

# FRACTIONAL BINARY NUMBERS



1. Discuss and solve Practice Problem 2.45 (Page 147)

|  |  |  |
| --- | --- | --- |
| Fractional value | Binary representation | Decimal representation |
|  | 0.001 |  |
|  | 0.110 | 0.75 |
|  | 0.101 | 0.3125 |
|  | 10.1011 | 3.6875 |
|  | 1.001 | 1.125 |
|  | 101.111 | 5.875 |
|  |  | 3.1875 |

1. Discuss and solve Practice Problem 2.46

# FLOATING POINT NUMBERS

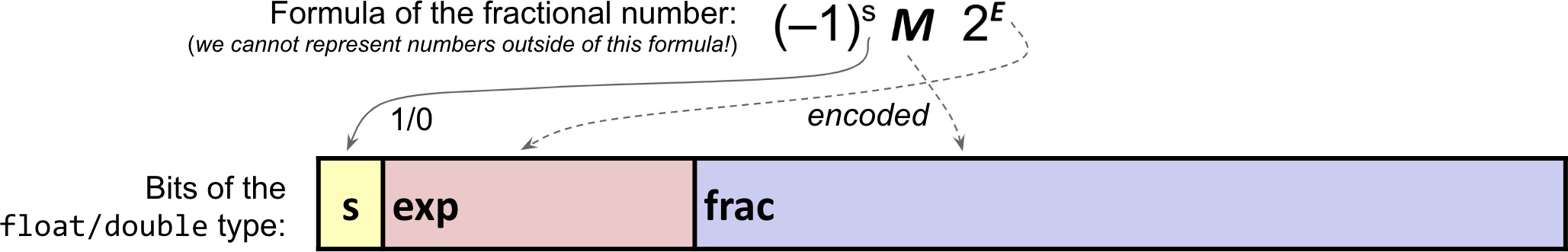


Illustration of values that can be represented by an 8-bit IEEE-format float with 4 bits exponent and 3 bits fraction (it’s a vector-based image, you can zoom in a lot without losing quality): [http://www.volkerschttp://www.volkerschatz.com/science/pics/fltscale.svghatz.com/scien](http://www.volkerschatz.com/science/pics/fltscale.svg) [ce/pics/fltscale.svg](http://www.volkerschatz.com/science/pics/fltscale.svg)

Notice how the spacing between adjacent numbers jumps by a factor of two at every

power of two (above 0.25, which is the smallest normalized number in this tiny float).

1. Solve the following lines of Practice Problem 2.47 in your group (Page 153). a. 0 00 01

b. 0 01 11

c. 0 10 11

d. 0 11 00

e. 0 11 01

# FLOATING POINT ROUNDING

1. Discuss and solve Practice Problem 2.50 (Page 157).
2. Discuss and solve Practice Problem 2.52 (Page 158).
3. Consider the following floating point representation where each number has 2 exponent bits and 5 fraction bits. The exponent bias is 1. Round each number to have 3 fraction bits. Apply round-to-even and round-toward-zero modes.

a. 1 10 11100

b. 0 00 00101

c. 1 01 10101

* 1. Can you see a pattern for rounding floating point numbers when the number of exponent bits are intact?

# FLOATING POINT TYPES IN C

1. Discuss and solve Exercise Problem 2.89 (Page 175)

# OPTIONAL HOMEWORK

The *IEEE Standard 754* is not the only possible way to encode fractional numbers to binary format. If you are interested in a different one, check out this paper that proposes a new floating point format.

[Beating Floating Point at its Own Game: Posit Arithmetic 1. Background: Type I and Type](http://www.johngustafson.net/pdfs/BeatingFloatingPoint.pdf) [II Unums](http://www.johngustafson.net/pdfs/BeatingFloatingPoint.pdf)