Laboratory work 1 for TSTE18 Digital Arithmetic
Number representations and addition

Oscar Gustafsson

August 28, 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 responsibles 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"**.

On each sheet of paper write name, personal id number, and student-id, as well as the consecutive number assigned to you.

The reporting should consist of (emailed to oscar.gustafsson@liu.se or handed in to Oscar Gustafsson):

- 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 and in this case it is OK to physically hand them in)

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.

1 Fractional radix converter

Realize a function that can convert fixed-point numbers from one radix to another. The inputs should be: integer part, fractional part, source radix, destination radix.

> Note that you are dealing with digits, and, hence, suitable input and output formats are arrays of digits. Possible calls in Matlab could therefore be:
> `radixconversion([2 5 6], [3 4], 7, 3)`
> `radixconversion([24 15 6], [13 24], 31, 7)`

2 Sign-magnitude adder/subtracter

Realize a radix-2 sign-magnitude adder/subtracter. The inputs should be: two input words, add/subtract control. **Sketch** a possible high-level architecture of an implementation.

3 Binary signed-digit adder

Realize a binary signed-digit adder with limited carry-propagation. The inputs should be: two input words.

4 High-radix carry-free adder

Realize a high-radix carry-free adder. The inputs should be: two input words, radix, lower and upper bounds on acceptable digits.