Computer Logic 1

Giorgio Grigolo

October 25, 2021

Contents

Lecture 1: Tutorial Sheet 1

Write the following numbers in 2's complement:

$$1_{10} = 00000001_2 = 11111111_2$$

$$56_{10} = 00111000_2 = 11000111_2 + 1_2 = \textit{something}$$

$$128_{10} = 10000000_2 = 01111111_2 + 1_2 = \textit{somthing}$$

Lecture 2: Floating point addition

Converting 2.25_{10} into **IEEE 754**:

$$\begin{aligned} 2.25 &= 2 + 0.25 \\ &= 2^1 + 2^{-2} \\ &= (1 + 2^{-3}) \times 2^1 \\ &= (-1)^s \times (1 + m) \times 2^{(x-127)} \\ &= 0\ 10000000\ 0010\ 000\ 000\ 000 \end{aligned}$$

s	x	m
1	1000 0000	0010 0000 0000 000

Converting 134.0625_{10} into **IEEE 754**:

$$\begin{aligned} 134.0625 &= \left(\frac{2^7}{2^7} + \frac{6}{2^7} + \frac{0.0625}{2^7}\right) \times 2^7 \\ &= (1 + \frac{6}{2^7} + \frac{2^{-4}}{2^7}) \end{aligned}$$

In floating point arithmetic, operations become more complex. The following is the algorithm for addition:

- Check for zeros. If one operand is zero, the result is the other operand.
- Align the signifcands of the other operands.
- Add the significands.
- Normalize the result.