Laboratory work 5 for TSTE18 Digital Arithmetic
Floating-Point Arithmetic

Oscar Gustafsson

September 13, 2017

The purpose of this laboratory work is to realize different arithmetic operations with the help of a computer. The recommended programming languages to use are **Matlab**, **VHDL**, and **Verilog**. However, if you feel more comfortable with using another language feel free to do so, but make sure you have confirmed with the course responsible that they can understand that programming language and have a fair chance of running your instances. **It is OK to use different programming languages for different parts/labs.**

It is required that you use a digit-based representation format for all your data, i.e., arrays, rather than integer types. Feel free to write conversion functions for easy use, but the main realization should use a digit-based representation in the appropriate radix.

The laboratories are **nominally individual**. However, as there are a few more students than can fit in a lab, it is OK to use the smallest number of pairs possible per lab. Note that each student is only allowed to work in pairs a minimum number of times. Accordingly, even though you happen to be working in a pair, you must make sure that you understand everything you submit. Also, note that the pairings will be decided in the lab, based on the number of students attending. Hence, there is no point in planning the pairs beforehand as some students may choose to do the labs at home. **All source codes will be cross-correlated, so please write your own and do not get "inspired"**.

The reporting should consist of (emailed to `oscar.gustafsson@liu.se`):

- Source code
- Example run, showing the usage of the realization
- Some non-trivial examples showing the correctness of the realization
- Where applicable, make sure that all relevant intermediate results are also shown in the examples
- Where applicable, drawings (hand-written is OK scanned/photographed and emailed or handed in on paper)

For Matlab, an .m-file with the runs and the output log would be fine. For VHDL/Verilog a .do-file setting up the windows etc and providing stimuli or a testbench may be appropriate.

> It is OK to have separate functions for the different cases.

1 Floating-point addition

Realize a floating-point addition with IEEE 754 single precision inputs. Note that you can base your design on the sign-magnitude adder code from the first lab. It is enough to realize one rounding mode, but specify how the rounding is done.

2 Floating-point multiplication

Realize a floating-point multiplication with IEEE 754 single precision inputs. Note that you can base your design on the unsigned binary multiplier code from the third lab. The same thing holds for rounding as for the adder.